IEEE Transactions on Robotics and Automation, vol. 11, pp. 585-597, Aug. 1995

Base–Invariant Symmetric Dynamics of Free–Flying Manipulators

Abhinandan Jain and Guillermo Rodriguez
Jet Propulsion Laboratory/California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

Abstract

The freedom of choice in designating a base–body for free–flying manipulators gives these manipulators a base–invariance symmetry that is not encountered in terrestrial manipulators. This paper analyzes the relationship between this natural symmetry and the dynamical equations for free-flying manipulators. The base-invariance symmetry is used to develop a new formulation of the manipulator dynamics in which two independent $O(\mathcal{N})$ recursions proceeding in opposite directions are summed together to obtain the complete free–flying manipulator dynamics. Computation of the operational space inertia for the links in the manipulator is also discussed.

1 Introduction

The dynamics and control of free-flying manipulators has received considerable attention during recent years [1–5]. In addition to the internal hinge degrees of freedom, free–flying manipulators have 6 additional degrees of freedom associated with the overall location and orientation of the manipulator. These additional 6 degrees of freedom are typically assigned to the manipulator base–body. While the base–body link is typically chosen based on design and use considerations, from a more general perspective, any one of the manipulator links can be designated as the base–body with equal validity. This freedom of choice, not available for terrestrial manipulators, is responsible for a *base–invariance symmetry* possessed by free-flying manipulators. In this paper, we use this symmetry to study free-flying manipulator dynamics.

The point of departure is the well known class of $O(\mathcal{N})$ articulated body inertia forward dynamics algorithms for general manipulators [6–8]. These algorithms are highly sequential and consist of the spatially recursive computation of the generalized accelerations for the manipulator. We use the base–invariance symmetry of free-flying manipulators to transform this forward dynamics algorithm into one with a highly decoupled structure. The new algorithm consists of a pair of independent articulated body inertia recursions which proceed in opposite directions. The structure of the algorithm offers obvious advantages for parallelization. It is also shown that the operational space inertia inertia for free-flying manipulators can be obtained by simply combining the quantities computed by the pair of articulated body inertia recursions. Our analysis is general and applicable even when the base body forces are non–zero, that is, even when the linear and angular momenta are not constant. The subject of inverse dynamics algorithms for free-flying manipulators has previously been dealt with in reference [9] as a special case of the class of under–actuated manipulators.

2 Modeling and Dynamics of Manipulators

Consider a general serial manipulator with n rigid body links. As shown in Figure 1, the links are numbered in increasing order from tip to base. The outer most link is link 1, and for now we

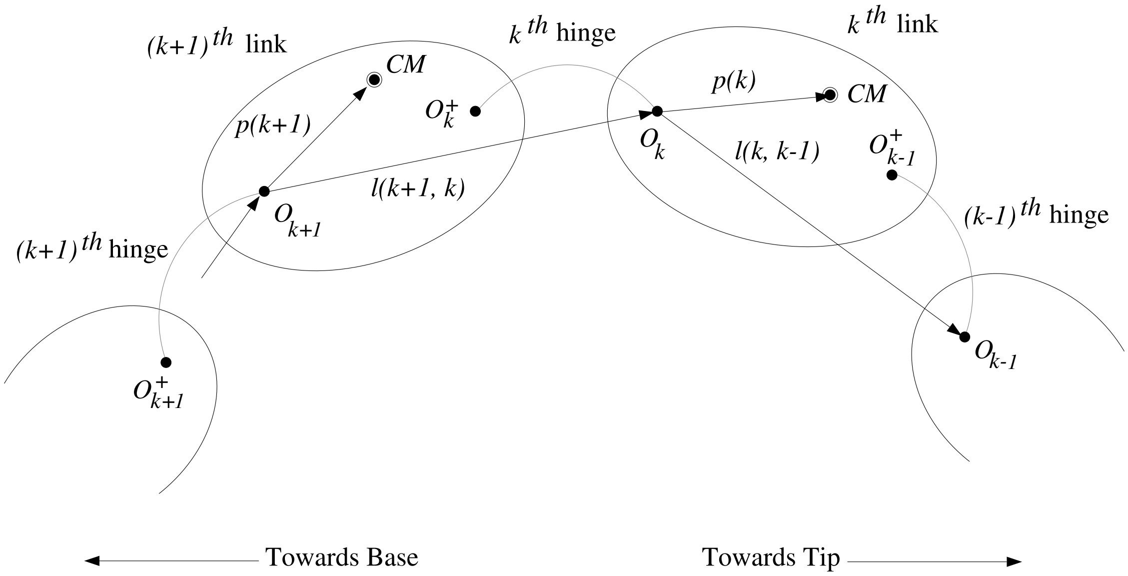

Figure 1: Illustration of the links and hinges in a manipulator

designate link n as the manipulator base–body. The inertial frame is referred to as "link $(n+1)$". The k^{th} hinge connects the $(k+1)^{th}$ and k^{th} links. Associated with the k^{th} hinge are two co-located frames, $\mathcal{O}_k^+$ and $\mathcal{O}_k$ which are attached to the the $(k+1)^{th}$ and k^{th} links respectively. The motion of the k^{th} hinge is defined by the motion of frame $\mathcal{O}_k$ with respect to frame $\mathcal{O}_k^+$. Resulting from the definition of these hinge frames, the k^{th} link has the two frames $\mathcal{O}_k$ and $\mathcal{O}_{k-1}^+$ attached to it. We choose frame $\mathcal{O}_k$ which is on the inboard side, to be the reference body frame for the k^{th} link.

The n^{th} hinge connects the base–body to the inertial frame. For a terrestrial manipulator, this hinge has less than 6 degrees of freedom, while for a free–flying manipulator this hinge has full 6 orientation and positional degrees of freedom. In general the k^{th} hinge has $r(k)$ degrees of freedom, where $1 \leq r(k) \leq 6$, and we denote its vector of generalized coordinates by $\boldsymbol{\theta}(k)$. For simplicity, and without any loss in generality, we assume that the number of generalized velocities for the hinge is also $r(k)$, i.e., there are no local nonholonomic constraints on the motion of the hinge. The vector of generalized velocities for the k^{th} hinge is $\boldsymbol{\beta}(k) \in \mathcal{R}^{r(k)}$. The choice of the hinge angle rates, $\dot{\boldsymbol{\theta}}(k)$, for the generalized velocities, $\boldsymbol{\beta}(k)$, is often an obvious and convenient choice. However, when the number of hinge degrees of freedom is larger than 1, alternative choices are often preferred since they simplify and decouple the kinematic and dynamic parts of the equations of motion. This is true in particular for 6 degree of freedom hinges. Three of the 6 degrees of freedom are orientation degrees of freedom, and it is well known that at least four generalized coordinates (such as for a quaternion representations) are needed to provide a global, singularity free representation. Thus $\boldsymbol{\theta}(n)$ is seven dimensional for free-flying manipulators. Moreover, for this type of hinge, the use of "derivatives of quasi-coordinates" such as the angular velocity of the base-body for the orientation generalized velocities is preferred since it simplifies the equations of motion. The use of non-integrable velocity coordinates such as the angular velocity implies that these velocity coordinates must be transformed into integrable variables prior to any numerical integration. We omit the details here since these issues are well known and discussed in standard texts. The overall number of degrees of freedom for the manipulator is given by $\mathcal{N} = \sum_{k=1}^{n} r(k)$.

The *spatial velocity* $V(k)$, of the k^{th} body frame $\mathcal{O}_k$, is defined as $V(k) = col[\omega(k), v(k)] \in \mathcal{R}^6$, with $\omega(k)$ and $v(k)$ denoting the angular and linear velocities of frame $\mathcal{O}_k$. We use coordinate-free notation for the various velocity, force etc. quantities defined in this paper. The relative spatial velocity across the k^{th} hinge is given by $\boldsymbol{H}^*(k)\boldsymbol{\beta}(k)$ where $\boldsymbol{H}^*(k) \in \mathcal{R}^{6 \times r(k)}$ is the *hinge map matrix* for the hinge. The *spatial force* of interaction at the k^{th} hinge is denoted $f(k) = col[N(k), F(k)] \in \mathcal{R}^6$, with $N(k)$ and $F(k)$ denoting the moment and force components respectively. The *spatial inertia* $\boldsymbol{M}(k)$ of the k^{th} link about frame $\mathcal{O}_k$ is defined as

$$\boldsymbol{M}(k) = \begin{pmatrix} \boldsymbol{\mathcal{J}}(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)\boldsymbol{I}_3 \end{pmatrix} \in \mathcal{R}^{6 \times 6}$$

where $m(k)$ is the link's mass, $p(k) \in \mathcal{R}^3$ is the vector from $\mathcal{O}_k$ to the link's center of mass, and $\boldsymbol{\mathcal{J}}(k) \in \mathcal{R}^{3 \times 3}$ is the rotational inertia of the link about $\mathcal{O}_k$. Here $\tilde{x} \in \mathcal{R}^{3 \times 3}$ denotes the skew–symmetric cross–product matrix associated with the vector x.

With $V(k)$ denoting the spatial velocity, $\alpha(k)$ the *spatial acceleration*, $f(k)$ the spatial force and $\boldsymbol{T}(k)$ the hinge generalized force at $\mathcal{O}_k$ for the k^{th} link, the following Newton–Euler recursive equations [7, 10] describe the equations of motion as well as an $O(\mathcal{N})$ inverse dynamics algorithm for the serial manipulator:

Algorithm 2.1 *Newton-Euler Inverse Dynamics Algorithm*

$$\left\{\begin{array}{rcl} & V(n+1)=0, \quad \alpha(n+1)=0 & \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{n}\cdots\mathbf{1} \\ V(k) & = & \boldsymbol{\phi}^*(k+1,k)V(k+1)+\boldsymbol{H}^*(k)\boldsymbol{\beta}(k) \\ \alpha(k) & = & \boldsymbol{\phi}^*(k+1,k)\alpha(k+1)+\boldsymbol{H}^*(k)\dot{\boldsymbol{\beta}}(k)+a(k) \\ \textbf{end loop} & & \end{array}\right.$$

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

where $b(k)$ denotes the velocity dependent *gyroscopic force* term

$$b(k) \triangleq \begin{pmatrix} \tilde{\omega}(k) & \tilde{v}(k) \\ 0 & \tilde{\omega}(k) \end{pmatrix} \boldsymbol{M}(k)V(k) \tag{2}$$

and $a(k)$ denotes the velocity dependent *Coriolis acceleration* term whose value depends upon the type of the hinge. Expressions for $a(k)$ for the case of 1 degree of freedom rotational and prismatic hinges respectively are as follows:

$$\begin{pmatrix} \tilde{\Delta}_\omega(k)\omega(k) \\ \tilde{\Delta}_\omega(k)v(k) \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 0 \\ \omega(k)\tilde{\Delta}_v(k) \end{pmatrix}$$

where $\Delta_\omega(k)$ and $\Delta_v(k)$ denote the relative angular and linear velocity across the hinge. $\boldsymbol{\phi}(k,k-1)$ denotes the *spatial transformation operator* from $\mathcal{O}_{k-1}$ to $\mathcal{O}_k$ and is given by

$$\boldsymbol{\phi}(k,k-1) \triangleq \begin{pmatrix} \boldsymbol{I}_3 & \tilde{l}(k,k-1) \\ \mathbf{0} & \boldsymbol{I}_3 \end{pmatrix} \in \mathcal{R}^{6\times 6}$$

where $l(k,k-1)$ is the vector from frame $\mathcal{O}_k$ to frame $\mathcal{O}_{k-1}$. Though not shown explicitly, external forces on any link in the manipulator are handled by adding their effect to the $b(\cdot)$ vector for the link.

Spatial operators [7] lead to compact expressions for the equations of motion and other key dynamical quantities. The vector $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}^*(1), \cdots \boldsymbol{\theta}^*(n)]^* \in \mathcal{R}^{\mathcal{N}}$ denotes the vector of generalized coordinates for the manipulator. Similarly, we define the vectors of generalized velocities $\boldsymbol{\beta} \in \mathcal{R}^{\mathcal{N}}$ and generalized (hinge) forces $\boldsymbol{T} \in \mathcal{R}^{\mathcal{N}}$ for the manipulator. The vector of spatial velocities V is defined as $V \triangleq [V^*(1) \cdots V^*(n)]^* \in \mathcal{R}^{6n}$. The vector of spatial accelerations is denoted $\alpha \in \mathcal{R}^{6n}$, that of the Coriolis accelerations by $a \in \mathcal{R}^{6n}$, the link gyroscopic forces by $b \in \mathcal{R}^{6n}$, and the link interaction spatial forces by $f \in \mathcal{R}^{6n}$. The equations of motion for the serial manipulator can be written in operator form as follows [7]:

$$V = \boldsymbol{\phi}^* \boldsymbol{H}^* \boldsymbol{\beta} \tag{3a}$$
$$\alpha = \boldsymbol{\phi}^* [\boldsymbol{H}^* \dot{\boldsymbol{\beta}} + a] \tag{3b}$$
$$f = \boldsymbol{\phi}[\boldsymbol{M}\alpha + b] \tag{3c}$$
$$\boldsymbol{T} = \boldsymbol{H} f = \boldsymbol{\mathcal{M}} \dot{\boldsymbol{\beta}} + \boldsymbol{\mathcal{C}} \tag{3d}$$

where

$$\boldsymbol{\mathcal{M}} \stackrel{\triangle}{=} \boldsymbol{H}\boldsymbol{\phi}\boldsymbol{M}\boldsymbol{\phi}^*\boldsymbol{H}^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}} \tag{4a}$$
$$\boldsymbol{\mathcal{C}} \stackrel{\triangle}{=} \boldsymbol{H}\boldsymbol{\phi}[\boldsymbol{M}\boldsymbol{\phi}^* a + b] \in \mathcal{R}^{\mathcal{N}} \tag{4b}$$

and $\boldsymbol{H} \stackrel{\triangle}{=} \text{diag}\left\{\boldsymbol{H}(k)\right\} \in \mathcal{R}^{\mathcal{N}\times 6n}$, $\boldsymbol{M} \stackrel{\triangle}{=} \text{diag}\left\{\boldsymbol{M}(k)\right\} \in \mathcal{R}^{6n\times 6n}$,

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

$$\boldsymbol{\phi} \stackrel{\triangle}{=} (\boldsymbol{I}_{6n} - \boldsymbol{\mathcal{E}_\phi})^{-1} = \begin{pmatrix} \boldsymbol{I}_6 & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{\phi}(2,1) & \boldsymbol{I}_6 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}(n,1) & \boldsymbol{\phi}(n,2) & \dots & \boldsymbol{I}_6 \end{pmatrix} \in \mathcal{R}^{6n\times 6n} \tag{5}$$

with

$$\boldsymbol{\phi}(i,j) \stackrel{\triangle}{=} \boldsymbol{\phi}(i,i-1) \cdots \boldsymbol{\phi}(j+1,j) \in \mathcal{R}^{6\times 6} \quad \text{for} \quad i > j$$

$\boldsymbol{\mathcal{M}}(\boldsymbol{\theta})$ is the *mass matrix* of the manipulator and the vector $\boldsymbol{\mathcal{C}}(\boldsymbol{\theta}, \boldsymbol{\beta})$ contains the velocity dependent Coriolis and gyroscopic hinge forces. (4a) is referred to as the *Newton–Euler Operator Factorization* of the mass matrix $\boldsymbol{\mathcal{M}}$ [7].

2.1 Spatial Operator Factorization of $\boldsymbol{\mathcal{M}}^{-1}$

As discussed in reference [7], operator factorization and inversion techniques can be used to obtain a closed form spatial operator expression for $\boldsymbol{\mathcal{M}}^{-1}$. First, we define the *articulated body inertia* quantities $\boldsymbol{P}(.), \boldsymbol{D}(.), \boldsymbol{G}(.), \boldsymbol{K}(.), \overline{\boldsymbol{\tau}}(.), \boldsymbol{P}^+(.)$ and $\boldsymbol{\psi}(.,.)$ for the manipulator links using the following tip–to–base recursive algorithm [6, 7]:

Algorithm 2.2 Computation of Articulated Body Inertias

$$
\left\{
\begin{array}{rcl}
\boldsymbol{P}^{+}(0) = \boldsymbol{0} & & \\
\textbf{for } \boldsymbol{k} \;=\; \boldsymbol{1 \cdots n} & & \\
\boldsymbol{P}(k) & = & \boldsymbol{\phi}(k,k-1)\boldsymbol{P}^{+}(k-1)\boldsymbol{\phi}^{*}(k,k-1) + \boldsymbol{M}(k) \\
\boldsymbol{D}(k) & = & \boldsymbol{H}(k)\boldsymbol{P}(k)\boldsymbol{H}^{*}(k) \\
\boldsymbol{G}(k) & = & \boldsymbol{P}(k)\boldsymbol{H}^{*}(k)\boldsymbol{D}^{-1}(k) \\
\boldsymbol{K}(k+1,k) & = & \boldsymbol{\phi}(k+1,k)\boldsymbol{G}(k) \\
\overline{\boldsymbol{\tau}}(k) & = & \boldsymbol{I}_6 - \boldsymbol{G}(k)\boldsymbol{H}(k) \\
\boldsymbol{P}^{+}(k) & = & \overline{\boldsymbol{\tau}}(k)\boldsymbol{P}(k)\overline{\boldsymbol{\tau}}^{*}(k) = \overline{\boldsymbol{\tau}}(k)\boldsymbol{P}(k) \\
\boldsymbol{\psi}(k+1,k) & = & \boldsymbol{\phi}(k+1,k)\overline{\boldsymbol{\tau}}(k) \\
\textbf{end loop} & &
\end{array}
\right.
\tag{6}
$$

The operator $\boldsymbol{P} \in \mathcal{R}^{6n\times 6n}$ is defined as a block diagonal matrix with the k^{th} diagonal element being $\boldsymbol{P}(k) \in \mathcal{R}^{6\times 6}$. The quantities in (6) are used to define the following spatial operators:

$$
\begin{array}{rcl}
\boldsymbol{D} & \stackrel{\triangle}{=} & \boldsymbol{HPH}^{*} \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}} \\
\boldsymbol{G} & \stackrel{\triangle}{=} & \boldsymbol{PH}^{*}\boldsymbol{D}^{-1} \in \mathcal{R}^{6n\times\mathcal{N}} \\
\boldsymbol{K} & \stackrel{\triangle}{=} & \boldsymbol{\mathcal{E}_{\phi}G} \in \mathcal{R}^{6n\times\mathcal{N}} \\
\overline{\boldsymbol{\tau}} & \stackrel{\triangle}{=} & \boldsymbol{I} - \boldsymbol{GH} \in \mathcal{R}^{6n\times 6n} \\
\boldsymbol{P}^{+} & \stackrel{\triangle}{=} & \overline{\boldsymbol{\tau}}\boldsymbol{P}\overline{\boldsymbol{\tau}}^{*} = \overline{\boldsymbol{\tau}}\boldsymbol{P} \in \mathcal{R}^{6n\times 6n} \\
\boldsymbol{\mathcal{E}_{\psi}} & \stackrel{\triangle}{=} & \boldsymbol{\mathcal{E}_{\phi}}\overline{\boldsymbol{\tau}} \in \mathcal{R}^{6n\times 6n}
\end{array}
\tag{7}
$$

The operators $\boldsymbol{D}, \boldsymbol{G}$ and $\overline{\boldsymbol{\tau}}$ are all block diagonal. Even though $\boldsymbol{K}$ and $\boldsymbol{\mathcal{E}_{\psi}}$ are not block diagonal matrices, their only nonzero block elements are the elements $\boldsymbol{K}(k,k-1)$'s and $\boldsymbol{\psi}(k,k-1)$'s respectively along the first subdiagonal. It is easy to verify from (6) that $\boldsymbol{P}$ satisfies the Riccati equation

$$\boldsymbol{M} = \boldsymbol{P} - \boldsymbol{\mathcal{E}_{\psi}P\mathcal{E}_{\psi}^{*}} = \boldsymbol{P} - \boldsymbol{\mathcal{E}_{\phi}P\mathcal{E}_{\psi}^{*}} \tag{8}$$

Now define the lower–triangular operator $\boldsymbol{\psi} \in \mathcal{R}^{6n\times 6n}$ as

$$\boldsymbol{\psi} \stackrel{\triangle}{=} (\boldsymbol{I} - \boldsymbol{\mathcal{E}_{\psi}})^{-1} \tag{9}$$

Its block elements $\boldsymbol{\psi}(i,j) \in \mathcal{R}^{6\times 6}$ are as follows:

$$
\boldsymbol{\psi}(i,j) \stackrel{\triangle}{=} \left\{
\begin{array}{lll}
\boldsymbol{\psi}(i,i-1) \;\cdots\; \boldsymbol{\psi}(j+1,j) & \text{for} & i > j \\
\boldsymbol{I}_6 & \text{for} & i = j \\
\boldsymbol{0} & \text{for} & i < j
\end{array}
\right.
$$

The structure of the operators $\boldsymbol{\psi}$ and $\boldsymbol{\mathcal{E}_{\psi}}$ is identical to that of the operators $\boldsymbol{\phi}$ and $\boldsymbol{\mathcal{E}_{\phi}}$ except that the elements are now $\boldsymbol{\psi}(i,j)$ rather than $\boldsymbol{\phi}(i,j)$.

Lemma 2.1 below describes an alternative (to (4a)) operator factorization of $\boldsymbol{\mathcal{M}}$ as well as an expression for its inverse [7,8].

Lemma 2.1 *The* **Innovations Operator Factorization** *of the mass matrix $\boldsymbol{\mathcal{M}}$ and the operator expression for its inverse are as follows:*

$$\begin{aligned} \boldsymbol{\mathcal{M}} &= [\boldsymbol{I} + \boldsymbol{H\phi K}]\boldsymbol{D}[\boldsymbol{I} + \boldsymbol{H\phi K}]^* && \text{(10a)} \\ [\boldsymbol{I} + \boldsymbol{H\phi K}]^{-1} &= [\boldsymbol{I} - \boldsymbol{H\psi K}] && \text{(10b)} \\ \boldsymbol{\mathcal{M}}^{-1} &= [\boldsymbol{I} - \boldsymbol{H\psi K}]^* \boldsymbol{D}^{-1}[\boldsymbol{I} - \boldsymbol{H\psi K}] && \text{(10c)} \end{aligned}$$

■

The factor $[\boldsymbol{I} + \boldsymbol{H\phi K}] \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$ is square while the factor $\boldsymbol{D}$ is block diagonal. Thus, the factorization in Lemma 2.1 can also be regarded as a closed–form LDL^* factorization of $\boldsymbol{\mathcal{M}}$. The closed form operator expression for the inverse of the factor $[\boldsymbol{I} + \boldsymbol{H\phi K}]$ is given by (10b). It leads to the closed form operator expression for the inverse of the mass matrix $\boldsymbol{\mathcal{M}}$ in (10c). This factorization can be regarded as a closed–form L^*DL factorization of $\boldsymbol{\mathcal{M}}^{-1}$.

2.2 Articulated Body Forward Dynamics Algorithm

Using (10c) in (3d) and operator identities described in reference [8], we obtain the following operator expression for the generalized accelerations vector $\dot{\boldsymbol{\beta}}$:

$$\dot{\boldsymbol{\beta}} = [\boldsymbol{I} - \boldsymbol{H\psi K}]^* \boldsymbol{D}^{-1}[\boldsymbol{T} - \boldsymbol{H\psi}(\boldsymbol{KT} + \boldsymbol{P}a + b)] - \boldsymbol{K}^*\boldsymbol{\psi}^*\boldsymbol{H}^*a \qquad (11)$$

This expression forms the basis for the $O(\mathcal{N})$ *articulated body inertia forward dynamics algorithm* [6,7] for manipulators. The structure of the algorithm is more easily seen by first decomposing (11) into the following sequence of expressions:

$$\begin{aligned} z &= \boldsymbol{\mathcal{E}_\phi} z^+ + \boldsymbol{P}a + b && \text{(12a)} \\ z^+ &= z + \boldsymbol{G\epsilon} && \text{(12b)} \\ \boldsymbol{\epsilon} &= \boldsymbol{T} - \boldsymbol{H}z && \text{(12c)} \\ \boldsymbol{\nu} &= \boldsymbol{D}^{-1}\boldsymbol{\epsilon} && \text{(12d)} \\ \alpha^+ &= \boldsymbol{\mathcal{E}}^*_{\boldsymbol{\phi}}\alpha && \text{(12e)} \\ \dot{\boldsymbol{\beta}} &= \boldsymbol{\nu} - \boldsymbol{G}^*\alpha^+ && \text{(12f)} \\ \alpha &= \alpha^+ + \boldsymbol{H}^*\dot{\boldsymbol{\beta}} + a && \text{(12g)} \end{aligned}$$

The expressions in (12) map into the following computational algorithm:

Algorithm 2.3 *Articulated Body Inertia Forward Dynamics Algorithm*

$$\left\{\begin{array}{rcl} & z(0) = 0 & \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{1 \cdots n} \\ z(k) & = & \boldsymbol{\phi}(k,k-1)z^{+}(k-1) + b(k) + \boldsymbol{P}(k)a(k) \\ \boldsymbol{\epsilon}(k) & = & \boldsymbol{T}(k) - \boldsymbol{H}(k)z(k) \\ z^{+}(k) & = & z(k) + \boldsymbol{G}(k)\boldsymbol{\epsilon}(k) \\ \boldsymbol{\nu}(k) & = & \boldsymbol{D}^{-1}\boldsymbol{\epsilon}(k) \\ \textbf{end loop} & & \end{array}\right. \tag{13a}$$

$$\left\{\begin{array}{rcl} & \alpha^{+}(n+1) = 0 & \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{n \cdots 1} \\ \alpha^{+}(k) & = & \boldsymbol{\phi}^{*}(k+1,k)\alpha(k+1) \\ \dot{\boldsymbol{\beta}}(k) & = & \boldsymbol{\nu}(k) - \boldsymbol{G}^{*}(k)\alpha^{+}(k) \\ \alpha(k) & = & \alpha^{+}(k) + \boldsymbol{H}^{*}(k)\dot{\boldsymbol{\beta}}(k) + a(k) \\ \textbf{end loop} & & \end{array}\right. \tag{13b}$$

The overall steps in this $O(\mathcal{N})$*articulated body inertia forward dynamics algorithm* are as follows:

1. Use the first base–to–tip recursion in Algorithm 2.1 to compute the spatial velocities $V(k)$, and the nonlinear velocity dependent terms $a(k)$ and $b(k)$ for each of the links.

2. Use the tip–to–base recursion in Algorithm 2.2 to compute the articulated body inertia quantities $\boldsymbol{P}(k)$ etc.

3. Use the first tip–to–base recursion in Algorithm 2.3 to compute the residual force quantities $z(k)$ etc.

4. Use the subsequent base–to–tip recursion in Algorithm 2.3 to compute the link and joint accelerations $\alpha(k)$ and $\dot{\boldsymbol{\beta}}(k)$ for all the links and hinges.

3 Free–Flying Manipulators

We now specialize the manipulator model described above to the case of a serial free-flying manipulators shown in Figure 2. For free-flying manipulators, the n^{th} hinge (between the base–body and the inertial frame) has 6 degrees of freedom, and the components of the generalized velocities vector, $\boldsymbol{\beta}(n)$, for this hinge are chosen as the components of the 6–dimensional spatial velocity of the base–body, $V(n)$, represented in the body frame $\mathcal{O}(n)$. The hinge map matrix $\boldsymbol{H}(n)$ for this hinge is the 6×6 identity matrix, i.e. $\boldsymbol{H}(n) = \boldsymbol{I}_6$.

The operator factorization and inversion result as well as the $O(\mathcal{N})$ articulated body inertia forward dynamics algorithm described earlier for general manipulators extend to free-flying

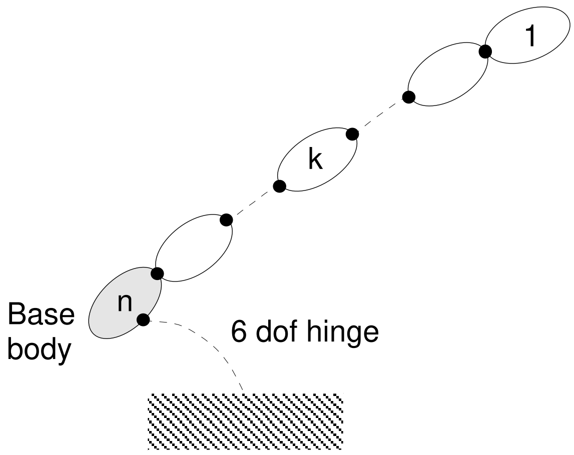

Figure 2: A serial free-flying manipulator with a 6 degree of freedom hinge at the base–body

manipulators as well. Since $\boldsymbol{H}(n) = \boldsymbol{I}_6$, the n^{th} recursion step (for the base–body) in Algorithm 2.2 simplifies to

$$\boldsymbol{D}(n) = \boldsymbol{P}(n), \quad \boldsymbol{G}(n) = \boldsymbol{I}_6, \quad \boldsymbol{\tau}(n) = \boldsymbol{I}_6, \quad \overline{\boldsymbol{\tau}}(n) = \boldsymbol{0} \tag{14}$$

Also, the residual force computations in Algorithm 2.3 simplifies to

$$\boldsymbol{\epsilon}(n) = \boldsymbol{T}(n) - z(n), \quad \dot{\boldsymbol{\beta}}(n) = \boldsymbol{\nu}(n) = \boldsymbol{P}^{-1}(n)\boldsymbol{\epsilon}(n), \quad \alpha(n) = \boldsymbol{\nu}(n) + a(n) \tag{15}$$

For a terrestrial manipulator, the choice of a link as the base–body is uniquely the link attached to the ground. For free-flying manipulators, since none of the links are attached to the ground, the choice of a link as the base-body is in principle an arbitrary choice. In practice, this choice is made based upon operational convenience.

It is our contention here that the freedom of choice in designating the base-body link is an inherent *base-invariance symmetry* of free-flying manipulators which has interesting analytical and algorithmic consequences explored in the remainder of this paper. In Section 2, link n was designated as the base–body link for the manipulator, and this choice dictated the specific definition of the generalized coordinates, the expressions for the equations of motion, the definitions of the spatial operators and also the structure of the inverse and forward dynamics computational algorithms. Appendix A establishes rigorously that all of the modeling and operator factorization results in Section 2 carry over completely even when links other than link n are designated as the base–body. In particular, Appendix A describes the structure of the various dynamics quantities and operators associated with different choices of the base-body. It also describes the transformations required to convert operators associated with one base-body choice to those corresponding to a different choice.

The models and algorithms associated with the choice of *extremal* links as base-bodies are *fundamental* models, in the sense that algorithms associated with the choice of interior links as base-bodies can be recovered from and rely upon computations for the fundamental models. We have already seen one fundamental model in Section 2 where the extremal link n was designated the base-body and we refer to this model as the *regular model*. In the following section we look at the other fundamental model for a serial free-flying manipulator where link 1 is designated the base-body for the manipulator. We refer to this new model as being the *dual* of the *regular* model.

We henceforth use the subscript "p" for the articulated body inertia and residual force quantities defined in Section 2 for the regular model. For quantities associated with the dual model (with link 1 as base–body) we use the subscript "s". Thus the vectors $\boldsymbol{\beta}_p$ and $\boldsymbol{\beta}_s$ denote the generalized velocity vectors for the regular and dual manipulator models respectively.

3.1 Dynamics Algorithm with Link 1 as Base Body

Figure 3 shows the configuration of a serial free-flying manipulator with link 1 chosen as the base–body. As discussed in Appendix A, the six components of the base–body's spatial velocity

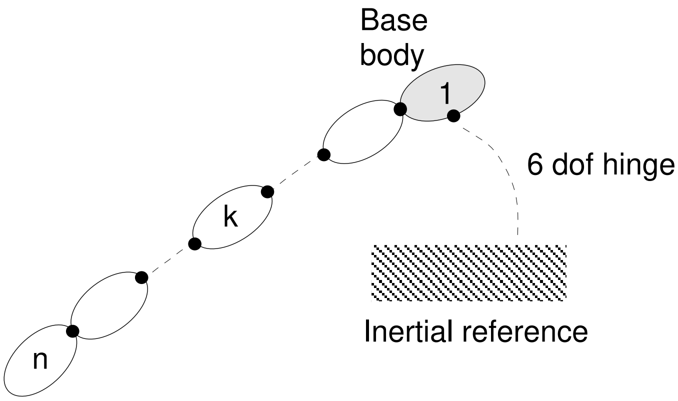

Figure 3: A serial free-flying manipulator with the outer most link as base–body

vector form part of the generalized velocity coordinates for the manipulator. Thus, when the base–body is moved from link n to link 1, the components of $\boldsymbol{\beta}(n)$ in $\boldsymbol{\beta}_p$ are replaced by the components $\boldsymbol{\beta}(0) \stackrel{\triangle}{=} V(1)$ to obtain the new generalized velocity coordinates vector $\boldsymbol{\beta}_s$. Note that the definition of the $\boldsymbol{H}_s$, $\boldsymbol{\phi}_s$ etc. operators and the mass matrix $\boldsymbol{\mathcal{M}}_s$ all change in the dual formulation, Nevertheless, as discussed in Appendix A, the operator formulation and operator factorization and inversion results for the new mass matrix still carry through. Consequently, there is a corresponding version of the articulated body inertia forward dynamics algorithm (Algorithm

2.3) for this model as well.

One important difference between the dual articulated body inertia algorithm and Algorithm 2.3 is that the roles of the base and tip links are reversed. Thus tip–to–base (base–to–tip) recursions now proceed from link n to 1 (link 1 to n) rather than in the opposite direction. We designate the new dual articulated body inertia by the symbol $\boldsymbol{S}^+(k)$, and the other dual quantities by the subscript s, i.e. $\boldsymbol{D}_s$, $\boldsymbol{G}_s$, $\boldsymbol{\tau}_s$ etc. The quantity corresponding to $\boldsymbol{P}^+$ in the dual formulation is designated $\boldsymbol{S}$ and is given by by the expression

$$\boldsymbol{S} \stackrel{\triangle}{=} \overline{\boldsymbol{\tau}}_s \boldsymbol{S}^+$$

Note that the reversal in direction also reverses the sense of orientation of the internal hinge axes, and therefore all of the hinge map matrices, $\boldsymbol{H}(.)$, reverse sign. The recursions corresponding to (6) for the dual articulated body inertias are as follows:

Algorithm 3.1 <u>*Computation of Dual Articulated Body Inertias*</u>

$$\left\{ \begin{array}{rcl} & \boldsymbol{S}(n) = \boldsymbol{0} & \\ \textbf{for } \boldsymbol{k} & \boldsymbol{=} & \boldsymbol{n-1 \cdots 0} \\ \boldsymbol{S}^+(k) & = & \boldsymbol{\phi}(k,k+1)[\boldsymbol{S}(k+1) + \boldsymbol{M}(k+1)]\boldsymbol{\phi}^*(k,k+1) \\ \boldsymbol{D}_s(k) & = & \boldsymbol{H}(k)\boldsymbol{S}^+(k)\boldsymbol{H}^*(k) \\ \boldsymbol{G}_s(k) & = & -\boldsymbol{S}^+(k)\boldsymbol{H}^*(k)\boldsymbol{D}_s^{-1}(k) \\ \boldsymbol{K}_s(k-1,k) & = & \boldsymbol{\phi}(k-1,k)\boldsymbol{G}_s(k) \\ \overline{\boldsymbol{\tau}}_s(k) & = & \boldsymbol{I}_6 + \boldsymbol{G}_s(k)\boldsymbol{H}(k) \\ \boldsymbol{S}(k) & = & \overline{\boldsymbol{\tau}}_s(k)\boldsymbol{S}^+(k) \\ \boldsymbol{\psi}_s(k-1,k) & = & \boldsymbol{\phi}(k-1,k)\overline{\boldsymbol{\tau}}_s(k) \\ \textbf{end loop} & & \end{array} \right. \tag{16}$$

The quantity $\boldsymbol{\phi}(k-1,k)$ is defined as the inverse of $\boldsymbol{\phi}(k,k-1)$, that is

$$\boldsymbol{\phi}(k-1,k) \stackrel{\triangle}{=} \boldsymbol{\phi}^{-1}(k,k-1) = \begin{pmatrix} \boldsymbol{I}_3 & -\tilde{l}(k,k-1) \\ \boldsymbol{0} & \boldsymbol{I}_3 \end{pmatrix} \tag{17}$$

Similarly, an algorithm corresponding to the residual–forces recursion in (13) can be developed for the dual model and forms the remaining part of the dual articulated body inertia forward dynamics algorithm and is as follows:

Algorithm 3.2 <u>*Computation of Dual Residual Forces*</u>

$$\left\{\begin{array}{rcl} & z_s(n) = 0 & \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{n-1\cdots 0} \\ z_s^+(k) & = & \boldsymbol{\phi}(k,k+1)\left[z_s(k+1)+b(k+1)+\{\boldsymbol{S}(k+1)+\boldsymbol{M}(k+1)\}a_s(k+1)\right] \\ \boldsymbol{\epsilon}_s(k) & = & \boldsymbol{T}(k)+\boldsymbol{H}(k)z_s(k) \\ z_s(k) & = & z_s^+(k)+\boldsymbol{G}_s(k)\boldsymbol{\epsilon}_s(k) \\ \boldsymbol{\nu}_s(k) & = & \boldsymbol{D}^{-1}\boldsymbol{\epsilon}_s(k) \\ \textbf{end loop} & & \end{array}\right. \tag{18a}$$

$$\left\{\begin{array}{rcl} & \alpha(0) = 0 & \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{0\cdots n-1} \\ \alpha(k) & = & \boldsymbol{\phi}^*(k-1,k)\alpha^+(k-1) \\ \dot{\boldsymbol{\beta}}(k) & = & \boldsymbol{\nu}_s(k)-\boldsymbol{G}_s^*(k)\alpha(k) \\ \alpha^+(k) & = & \alpha(k)-\boldsymbol{H}^*(k)\dot{\boldsymbol{\beta}}(k)+a_s(k) \\ \textbf{end loop} & & \end{array}\right. \tag{18b}$$

The dual Coriolis acceleration $a_s(k)$ is defined as

$$a_s(k) \stackrel{\triangle}{=} -\boldsymbol{\phi}^*(k+1,k)a_p(k+1) \tag{19}$$

3.2 Direct Computation of Link Spatial Acceleration

An important relationship which relates the inter–link forces, $f(k)$, the residual forces $z_p(k)$, and the link spatial accelerations $\alpha(k)$ is described by the following lemma.

Lemma 3.1 *We have*

$$f \;=\; \boldsymbol{P}^+\alpha^+ + z_p^+ = \boldsymbol{P}(\alpha - a_p) + z_p \tag{20}$$

Proof: *From* (12) *it follows that*

$$\begin{aligned} \alpha \;&\stackrel{12f,12g}{=}\; \alpha^+ + \boldsymbol{H}_p^*\boldsymbol{\nu}_p - \boldsymbol{\tau}_p^*\alpha^+ + a_p \;\stackrel{12e}{=}\; \boldsymbol{\mathcal{E}}_{\boldsymbol{\psi}_p}^*\alpha_p + \boldsymbol{H}_p^*\boldsymbol{\nu}_p + a_p \\ &\stackrel{9}{=}\; \boldsymbol{\psi}_p^*(\boldsymbol{H}_p^*\boldsymbol{\nu}_p + a_p) \end{aligned} \tag{21}$$

Thus

$$\overline{\boldsymbol{\tau}}_p^*\alpha_p^+ \;\stackrel{12e}{=}\; \boldsymbol{\mathcal{E}}_{\boldsymbol{\psi}_p}^*\alpha_p \;\stackrel{21}{=}\; \tilde{\boldsymbol{\psi}}_p^*(\boldsymbol{H}^*\boldsymbol{\nu}_p + a_p) \tag{22}$$

where $\tilde{\boldsymbol{\psi}} \stackrel{\triangle}{=} \boldsymbol{\mathcal{E}}_{\boldsymbol{\psi}}\boldsymbol{\psi} = \boldsymbol{\psi} - I$. *Also,*

$$z_p^+ \stackrel{12a,12b}{=} \boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_p}z_p^+ + \boldsymbol{G}_p\boldsymbol{\epsilon}_p + \boldsymbol{P}a_p + b_p \;\stackrel{5}{=}\; \boldsymbol{\phi}_p(\boldsymbol{G}_p\boldsymbol{\epsilon}_p + \boldsymbol{P}a_p + b_p) \tag{23}$$

Therefore,

$$\begin{aligned} f &\stackrel{3c}{=} \boldsymbol{\phi}_p(\boldsymbol{M}_p\alpha + b) \\ &\stackrel{21}{=} \boldsymbol{\phi}_p\boldsymbol{M}\boldsymbol{\psi}_p^*(\boldsymbol{H}_p^*\boldsymbol{\nu}_p + a_p) + \boldsymbol{\phi}_p b \\ &\stackrel{8}{=} \left[\boldsymbol{\phi}_p\boldsymbol{P} + \boldsymbol{P}\tilde{\boldsymbol{\psi}}_p^*\right](\boldsymbol{H}_p^*\boldsymbol{\nu}_p + a_p) + \boldsymbol{\phi}_p b \\ &\stackrel{22}{=} \boldsymbol{P}^+\alpha^+ + \boldsymbol{\phi}_p\boldsymbol{P}(\boldsymbol{H}_p^*\boldsymbol{\nu}_p + a_p) + \boldsymbol{\phi}_p b \\ &\stackrel{7,12d,23}{=} \boldsymbol{P}^+\alpha^+ + z^+ \end{aligned}$$

The latter half of (20) *can be proved similarly.* ∎

The dual version of (20) is given by:

$$-f \quad = \quad \boldsymbol{S}^+(\alpha^+ - a_s) + z_s^+ = \boldsymbol{S}\alpha + z_s \tag{24}$$

The relationships in (20) and (24) give alternative ways of expressing the inter–link spatial force f using either the conventional or the dual articulated body inertia quantities. Combining these alternative expressions provides a direct method for computing the spatial accelerations of the links as described in the following lemma.

Lemma 3.2 *The spatial acceleration $\alpha(k)$ of the k^{th} link is given by:*

$$\begin{aligned} \alpha(k) &= -[\boldsymbol{P}(k) + \boldsymbol{S}(k)]^{-1}[z_p(k) + z_s(k) - \boldsymbol{P}(k)a_p(k)] \\ \alpha^+(k) &= -[\boldsymbol{P}^+(k) + \boldsymbol{S}^+(k)]^{-1}[z_p^+(k) + z_s^+(k) - \boldsymbol{S}^+(k)a_s(k)] \end{aligned} \tag{25}$$

Proof: *Combine together* (20) *and* (24)*, and eliminate f.* ∎

This result implies that the link accelerations α and α^+ can be obtained by combining together the results from the regular and dual articulated body inertia and residual force recursions. The hinge generalized acceleration is given by the following pair of expressions:

$$\dot{\boldsymbol{\beta}}(k) = \boldsymbol{\nu}_p(k) - \boldsymbol{G}_p^*(k)\alpha^+(k) = \boldsymbol{\nu}_s(k) - \boldsymbol{G}_s^*(k)\alpha(k) \tag{26}$$

3.3 Physical Interpretation

We now discuss the physical interpretation of (25). If we examine the (regular and dual) articulated body inertia forward dynamics algorithms, we see that a key part of the algorithms is the computation of the link articulated body inertias, culminating with the computation of the articulated body inertia for the base body. Once this inertia is obtained, the hinge acceleration for the base–body is computed, followed by the rest of the link and hinge accelerations.

As shown in Figure 4, let us now consider the intermediate link k as the base–body for the manipulator. For this choice, the components of the spatial velocity vector $V(k)$ for the k^{th} body

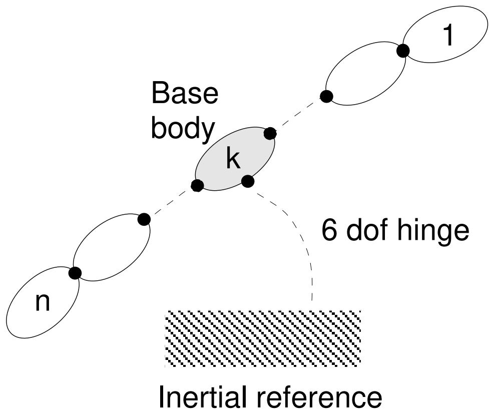

Figure 4: A serial free-flying manipulator with an intermediate link as base–body

provide six of the generalized velocity coordinates for the system. The manipulator now has a tree topology configuration, with two branches starting at the base. The extension of the articulated body inertia forward dynamics algorithm for such a tree topology configuration has been described in [11]. The basic algorithm remains the same as for a serial chain except that the recursions now have a scatter/gather structure. All recursions towards the base "gather" all the inputs from the incoming branches, while those proceeding outwards scatter their outputs along the outgoing branches. Thus the articulated body inertia algorithm now involves two separate articulated body inertia recursions starting off at link 1 and link n respectively and proceeding independently towards link k. The first recursion, from link 1 to link k, is identical to the recursion in Algorithm 2.2 for computing $\boldsymbol{P}(.)$'s for the regular model. On the other hand, the second recursion from link n to link k is identical to the recursion in Algorithm 3.1 for computing $\boldsymbol{S}(.)$'s for the dual model. These recursions come to a stop when the k^{th} link is reached.

The results from the two recursions are "gathered" together at the k^{th} link to form the quantity $\boldsymbol{P}(k) + \boldsymbol{S}(k)$ which is precisely the articulated body inertia of the whole manipulator as seen at frame $\mathcal{O}_k$ with the k^{th} link regarded as the base–body. Similarly $z_p(k) + z_s(k)$ is the residual force at frame $\mathcal{O}_k$ with the k^{th} link being the base–body. The corresponding relationships in (25) for the "+" quantities have a similar interpretation but with frame $\mathcal{O}_k^+$ serving as the reference frame for the k^{th} link. Thus according to (25), we can compute the spatial accelerations for the k^{th}

link by first making it the base–body, computing its articulated body inertia and residual forces and then using Lemma 3.2 to obtain its spatial acceleration. This observation forms the basis for the *base-invariant forward dynamics algorithm* described in the next section.

3.4 The Base–Invariant Forward Dynamics Algorithm

The previous section discussed how each link in the manipulator can in principle be regarded as a base–body. This arbitrariness in the choice of the base–body reflects the inherent base-invariance symmetry of free-flying manipulators. However, the articulated body inertia forward dynamics algorithms discussed earlier requires the designation of a specific link as the base–body for the manipulator, and this specific choice breaks the symmetry. We show here that breaking the symmetry is unnecessary and we reformulate the algorithm so as to preserve and take advantage of the symmetry. *The key to this is to simultaneously treat every link in the manipulator as a base–body.* With this in mind, and making use of Lemma 3.2, we propose the following new forward dynamics algorithm:

Algorithm 3.3 *Base–Invariant Forward Dynamics Algorithm*

1. *Use the first part of Algorithm 2.1 to compute the orientations, spatial velocities $V(k)$, and the Coriolis and gyroscopic terms a and b for all the links recursively.*
2. *(a) Compute the articulated body quantities $\boldsymbol{P}(.)$ etc. and the residual forces $z_p(.)$ in a recursion from link 1 to link n using Algorithms 2.2 and 2.3. (b) Simultaneously compute the dual articulated body quantities $\boldsymbol{S}^+(.)$ etc. and the dual residual forces $z_s^+(.)$ in a recursion from link n to link 1 using Algorithms 3.1 and 3.2. The recursions in (a) and (b) can be carried out independently.*
3. *For the k^{th} link, compute the link spatial acceleration $\alpha(k)$ using Lemma 3.2, and the hinge acceleration $\dot{\boldsymbol{\beta}}$ using (26). These computations can be carried out independently for each link.*
4. *For each link, integrate its hinge acceleration and velocity to update its hinge velocity and angle. Return to step 1.*

Step 4 above is necessary for numerical simulations and is used to propagate the state of the system in time. It is important to bear in mind that some of the components of $\boldsymbol{\beta}$ contain non-integrable coordinates, and these need to be converted (via kinematic transformations) into integrable quantities prior to integration. In the next section we discuss further simplification of the algorithm by choosing a non–minimal set of generalized coordinates for the free-flying manipulator.

3.4.1 Simplifications Using Non–Minimal Coordinates

So far we have used the hinge angles together with the six base–body positional and orientation coordinates as the generalized coordinates for the free-flying manipulator. These coordinates form a minimal set since their dimension is the same as the number of degrees of freedom for the system. We now look at an alternative and non–minimal choice of coordinates which simplifies the computations in Algorithm 3.3.

Recall that in Algorithm 3.3, the very first step in the dynamics computations involves a recursion to compute the orientations and spatial velocities of all the links using the generalized coordinates $\boldsymbol{\theta}$ and the generalized velocities $\boldsymbol{\beta}$. Step 3 computes the hinge accelerations from the link spatial accelerations and the last step updates the manipulator hinge coordinates and velocities using an integration routine. These steps perform transformations between the hinge coordinates and the spatial coordinates and can in fact be entirely dispensed with.

Towards this goal, we treat each link as an independent rigid body system in its own right. For each link, we choose its orientation and positional coordinates as its generalized coordinates, and its spatial velocity vector as its 6–dimensional generalized velocity coordinates. Taken together, this gives us a system with $6n$ generalized velocity coordinates. These coordinates are clearly non–minimal since only $\mathcal{N}$ of them are truly independent. However, with these coordinates, transformations between the hinge and spatial coordinate domains are unnecessary. The modified decoupled dynamics algorithm is as follows:

Algorithm 3.4 *Decoupled Forward Dynamics Algorithm with Non-Minimal Coordinates*

1. *Use each links' generalized velocities (i.e. spatial velocity $V(k)$) to compute the Coriolis and gyroscopic terms a and b for the link. These can be computed completely independently for each link.*

2. *(a) Compute the articulated body quantities $\boldsymbol{P}(.)$ etc. and the residual forces $z_p(.)$ in a recursion from link 1 to link n using Algorithms 2.2 and 2.3. (b) Simultaneously compute the dual articulated body quantities $\boldsymbol{S}^+(.)$ etc. and the dual residual forces $z_s(.)$ in a recursion from link n to link 1 using Algorithms 3.1 and 3.2. The recursions in (a) and (b) can be carried out independently.*

3. *For the k^{th} link, compute the link spatial acceleration $\alpha(k)$ using Lemma 3.2.*

4. *For each link, integrate its spatial acceleration $\alpha(k)$ and spatial velocity $V(k)$ to update its spatial velocity, position and orientation. Go back to step 1.*

The use of these non–minimal coordinates eliminates the need for the minimal $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ coordinates and hence dispenses with the kinematics computations in Algorithm 3.3 for obtaining the link spatial velocities and orientations. The price paid with the use of redundant coordinates is that the integration method now involves a differential–algebraic equation rather than an ordinary differential equation. The remarks following Algorithm 3.3 about the integration of non-integrable coordinates apply here as well.

3.4.2 Computational Issues

As is the case for the articulated body forward dynamics algorithm (Algorithm 2.3), the decoupled dynamics algorithm described in Algorithm 3.4 is also of $O(\mathcal{N})$ complexity. However, since the latter involves a pair of articulated body recursions, it is computationally more expensive. On the other hand, since many of the computations are decoupled and independent of each other, it is useful for parallel implementation. In Algorithm 3.4, the computations in step 1 can be carried out independently and in parallel for all the links. In step 2, the articulated body recursion in one direction is completely independent of the one in the opposite direction. Thus they can be computed in parallel. Using an architecture in which each link is assigned its own computational node, each link (node) receives the results of the articulated body recursions from its neighbors, updates its own articulated body inertias, and passes the results onto its neighbors. As in step 1, the computations in step 3 are independent from link to link. Thus each link computes its own spatial acceleration independent of the other nodes. Each node even has its own local integrator to update the state of its link.

3.4.3 Smoothing Interpretation of the Algorithm

As has been discussed in references [12, 13], the $O(\mathcal{N})$ articulated body inertia forward dynamics algorithm in Section 2.2 resembles fixed–interval optimal smoothing algorithms from optimal estimation theory. The underlying estimation problem consists of the computation of the optimal smoothed estimates of the states of a discrete time system driven by white noise over a finite interval. The smoothing algorithm contains a causal Kalman filter to obtain optimal filtered estimates of the state process. Once the filtered estimates are obtained, an anti–causal smoothing recursion is used to compute the smoothed estimates. These causal and anti–causal recursions are similar to the tip–to–base and base–to–tip recursions in the articulated body inertia forward dynamics algorithm.

An alternative formulation of the smoothing algorithm is possible when all the observations over the finite interval are available. It consists of running two independent Kalman filters – one causal and the other anti–causal – to generate two sets of filtered estimates of the system state [14, 15]. While one of the filtered estimates is based upon all the "past" observations, the other one is based upon all the "future" observations. The optimal smoothed estimate has been shown to consist of a simple linear combination of the causal and anti–causal filtered estimates. The structure of this decoupled smoothing algorithm closely resembles the structure of the base-invariant dynamics algorithm described in Algorithms 3.3 and 3.4.

3.5 Extensions to Tree–Topology Manipulators

The extension of the conventional articulated body inertia forward dynamics algorithm in Algorithm 2.3 to tree–topology systems has been described in reference [11]. The recursive computations take on a gather/scatter structure. Recursions proceeding from the tips towards the base gather and sum the outputs from the incoming branches as the recursion progresses. On the other hand, the recursions that start from the base and proceed towards the tips scatter their outputs along each of the outgoing branches. Apart from this difference, the algorithm when extended to tree–topology systems retains the same sequential recursion steps as for serial–chain systems.

As in the case of serial–chain free-flying manipulators, tree–topology free-flying manipulators also possess the base-invariance symmetry arising from the non–unique choice for the base–body. The structure of the decoupled dynamics algorithm for tree–topology free-flying manipulators is illustrated in Figure 5 and takes into account the fact there are more than two extremal bodies. The overall structure of the decoupled dynamics algorithm remains the same as in Algorithm 3.4. Corresponding to each extremal body, there is a fundamental model in which the extremal body is treated as the base-body for the model. Articulated body inertia computations are required for each of these fundamental models. At a link with multiple branches, every branch collects the articulated body inertia outputs from all of the other branches at the link. This data is accumulated by each branch to allow its articulated body inertia procedure to proceed. The overlap in the computations among the articulated body inertia recursions for the fundamental models is such that there are precisely two recursions proceeding in opposite directions across any serial link segment of the manipulator. Lemma 3.2 is still valid and is used to compute the spatial accelerations of each of the links.

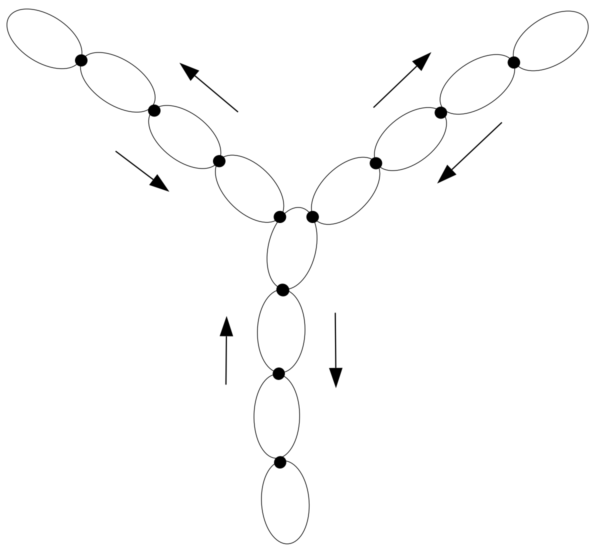

Figure 5: The structure of the decoupled dynamics algorithm for tree topology free-flying manipulators

4 Base–Invariant Operational Space Inertia

We now look at the role of the *operational space inertia* [16, 17] in the dynamics of free-flying manipulators. The operational space inertia inertia $\Lambda(1) \in \mathcal{R}^{6\times 6}$, has traditionally been defined as the effective mass matrix of the whole manipulator as seen at the end–effector. The expression for its inverse is given by

$$\Lambda^{-1}(1) = J_p(1)\boldsymbol{\mathcal{M}}_p^{-1}J_p^*(1) \tag{27}$$

where $J_p(1) \in \mathcal{R}^{6\times\mathcal{N}}$ denotes the Jacobian to the link 1 spatial velocity and is given by

$$J_p(1) = B^*(1)\boldsymbol{\phi}^*\boldsymbol{H}^*, \quad \text{where} \quad B(1) \stackrel{\triangle}{=} \text{col}\left\{\boldsymbol{I}_6\delta(i,1)\right\}_{i=1}^{n} \in \mathcal{R}^{6n\times 6}$$

with $\delta(.,.)$ denoting the Kronecker delta function.

$$\tag{28}$$

We generalize the notion of the operational space inertia so that it can be defined for all links on the manipulator. Thus the operational space inertia for the k^{th} link, $\Lambda(k)$, is the effective mass matrix of the manipulator as seen at frame $\mathcal{O}_k$. Analogous to the earlier definition in (27), the inverse of $\Lambda_p(k)$ is given by the expression

$$\Lambda_p^{-1}(k) = J_p(k)\boldsymbol{\mathcal{M}}_p^{-1}J_p^*(k) \tag{29}$$

where $J_p(k) \in \mathcal{R}^{6\times\mathcal{N}}$ denotes the Jacobian to the k^{th} link spatial velocity and is given by

$$J_p(k) = B^*(k)\boldsymbol{\phi}^*\boldsymbol{H}^*, \quad \text{where} \quad B(k) \stackrel{\triangle}{=} \text{col}\left\{\boldsymbol{I}_6\delta(i,k)\right\}_{i=1}^{n} \in \mathcal{R}^{6n\times 6} \tag{30}$$

We now make use of the following operator identity which has been established in reference [7]:

$$\boldsymbol{\phi}_p^*\boldsymbol{H}_p^*\boldsymbol{\mathcal{M}}_p^{-1}\boldsymbol{H}_p\boldsymbol{\phi}_p = \boldsymbol{\psi}_p^*\boldsymbol{H}_p^*\boldsymbol{D}_p^{-1}\boldsymbol{H}_p\boldsymbol{\psi}_p \stackrel{\triangle}{=} \boldsymbol{\Omega}_p \in \mathcal{R}^{6n\times 6n} \tag{31}$$

It follows from this identity that

$$\Lambda_p^{-1}(k) \stackrel{29}{=} J_p(k)\boldsymbol{\mathcal{M}}_p^{-1}J_p^*(k) \stackrel{30,31}{=} B^*(k)\boldsymbol{\Omega}_p B(k) \tag{32}$$

The subscript p above is a reminder that the above expressions assume that the base–body is link n. However, it is easy to show that even though $J_p(k)$ and $\boldsymbol{\mathcal{M}}_p$ depend on the choice of base–body, $\boldsymbol{\Omega}_p$ is in fact independent of this choice. To see this, let us use the subscript k to denote the use of the k^{th} link as the base–body. It follows from Lemma A.1 in Appendix A that

$$\boldsymbol{\phi}_k^*\boldsymbol{H}_k^* \stackrel{41}{=} \boldsymbol{\phi}_p^*\boldsymbol{H}_p^*\boldsymbol{\mathcal{T}}_{n,k}, \quad \text{and} \quad \boldsymbol{\mathcal{M}}_k \stackrel{44}{=} \boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{\mathcal{M}}_p\boldsymbol{\mathcal{T}}_{n,k}$$

Therefore,

$$\begin{aligned}
\boldsymbol{\Omega}_k &\stackrel{\triangle}{=} \boldsymbol{\phi}_k^*\boldsymbol{H}_k^*\boldsymbol{\mathcal{M}}_k^{-1}\boldsymbol{H}_k\boldsymbol{\phi}_k \\
&= \boldsymbol{\phi}_p^*\boldsymbol{H}_p^*\boldsymbol{\mathcal{T}}_{n,k}[\boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{\mathcal{M}}_p\boldsymbol{\mathcal{T}}_{n,k}]^{-1}\boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{H}_p\boldsymbol{\phi}_p \\
&= \boldsymbol{\phi}_p^*\boldsymbol{H}_p^*\boldsymbol{\mathcal{M}}_p^{-1}\boldsymbol{H}_p\boldsymbol{\phi}_p \\
&\stackrel{31}{=} \boldsymbol{\Omega}_p
\end{aligned}$$

Since $\boldsymbol{\Omega}_p = \boldsymbol{\Omega}_k$ for all k, this quantity is independent of the choice of the base body and we drop the subscript from $\boldsymbol{\Omega}$ altogether. This fact, taken together with (29) also establishes the invariance of the operational space inertia $\Lambda(k)$ with respect to the choice of the base–body.

It has been shown in references [7, 17] that a block diagonal operator $\boldsymbol{\Upsilon} \in \mathcal{R}^{6n\times 6n}$, can be used to decompose $\boldsymbol{\Omega}$ as follows:

$$\boldsymbol{\Omega} = \boldsymbol{\Upsilon} + \boldsymbol{\psi}_p^*\boldsymbol{\Upsilon} + \boldsymbol{\Upsilon}\boldsymbol{\psi}_p \tag{33}$$

The block diagonal components of $\boldsymbol{\Upsilon}$ are denoted $\boldsymbol{\Upsilon}(k) \in \mathcal{R}^{6\times 6}$, and are defined by the following link n to link 1 recursion:

$$\left\{\begin{array}{rcl}
 & \boldsymbol{\Upsilon}^+(n) = \mathbf{0} & \\
\textbf{for } \boldsymbol{k} & \boldsymbol{=} & \boldsymbol{n \cdots 1} \\
\boldsymbol{\Upsilon}(k) & = & \overline{\boldsymbol{\tau}}_p^*(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k) + \boldsymbol{H}_p^*(k)\boldsymbol{D}_p^{-1}(k)\boldsymbol{H}_p(k) \\
\boldsymbol{\Upsilon}^+(k-1) & = & \boldsymbol{\phi}^*(k,k-1)\boldsymbol{\Upsilon}(k)\boldsymbol{\phi}(k,k-1) \\
\textbf{end loop} & &
\end{array}\right. \tag{34}$$

On the face of it, it appears from (34) that we should be using the subscript p on $\boldsymbol{\Upsilon}$ and its components to indicate their dependence on the choice of link n as the base–body. However, this is unnecessary because $\boldsymbol{\Upsilon}$ is in fact independent of the choice of the base–body. This fact is obvious once we realize that the three terms on the right hand side of (33) are block diagonal, block strictly upper–triangular and block strictly lower-triangular respectively. Since $\boldsymbol{\Omega}$ is independent of the choice of base–body, therefore so also is $\boldsymbol{\Upsilon}$ and its components. From the definition in (32) and the decomposition in (33), it follows that

$$\Lambda^{-1}(k) \overset{32,33}{=} B^*(k)[\boldsymbol{\Upsilon} + \boldsymbol{\psi}_p^*\boldsymbol{\Upsilon} + \boldsymbol{\Upsilon}\boldsymbol{\psi}_p]B(k) = B^*(k)\boldsymbol{\Upsilon}B(k) = \boldsymbol{\Upsilon}(k) \tag{35}$$

That is, $\boldsymbol{\Upsilon}(k)$ is the inverse of the operational space inertia $\Lambda(k)$. For ground–based manipulators, $\boldsymbol{\Upsilon}(k)$ is singular for the first 5 links connected to the base. The singularity reflects the fact that there are directions along which spatial forces induce no motion in the manipulator. In contrast, at the base–body (link n) of a free-flying manipulator

$$\boldsymbol{\Upsilon}(n) \overset{34}{=} \boldsymbol{P}^{-1}(n)$$

Thus, for free-flying manipulators, $\boldsymbol{\Upsilon}(n)$ is always invertible. Indeed, it can be shown that $\boldsymbol{\Upsilon}(k)$ is invertible for all k. The invertibility property reflects the fact that any spatial force at any point on the free-flying manipulator will cause a non–zero acceleration of the free–flying manipulator.

Even though we have seen that the values of the $\boldsymbol{\Upsilon}(.)$'s do not depend of the choice of a base–body, the computational scheme in (34) certainly does, since it makes use of the articulated body inertia quantities computed with link n as the base. Using link 1 as the base body, we obtain the following dual algorithm for computing $\boldsymbol{\Upsilon}(k)$ which makes use of the dual articulated body inertia quantities computed in Algorithm 3.1:

$$\begin{cases} \quad\quad \boldsymbol{\Upsilon}(0) = \mathbf{0} \\ \textbf{for } \boldsymbol{k} \;=\; \mathbf{0}\cdots\boldsymbol{n}-\mathbf{1} \\ \quad \boldsymbol{\Upsilon}^+(k) \;=\; \overline{\boldsymbol{\tau}}_s^*(k)\boldsymbol{\Upsilon}(k)\overline{\boldsymbol{\tau}}_s(k) + \boldsymbol{H}^*(k)\boldsymbol{D}_s^{-1}(k)\boldsymbol{H}(k) \\ \quad \boldsymbol{\Upsilon}(k+1) \;=\; \boldsymbol{\phi}^*(k,k+1)\boldsymbol{\Upsilon}^+(k)\boldsymbol{\phi}(k,k+1) \\ \textbf{end loop} \end{cases} \tag{36}$$

Both (34) and (36) describe computational schemes consisting of one recursion to compute the articulated body inertia quantities, followed by a recursion in the opposite direction to compute the $\boldsymbol{\Upsilon}(k)$'s. The lemma below shows that the symmetry of free-flying manipulators in fact allows us to dispense with these algorithms and express the $\boldsymbol{\Upsilon}(k)$'s directly using the articulated body inertia $\boldsymbol{P}(k)$ and its dual $\boldsymbol{S}^+(k)$.

Lemma 4.1

$$[\boldsymbol{\Upsilon}(k)]^{-1} = \boldsymbol{P}(k) + \boldsymbol{S}(k) \tag{37a}$$

$$[\boldsymbol{\Upsilon}^+(k)]^{-1} = \boldsymbol{P}^+(k) + \boldsymbol{S}^+(k) \tag{37b}$$

Proof: *It follows from* (6), (16) *and* (34) *that* (37a) *is true for* $k = i$ *if and only if* (37b) *is true for* $k = i - 1$*. We have that* $\boldsymbol{\Upsilon}^{-1}(n) = \boldsymbol{P}(n)$*. Since by definition* $\boldsymbol{S}(n) = \boldsymbol{0}$*, this implies that* (37a) *holds for* $k = n$*. Thus* (37b) *holds for* $k = n - 1$*. We use proof by induction to establish the general result.*

Assume that (37b) *holds for a certain* k*. Then from* (6), (34) *and that* $\boldsymbol{S}(k)\boldsymbol{H}^*(k) = 0$*, it follows that*

$$\begin{aligned}
[\boldsymbol{P}(k) + \boldsymbol{S}(k)]\boldsymbol{\Upsilon}(k) &\overset{34}{=} [\boldsymbol{P}(k) + \boldsymbol{S}(k)][\overline{\boldsymbol{\tau}}_p^*(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k) + \boldsymbol{H}_p^*(k)\boldsymbol{D}_p^{-1}(k)\boldsymbol{H}_p(k)] \\
&\overset{6}{=} \boldsymbol{P}^+(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p^*(k) + \boldsymbol{\tau}_p(k) + \boldsymbol{S}(k)\overline{\boldsymbol{\tau}}_p^*(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k) \\
&= \boldsymbol{P}^+(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k) + \boldsymbol{\tau}_p(k) + \boldsymbol{S}(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k) \\
&= \boldsymbol{I}_6 - \boldsymbol{\tau}_s(k)\boldsymbol{S}^+(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k) \\
&= \boldsymbol{I}_6 - \boldsymbol{\tau}_s(k)[\boldsymbol{I}_6 - \boldsymbol{P}^+(k)\boldsymbol{\Upsilon}^+(k)]\overline{\boldsymbol{\tau}}_p(k) \\
&= \boldsymbol{I}_6 - \boldsymbol{\tau}_s(k)\overline{\boldsymbol{\tau}}_p(k)[\boldsymbol{I}_6 - \boldsymbol{P}^+(k)\boldsymbol{\Upsilon}^+(k)\overline{\boldsymbol{\tau}}_p(k)] \\
&= \boldsymbol{I}_6
\end{aligned}$$

The last step follows from the fact that

$$\boldsymbol{\tau}_s(k)\overline{\boldsymbol{\tau}}(k) \;=\; \boldsymbol{\tau}_s(k) - \boldsymbol{G}_s(k)\boldsymbol{H}(k)\boldsymbol{G}_p(k)\boldsymbol{H}(k) = \boldsymbol{\tau}_s(k) - \boldsymbol{G}_s(k)\boldsymbol{H}(k) = \boldsymbol{0}$$

Thus if (37b) *is true for a certain* k*,* (37a) *is also true for the same* k*. When combined with the earlier result, it implies that* (37a) *is also true for* $k - 1$*. This establishes the induction process since we have seen that* (37a) *is in fact true for* $k = n$*.*

∎

This result once again highlights the natural base-invariance symmetry of free-flying manipulators. The positive definiteness of $\boldsymbol{P}(.)$ and $\boldsymbol{S}^+(.)$ taken together with the above result clearly implies that $\boldsymbol{\Upsilon}(.)$ and $\boldsymbol{\Upsilon}^+(.)$ are also positive definite (and hence invertible). Also, the operational space inertia $\Lambda(k)$ is given by

$$\Lambda(k) \overset{35,37a}{=} \boldsymbol{P}(k) + \boldsymbol{S}(k) \tag{38}$$

Lemma 4.1 provides us with a new method to compute the operational space inertias for the links on the free-flying manipulator. The algorithm is as follows:

Algorithm 4.1 *Decoupled Computation of Operational Space Inertias*

1. *(a) Compute the articulated body quantities $\boldsymbol{P}(.)$ recursively from link 1 to link n using Algorithm 2.2. (b) Simultaneously compute the dual articulated body quantities $\boldsymbol{S}^{+}(.)$ recursively from link n to link 1 using Algorithm 3.1.*

2. *Compute $\Lambda(k) = [\boldsymbol{P}(k) + \boldsymbol{S}(k)]$ for the k^{th} link. These computations can be carried out independently for each link.*

Unlike the algorithms in (34) and (36), Algorithm 4.1 has a decoupled structure arising from the symmetry of the free-flying manipulators. The two sequential recursions in the earlier algorithms are now replaced by a pair of parallel recursions. This can be used to advantage in a parallel computing environment.

As is the case for serial chain manipulators, the operational space inertia at any link of a tree–topology free-flying manipulator is simply obtained by summing up the $\boldsymbol{P}$ and $\boldsymbol{S}$ articulated body inertias at the link.

5 Conclusions

Space manipulators possess a base-invariance symmetry not encountered in terrestrial manipulators. The symmetry arises from the freedom available in the choice of a base–body for the manipulator. We use this symmetry to develop a new $O(\mathcal{N})$ forward dynamics algorithm with a highly decoupled structure. A key idea is to treat "every link" as a base body. It is shown that key dynamical quantities can be obtained by combining results from independent articulated body inertia computations. Also discussed are the use of non–minimal coordinates to further decouple the forward dynamics algorithm and the extension of the base-invariant algorithm to tree–topology free-flying manipulators. In addition, it has also shown that the algorithm for computing the operational space inertia inertia for the manipulator can be simplified using the decoupled articulated body inertia recursions.

Acknowledgments

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

References

[1] H. Alexander and R. Cannon, "Experiments on the Control of a Satellite Manipulator," in *Proc. American Control Conference*, 1987.

[2] Z. Vafa, "Space Manipulator Motions with No Satellite Attitude Disturbances," in *IEEE International Conference on Robotics and Automation*, (Cincinnati, OH), May 1990.

[3] Y. Nakamura and R. Mukherjee, "Nonholonomic Path Planning for Space Robot via Bi–Directional Approach," in *IEEE International Conference on Robotics and Automation*, (Cincinnati, OH), Apr. 1990.

[4] E. Papadopoulos and S. Dubowsky, "On the Nature of Control Algorthms for Space Manipulators," in *IEEE International Conference on Robotics and Automation*, (Cincinnati, OH), May 1990.

[5] R. Mukherjee and Y. Nakamura, "Formulation and Efficient Computation of Inverse Dynamics of Space Robots," in *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 400–406, Mar. 1992.

[6] R. Featherstone, "The Calculation of Robot Dynamics using Articulated-Body Inertias," *The International Journal of Robotics Research*, vol. 2, pp. 13–30, Spring 1983.

[7] G. Rodriguez, K. Kreutz-Delgado, and A. Jain, "A Spatial Operator Algebra for Manipulator Modeling and Control," *The International Journal of Robotics Research*, vol. 10, pp. 371–381, Aug. 1991.

[8] A. Jain, "Unified Formulation of Dynamics for Serial Rigid Multibody Systems," *Journal of Guidance, Control and Dynamics*, vol. 14, pp. 531–542, May–June 1991.

[9] A. Jain and G. Rodriguez, "An Analysis of the Kinematics and Dynamics of Underactuated Manipulators," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 411–422, Aug. 1993.

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

[11] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "Spatial Operator Algebra for Multibody System Dynamics," *Journal of the Astronautical Sciences*, vol. 40, pp. 27–50, Jan.–March 1992.

[12] 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.

[13] 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.

[14] D. Fraser and J. Potter, "The Optimum Linear Smoother as a Combination of Two Optimum Linear Filters," *IEEE Transactions on Automatic Control*, pp. 387–390, Aug. 1969.

[15] A. Gelb, *Applied Optimal Estimation.* M.I.T. Press, Cambridge, Mass., 1974.

[16] O. Khatib, "Object Manipulation in a Multi-Effector System," in *4th International Symposium Robotics Research*, pp. 137–144, 1988.

[17] K. Kreutz-Delgado, A. Jain, and G. Rodriguez, "Recursive Formulation of Operational Space Control," *The International Journal of Robotics Research*, vol. 11, pp. 320–328, Aug. 1992.

A Mass Matrix Factorization with Link k as the Base-Body

Section 2 defined expressions for the spatial operators such as $\boldsymbol{H}$, $\boldsymbol{\phi}$ etc. and the mass matrix with link n as the manipulator base-body. The Newton-Euler Factorization of the mass matrix $\boldsymbol{\mathcal{M}}$ in (4a) formed the basis for the derivation of the expression for the mass matrix inverse in (10c). In this appendix we show that these expressions and results can be developed equally well when a link other than link n serves as the base-body. As the first step, Lemma A.1 defines an invertible transformation $\boldsymbol{\mathcal{T}}_{k,n}(.,.)$ which maps generalized coordinates for the link n base-body model into the generalized velocity coordinates for the link k base-body model. Subsequently, Lemma A.2 gives new expressions for the $\boldsymbol{H}$, $\boldsymbol{\phi}$ etc. spatial operators as well as the Newton-Euler Factorization of the new mass matrix for the link k base-body model. These latter expressions are sufficient to derive the new expressions for the mass matrix inverse in a manner identical to that for the case with link n as the base-body. These conclusions are not surprising, and can be obtained from physical arguments. However, for the sake of completeness, we include the mathematical details and proofs here.

The spatial velocity of the base–body contributes six of the generalized velocity coordinates for the manipulator. The generalized velocities vector $\boldsymbol{\beta}$ with link n as the base body consists of $\{\boldsymbol{\beta}(1), \cdots, \boldsymbol{\beta}(n-1), V(n)\}$ where we have used the fact that

$$\boldsymbol{\beta}(n) = V(n)$$

In this section we will use the base–body index as a subscript to denote the choice of the base–body. Thus $\boldsymbol{\beta}$ above will now be denoted $\boldsymbol{\beta}_n$.

When we switch the base–body from link n to another link, say link k, the six velocity coordinates given by $V(n)$ are replaced by the six coordinates $V(k)$ consisting of the the spatial velocity of link k so that the new coordinates $\boldsymbol{\beta}_k \in \mathcal{R}^{\mathcal{N}}$ are given by

$$\boldsymbol{\beta}_k \triangleq \begin{pmatrix} \boldsymbol{\beta}(1) \\ \vdots \\ \boldsymbol{\beta}(n-1) \\ V(k) \end{pmatrix} \tag{39}$$

Lemma A.1 below defines the nonlinear transformation $\boldsymbol{\mathcal{T}}_{k,n}(.,.)$ which provides the mapping between the $\boldsymbol{\beta}_k$ and $\boldsymbol{\beta}_n$ coordinates. First we rewrite $\boldsymbol{H}_n$ in the following partitioned form

$$\boldsymbol{H}_n = \begin{pmatrix} \boldsymbol{\mathcal{H}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_6 \end{pmatrix}, \quad \text{where} \quad \boldsymbol{\mathcal{H}} \triangleq \text{diag}\left\{\boldsymbol{H}(i)\right\}_{i=1}^{n-1} \in \mathcal{R}^{(\mathcal{N}-6)\times 6(n-1)} \tag{40}$$

Lemma A.1 *The transformation map $\boldsymbol{\mathcal{T}}_{k,n}$ is such that*

$$\boldsymbol{\beta}_k = \boldsymbol{\mathcal{T}}_{k,n}\boldsymbol{\beta}_n, \quad \textit{where} \quad \boldsymbol{\mathcal{T}}_{k,n} = \begin{pmatrix} \boldsymbol{I}_{n-1} & \mathbf{0} \\ X_k\boldsymbol{\mathcal{H}}^* & \boldsymbol{\phi}^*(n,k) \end{pmatrix} \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}} \tag{41}$$

with

$$X_k \triangleq [\mathbf{0}, \cdots, \boldsymbol{I}_6, \boldsymbol{\phi}^*(k+1,k), \cdots, \boldsymbol{\phi}^*(n-1,k)] \in \mathcal{R}^{6\times n-1} \tag{42}$$

The inverse transformation $\boldsymbol{\mathcal{T}}_{n,k}$ such that $\boldsymbol{\beta}_n = \boldsymbol{\mathcal{T}}_{n,k}\boldsymbol{\beta}_k$ is given by

$$\boldsymbol{\mathcal{T}}_{n,k} \triangleq \boldsymbol{\mathcal{T}}_{k,n}^{-1} = \begin{pmatrix} \boldsymbol{I}_{n-1} & \mathbf{0} \\ -\boldsymbol{\phi}^*(k,n)X_k\boldsymbol{\mathcal{H}}^* & \boldsymbol{\phi}^*(k,n) \end{pmatrix} \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}} \tag{43}$$

Proof: *From* (3a) *it follows that*

$$V(k) = \sum_{i=k}^{n} \boldsymbol{\phi}^*(i,k)\boldsymbol{H}^*(i)\boldsymbol{\beta}(i) = [X_k\boldsymbol{\mathcal{H}}^*,\ \boldsymbol{\phi}^*(n,k)]\boldsymbol{\beta}_n$$

From this follows the expression for $\boldsymbol{\mathcal{T}}_{k,n}$ in (41)*. The expression for its inverse, $\boldsymbol{\mathcal{T}}_{n,k}$, follows quite simply from matrix manipulation.* ∎

The kinetic energy of the manipulator is given by

$$\frac{1}{2}\boldsymbol{\beta}_n^*\boldsymbol{\mathcal{M}}_n\boldsymbol{\beta}_n = \frac{1}{2}\boldsymbol{\beta}_k^*\boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{\mathcal{M}}_n\boldsymbol{\mathcal{T}}_{n,k}\boldsymbol{\beta}_k$$

therefore, the mass matrix $\boldsymbol{\mathcal{M}}_k$ for the $\boldsymbol{\beta}_k$ set of velocity coordinates is given by

$$\boldsymbol{\mathcal{M}}_k = \boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{\mathcal{M}}_n\boldsymbol{\mathcal{T}}_{n,k} = \boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{H}_n\boldsymbol{\phi}_n\boldsymbol{M}_n\boldsymbol{\phi}_n^*\boldsymbol{H}_n^*\boldsymbol{\mathcal{T}}_{n,k} \tag{44}$$

We now show that the operator formalism developed with link n as the base–body – including the results related to the operator factorization and inversion of the mass matrix in Lemma 2.1 – also hold when link k is chosen as the base body. If we look closely at the derivation of the factorization and inversion results for the mass matrix and the articulated body inertia forward dynamics algorithm, we see that the key properties in the derivation were that the mass matrix has a Newton–Euler operator factorization as in (4a), that $\boldsymbol{H}_n^*$ and $\boldsymbol{M}_n$ are diagonal, and $\boldsymbol{\phi}_n$ has the form

$$\boldsymbol{\phi}_n = [\boldsymbol{I} - \boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_n}]^{-1}$$

where $\boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_n}$ is a nilpotent matrix. The factorization of the new mass matrix in (44) does not quite have the form of the desired Newton-Euler operator factorization due to the presence of the $\boldsymbol{\mathcal{T}}_{n,k}$ terms. We show in Lemma A.2 below that a similar Newton–Euler operator factorization of the new mass matrix is also possible, and one from which the remaining operator results follow. However, to do this we need to define a new velocity coordinates vector $\boldsymbol{\beta}_k^o$ obtained by reordering the components of $\boldsymbol{\beta}_k$ as follows:

$$\boldsymbol{\beta}_k^o = \boldsymbol{\mathcal{P}}\boldsymbol{\beta}_k = \begin{pmatrix} \boldsymbol{\beta}(1) \\ \vdots \\ \boldsymbol{\beta}(k-1) \\ V(k) \\ \boldsymbol{\beta}(k) \\ \vdots \\ \boldsymbol{\beta}(n-1) \end{pmatrix}, \quad \text{where} \quad \boldsymbol{\mathcal{P}} \stackrel{\triangle}{=} \begin{pmatrix} \boldsymbol{I}_{(k-1)} & \mathbf{0} \\ \mathbf{0} & \begin{pmatrix} \mathbf{0} & \boldsymbol{I}_6 \\ \boldsymbol{I}_{6(n-k-1)} & \mathbf{0} \end{pmatrix} \end{pmatrix} \tag{45}$$

Note that $\boldsymbol{\mathcal{P}}$ is simply a permutation matrix which reorders the coordinate elements within $\boldsymbol{\beta}_k$. Moreover, $\boldsymbol{\mathcal{P}}^{-1} = \boldsymbol{\mathcal{P}}^*$. In the β_k^o coordinates, the mass matrix $\boldsymbol{\mathcal{M}}_k^o$ is given by

$$\boldsymbol{\mathcal{M}}_k^o = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{M}}_k\boldsymbol{\mathcal{P}}^* \stackrel{44}{=} \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{\mathcal{M}}_n\boldsymbol{\mathcal{T}}_{n,k}\boldsymbol{\mathcal{P}}^* = \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{T}}_{n,k}^*\boldsymbol{H}_n\boldsymbol{\phi}_n\boldsymbol{M}_n\boldsymbol{\phi}_n^*\boldsymbol{H}_n^*\boldsymbol{\mathcal{T}}_{n,k}\boldsymbol{\mathcal{P}}^* \tag{46}$$

Lemma A.2 *The mass matrix $\boldsymbol{\mathcal{M}}_k^o$ has the following Newton-Euler operator factorization:*

$$\boldsymbol{\mathcal{M}}_k^o = \boldsymbol{H}_k\boldsymbol{\phi}_k\boldsymbol{M}_k\boldsymbol{\phi}_k^*\boldsymbol{H}_k^* \tag{47}$$

where

$$\begin{aligned} \boldsymbol{H}_k &\stackrel{\triangle}{=} \boldsymbol{\mathcal{P}}\boldsymbol{H}\boldsymbol{\mathcal{P}}^*, \quad \boldsymbol{\phi}_k \stackrel{\triangle}{=} [\boldsymbol{I} - \boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_k}]^{-1}, \quad \boldsymbol{M}_k \stackrel{\triangle}{=} \Delta_{\boldsymbol{\phi}}\boldsymbol{M}_n\Delta_{\boldsymbol{\phi}}^* \\ \Delta_{\boldsymbol{\phi}} &\stackrel{\triangle}{=} \begin{pmatrix} \boldsymbol{I}_{6k} & \mathbf{0} \\ \mathbf{0} & \operatorname{diag}\Big\{\boldsymbol{\phi}(i-1,i)\Big\}_{i=k+1}^{n} \end{pmatrix} \in \mathcal{R}^{6n\times 6n}, \quad \boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_k} \stackrel{\triangle}{=} \begin{pmatrix} Y_1 & Y_3 \\ \mathbf{0} & Y_2 \end{pmatrix} \end{aligned} \tag{48}$$

$\boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_k}$ *is nilpotent with* $Y_1 \in \mathcal{R}^{6k\times 6k}$, $Y_2 \in \mathcal{R}^{6(n-k)\times 6(n-k)}$ *and* $Y_3 \in \mathcal{R}^{6k\times 6(n-k)}$ *defined as*

$$Y_1 \triangleq \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\phi}(2,1) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\phi}(3,2) & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{\phi}(k,k-1) & \mathbf{0} \end{pmatrix}$$

$$Y_2 \triangleq \begin{pmatrix} \mathbf{0} & \boldsymbol{\phi}(k,k+1) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\phi}(k+1,k+2) & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \\ \vdots & \vdots & \ddots & \vdots & \boldsymbol{\phi}(n-2,n-1) & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

and

$$Y_3 \triangleq \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{I}_6 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

Proof: *We have*

$$\begin{aligned} \boldsymbol{H}^*\boldsymbol{\mathcal{T}}_{n,k} &\overset{43}{=} \begin{pmatrix} \boldsymbol{\mathcal{H}}^* & \mathbf{0} \\ -\boldsymbol{\phi}^*(k,n)X_k\boldsymbol{\mathcal{H}}^* & \boldsymbol{\phi}^*(k,n) \end{pmatrix} \\ &= \boldsymbol{\mathcal{Q}}\boldsymbol{H}^*, \quad \text{where} \quad \boldsymbol{\mathcal{Q}} \triangleq \begin{pmatrix} \boldsymbol{I}_{6(n-1)} & \mathbf{0} \\ -\boldsymbol{\phi}^*(k,n)X_k & \boldsymbol{\phi}^*(k,n) \end{pmatrix} \in \mathcal{R}^{6n\times 6n} \end{aligned} \tag{49}$$

Let

$$e_k \triangleq \left[\mathrm{col}\left\{\boldsymbol{I}_6\delta(i,k)\right\}_{i=1}^{n-1}\right]^* = [\mathbf{0},\cdots,\boldsymbol{I}_6,\cdots,\mathbf{0}], \quad \text{and} \quad \hat{\boldsymbol{I}} \triangleq [\boldsymbol{I}_{6(n-1)},\mathbf{0}]$$

where $\delta(.,.)$ *denotes the Kronecker delta function. We have that*

$$[X_k,\ \boldsymbol{\phi}^*(n,k)] = e_k\boldsymbol{\phi}_n^* \tag{50}$$

Therefore,

$$\boldsymbol{\mathcal{Q}}^{-1} = \begin{pmatrix} \boldsymbol{I}_{6(n-1)} & \mathbf{0} \\ X_k & \boldsymbol{\phi}^*(n,k) \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{I}} \\ e_k\boldsymbol{\phi}_n^* \end{pmatrix} \tag{51}$$

$$\boldsymbol{\phi}_n^*\boldsymbol{\mathcal{Q}} = \left[\boldsymbol{\mathcal{Q}}^{-1}\boldsymbol{\phi}_n^{-*}\right]^{-1} \overset{51}{=} \begin{pmatrix} \hat{\boldsymbol{I}}\boldsymbol{\phi}_n^{-*} \\ e_k \end{pmatrix}^{-1} = \begin{pmatrix} \hat{\boldsymbol{I}} - \hat{\boldsymbol{I}}\boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_n^*} \\ e_k \end{pmatrix}^{-1} \tag{52}$$

At the component level,

$$\begin{pmatrix} \hat{\boldsymbol{I}} - \hat{\boldsymbol{I}}\boldsymbol{\mathcal{E}}_{\phi_n^*} \\ e_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{I}_6 & -\boldsymbol{\phi}^*(2,1) & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_6 & -\boldsymbol{\phi}^*(3,2) & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \boldsymbol{I} & -\boldsymbol{\phi}^*(n,n-1) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{I}_6 & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

The above matrix is identical in form to $(\boldsymbol{I} - \boldsymbol{\mathcal{E}}^*_{\phi_n})$ *except for the last row. Straightforward matrix manipulation shows that*

$$\boldsymbol{\mathcal{P}} \begin{pmatrix} \hat{\boldsymbol{I}} - \hat{\boldsymbol{I}}\boldsymbol{\mathcal{E}}_{\phi_n^*} \\ e_k \end{pmatrix} \Delta_{\boldsymbol{\phi}}{}^* = [\boldsymbol{I} - \boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_k^*}] \tag{54}$$

In the above, the permutation matrix $\boldsymbol{\mathcal{P}}$ *transforms* $\begin{pmatrix} \hat{\boldsymbol{I}} - \hat{\boldsymbol{I}}\boldsymbol{\mathcal{E}}_{\phi_n^*} \\ e_k \end{pmatrix}$ *into a tri–diagonal matrix form, while* $\Delta_{\boldsymbol{\phi}}{}^*$ *normalizes the terms along the diagonal to* $\boldsymbol{I}_6$. *It is easy to verify that* $\boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_k}$ *is nilpotent, and hence* $(\boldsymbol{I} - \boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}_k})$ *is invertible. We denote this inverse by* $\boldsymbol{\phi}_k$. *Thus*

$$\begin{pmatrix} \hat{\boldsymbol{I}} - \hat{\boldsymbol{I}}\boldsymbol{\mathcal{E}}_{\boldsymbol{\phi}^*} \\ e_k \end{pmatrix}^{-1} \overset{54}{=} \Delta_{\boldsymbol{\phi}}{}^* \boldsymbol{\phi}_k \boldsymbol{\mathcal{P}} \tag{55}$$

Therefore we have that

$$\boldsymbol{\phi}^* \boldsymbol{H}^* \boldsymbol{\mathcal{T}}_{n,k} \boldsymbol{\mathcal{P}}^* \overset{49}{=} \boldsymbol{\phi}^* \boldsymbol{\mathcal{Q}} \boldsymbol{H}^* \boldsymbol{\mathcal{P}}^* \overset{52,55}{=} \Delta_{\boldsymbol{\phi}}{}^* \boldsymbol{\phi}_k^* \boldsymbol{\mathcal{P}} \boldsymbol{H}^* \boldsymbol{\mathcal{P}}^* = \Delta_{\boldsymbol{\phi}}{}^* \boldsymbol{\phi}_k^* \boldsymbol{H}_k^* \tag{56}$$

Thus

$$\boldsymbol{\mathcal{M}}_k^o = \boldsymbol{\mathcal{P}} \boldsymbol{\mathcal{T}}^*_{n,k} \boldsymbol{H} \boldsymbol{\phi} \boldsymbol{M} \boldsymbol{\phi}^* \boldsymbol{H}^* \boldsymbol{\mathcal{T}}_{n,k} \boldsymbol{\mathcal{P}}^* = \boldsymbol{H}_k \boldsymbol{\phi}_k \boldsymbol{M}_k \boldsymbol{\phi}_k^* \boldsymbol{H}_k^*$$

This establishes the result. ∎

Note that while $\boldsymbol{\phi}_k$ is no longer fully lower triangular, it is nevertheless block–wise triangular. The new indexing scheme is more natural in that the sequence of coordinates now follows the natural ordering of the hinges along the manipulator. Since we now see that $\boldsymbol{\mathcal{M}}_k^o$ has the necessary Newton–Euler operator factorization, the operator inversion results corresponding to Lemma 2.1 can be obtained here as well. Indeed, it can be shown that

$$\{\boldsymbol{\mathcal{M}}_k^o\}^{-1} = [\boldsymbol{I} - \boldsymbol{H}_k \boldsymbol{\psi}_k \boldsymbol{K}_k]^* \boldsymbol{D}_k^{-1} [\boldsymbol{I} - \boldsymbol{H}_k \boldsymbol{\psi}_k \boldsymbol{K}_k] \tag{57}$$