Base-Invariant Symmetric Dynamics of Free–Flying Space Manipulators

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

Abstract

Free–flying space manipulators have a symmetry not normally encountered in terrestrial manipulators, arising from the freedom of choice for the manipulator base-body. This paper analyzes the relationship between this natural symmetry and the dynamical equations of space manipulators. The 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. Each of the recursions is based upon sequential $O(\mathcal{N})$ forward dynamics algorithms whose amputations are decoupled and independent of each other. The new algorithms offer advantages for parallel computation. Computation of the operational space inertia inertia for the links in the manipulator is also discussed.

1 Introduction

This paper studies the dynamical properties of free-flying space manipulators. Space manipulators have several features not encountered in terrestrial manipulators [1-5]. One difference is that the base-body for free-flying space manipulators has full 6 degrees of freedom. Typically, from practical considerations a link is preferably chosen as the base-body, eg. the spacecraft bus. However, analytically any of the links in the manipulator can be chosen with equal validity as the base-body. The symmetry of space manipulators arises simply from this non-uniqueness in the choice of the base-body for the manipulator.

The generalization of the inverse dynamics algorithm to space manipulators has been dealt with in reference [6] as a special case of the class of under–actuated manipulators. In this paper the focus is developing a decoupled and symmetric forward dynamics algorithm for free--flying space manipulators., The point of departure is the well known and highly efficient $O(\mathcal{N})$ articulated body inertia forward dynamics algorithms for general manipulators [7-9]. This algorithm is highly sequential in nature and consists of a series of recursive computations to compute the generalized accelerations for the manipulator. 'The symmetry of the space manipulators is used to transform this forward dynamics algorithm into a new 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. Also, it is shown that the operational space inertia, inertia for space manipulators can be obtained as a simple combination of the pair of articulated body inertia recursions. The analysis is general and applicable even when the base body forces arc non--zero, that is, even when the linear and angular momenta arc not conserved.

2 Modeling and Dynamics of Manipulators

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

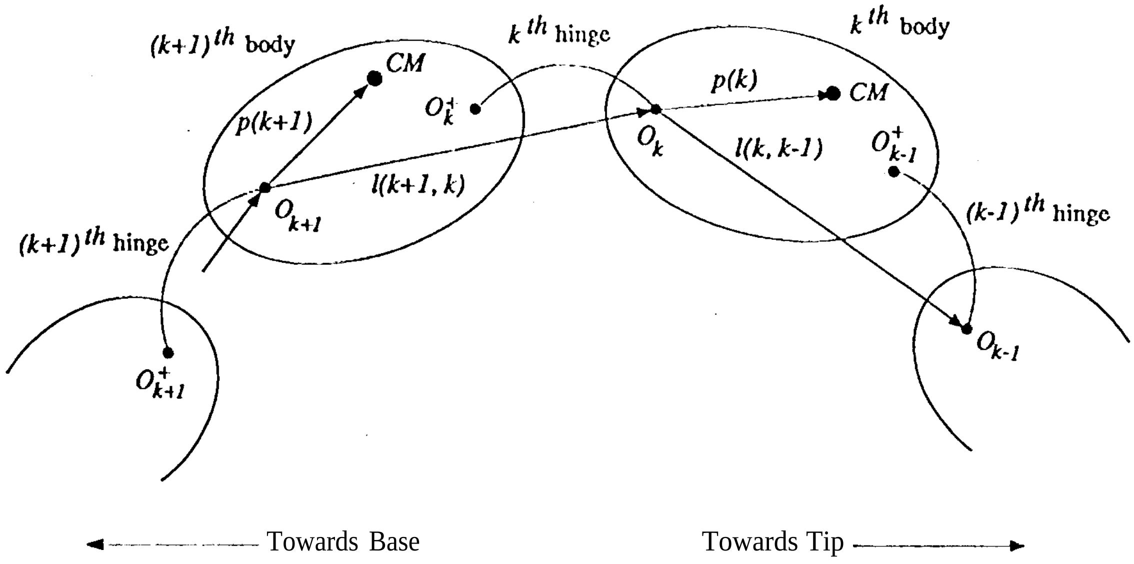

Figure 1: Illustration of' the links and hinges in a manipulator

denoted link n. The inertial frame is denoted "link $(n+1)$". The k^{th} link has two frames denoted $\mathcal{O}_k$ and $\mathcal{O}_{k-1}^+$ attached to it. Frame $\mathcal{O}_k$ is on the inboard side and is the body frame for the k^{th} link. The k^{th} hinge connects the $(k+1)^{th}$ and k^{th} links and its motion is defined as the motion of frame $\mathcal{O}_k$ with respect to frame $\mathcal{O}_k^+$. Free spat.c motion of the manipulator is handled by attaching a 6 degree of freedom hinge between the base link and the inertial frame. The k^{th} hinge is assumed to have $r(k)$ degrees of freedom where $1 \leq r(k) \leq 6$, and its vector of generalized coordinates is denoted $\theta(k)$. 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 arc no nonholonomic constraints on the hinge. The vector of generalized velocities for the k^{th} hinge is denoted $\beta(k) \in$ R'. The choice of the hinge angle rates /3(k) for the generalized velocities $\beta(k)$ is often an obvious and convenient choice. However, when the number of hinge degrees of freedom is larger than 1, alternative choices arc often preferred since they simplify and decouple the kinematic and dynamic parts of the equations of motion, An example is the usc of the relative angular velocity (rather than Euler angle rates) for the generalized velocities vector in the case of a free-flying rigid body. The overall number of degrees of freedom for the manipulator is given by $N = \sum_{k=1}^{n} r(k)$.

The *spatial velocity* $V(k)$ of the k^{th} body frame $\mathcal{O}_k$ is defined as $V(k) = \begin{pmatrix} w(k) \\ v(k) \end{pmatrix} \in \mathbb{R}^6$, with w(k) and v(k) denoting the angular and linear velocities of $\mathcal{O}_k$. The relative spatial "velocity across the k^{th} hinge is given by $H^*(k)\beta(k)$ where $H^*(k) \in \mathbb{R}^{6\times r(k)}$ is the *hinge map matrix* for the hinge. The *spatial force* of interaction $f(k)$ across the k^{th} hinge is denoted $f(k) = \begin{pmatrix} N(k) \\ F(k) \end{pmatrix} \in \mathbb{R}^6$, with $N(k)$ and $F(k)$ denoting the moment and force components respectively. The *spatial inertia* $M(k)$ of the k^{th} link about frame $\mathcal{O}_k$ is defined as

$$M(k) = \begin{pmatrix} \mathcal{J}(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)I_3 \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

where $m(k)$ is the mass, $p(k) \in \mathbb{R}^3$ is the vector from $\mathcal{O}_k$ to the center of 1113SS, and $\mathcal{J}(k) \in \mathbb{R}^{3\times 3}$ is the inertia of the k^{th} link about $\mathcal{O}_k$.

With $V(k)$ denoting the spatial velocity, $\alpha(k)$ the *spatial acceleration*, $f(k)$ the spatial force and $T(k)$ the hinge generalized force at $\mathcal{O}_k$ for the k^{th} link, the following Newton- Euler recursive equations [8, 10] describe the equations of motion as well as an $O(\mathcal{N})$ inverse dynamics algorithm for the serial manipulator:

Algorithm 1

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

$$\left\{\begin{array}{l} \mathbf{f(0)} = \mathbf{0} \\ for\ k = 1 \cdots n \\ \quad f(k) = \phi(k,k-1)f(k-1) + M(k)\alpha(k) + b(k) \\ \quad T(k) = H(k)f(k) \\ end\ loop \end{array}\right. \tag{2.1}$$

where $a(k)$ and $b(k)$ denote the velocity dependent *Coriolis acceleration and gyroscopic force* terms respectively and

$$a(k) \triangleq \begin{pmatrix} \tilde{\omega}(k+1)\omega(k) \\ \tilde{\omega}(k+1)[v(k) - v(k+1)] \end{pmatrix}, \quad \text{and } b(k) \triangleq \begin{pmatrix} \tilde{\omega}(k)\mathcal{J}(k)\omega(k) \\ m(k)\tilde{\omega}(k)\tilde{\omega}(k)p(k) \end{pmatrix} \tag{2.2}$$

$\phi(k,k-1)$ denotes the *spatial transformation operator* from $\mathcal{O}_{k-1}$ to $\mathcal{O}_k$ and is given by

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

Here $\tilde{x} \in \mathbb{R}^{3\times 3}$ denotes the skew-symmetric cross- product matrix associated with the vector x, and $l(k,k-1)$ is the vector from frame $\mathcal{O}_k$ to frame $\mathcal{O}_{k-1}$. '1'bough 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 [8] lead to compact expressions for the equations of motion and other kcy dynamical quantities. The vector $\theta \triangleq [\theta^*(1), \ldots \theta^*(n)]^* \in \mathbb{R}^{\mathcal{N}}$ denotes the vector of generalized coordinates for the manipulator. similarly, we define the vectors of generalized velocities $\beta \in \mathbb{R}^{\mathcal{N}}$ and generalized (hinge) forces $T \in \mathbb{R}^{\mathcal{N}}$ for the manipulator. The vector of spatial velocities V is defined as $V \triangleq [V^*(1) \ldots V^*(n)]^* \in \mathbb{R}^{6n}$. The vector of spatial accelerations is denoted $\alpha \in \mathbb{R}^{6n}$, that of the Coriolis accelerations by $a \in \mathbb{R}^{6n}$, the link centrifugal forces by $b \in \mathbb{R}^{6n}$, and the link interaction spatial forces by $f \in \mathbb{R}^{6n}$. The equations of motion for the serial manipulator can be written as follows [8]:

$$V = \phi^* H^* \beta \tag{2.3a}$$
$$\alpha = \phi^*[H^* \dot{\beta} + a] \tag{2.3b}$$
$$f = \phi[M\alpha + b] \tag{2.3c}$$
$$T = Hf = \mathcal{M}\dot{\beta} + C \tag{2.3d}$$

where

$$\mathcal{M} \triangleq H\phi M \phi^* H^* \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}} \tag{2.4a}$$
$$\mathcal{C} \triangleq H\phi[M\phi^* a + b] \in \mathbb{R}^{\mathcal{N}} \tag{2.4b}$$

and $H \triangleq \text{diag}\{H(k)\} \in \mathbb{R}^{\mathcal{N}\times 6n}$, $M \triangleq \text{diag}\{M(k)\} \in \mathbb{R}^{6n\times 6n}$,

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

$$\phi \triangleq (I_{6n} - \mathcal{E}_\phi)^{-1} = \begin{pmatrix} I_6 & 0 & \ldots & 0 \\ \phi(2,1) & I_6 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi(n,1) & \phi(n,2) & \ldots & I_6 \end{pmatrix} \in \mathbb{R}^{6n\times 6n} \tag{2.5}$$

with

$$\phi(i,j) \triangleq \phi(i,i-1)\cdots\phi(j+1,j) \in \mathbb{R}^{6\times 6} \quad \text{for } i > j$$

The spatial transformation operator $\phi(k,j)$ has the form $\begin{pmatrix} I_3 & \tilde{l}(k,j) \\ 0 & I_3 \end{pmatrix} \in \mathbb{R}^{6\times 6}$ with $l(k,j) \in \mathbb{R}^3$, denoting the vector from the k^{th} to the j^{th} body frame. $\mathcal{M}$ is the *mass matrix* of the manipulator and the vector C contains the velocity dependent Coriolis and centrifugal] hinge forces. Eq. (2.4a) is referred to [8] as the Newton-Euler factorization of the mass matrix M.

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

Operator factorization and inversion techniques developed in [8] are used to obtain a closed form spatial operator expression for M^{-1}. First, we define the *articulated body inertia* quantities $P(.)$, $D(.)$, $G(.)$, $K(.)$, $T(.)$, $P^+(.)$ and $\psi(.,.)$ for the manipulator links using the following tip- to base recursive algorithm [7, 8]:

Algorithm 2

$$
\begin{array}{l}
P^{+}(0) = 0 \\
for\ k = 1 \cdots n \\
\quad P(k) = \phi(k, k-1)P^{+}(k-1)\phi^{*}(k,k-1) + M(k) \\
\quad D(k) = H \quad U \quad H \ * \\
\quad G(k) = P(k)\, H^{*}(k)\, D^{-1}(k) \\
\quad K(k+1,k) = \phi(k+1, k)G(k) \\
\quad \overline{\tau}(k) = I_6 - G(k)H(k) \\
\quad P^{+}(k) = \overline{\tau}(k)\, P(k)\overline{\tau}^{*}(k) = \overline{\tau}(k)P(k) \\
\quad \psi(k+1,k) = \phi(k+1,k)\overline{\tau}(k) \\
end\ loop
\end{array}
\tag{2.6}
$$

The operator $P \in \mathbb{R}^{6n\times 6n}$ is defined as a block diagonal matrix with the k^{th} diagonal element being $P(k) \in \mathbb{R}^{6\times 6}$. The quantities in Eq. (2.6) arc also used to dc.fine the following spatial operators:

$$
\begin{array}{rcl}
D & \stackrel{\Delta}{=} & HPH^{*} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}} \\
G & \stackrel{\Delta}{=} & PH^{*}D^{-1} \in \mathbb{R}^{6n\times\mathcal{N}} \\
K & \stackrel{\Delta}{=} & \mathcal{E}_{\phi}G \in \mathbb{R}^{6n\times\mathcal{N}} \\
\overline{\tau} & \stackrel{\Delta}{=} & I - GH \in \mathbb{R}^{6n\times 6n} \\
P^{+} & \stackrel{\Delta}{=} & \overline{\tau}P\overline{\tau}^{*} = \overline{\tau}P \in \mathbb{R}^{6n\times 6n} \\
\mathcal{E}_{\psi} & \stackrel{\Delta}{=} & \mathcal{E}_{\phi}\overline{\tau} \in \mathbb{R}^{6n\times 6n}
\end{array}
\tag{2.7}
$$

The operators D, G and $\overline{\tau}$ are all Mock diagonal. Even though K and $\mathcal{E}_{\psi}$ are not block diagonal matrices, their only non zero block elements are the elements $K(k,k-1)$'s and $\psi(k,k-1)$'s respectively along the first subdiagonal. It is easy to verify from Eq. (2.6) that P satisfies the Riccati equation

$$
M = P - \mathcal{E}_{\psi}P\mathcal{E}_{\psi}^{*} = P - \mathcal{E}_{\phi}P\mathcal{E}_{\psi}^{*}
\tag{2.8}
$$

Now define the lower--triangular operato, $\psi \in \mathbb{R}^{6n\times 6n}$ as

$$
\psi \stackrel{\Delta}{=} (I - \mathcal{E}_{\psi})^{-1}
\tag{2.9}
$$

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

$$
\psi(i,j) \stackrel{\Delta}{=} \begin{cases} \psi(i,i-1)\ldots\psi(j+1,j) & \text{for } i > j \\ I_6 & \text{for } i = j \\ 0 & \text{for } i < j \end{cases}
$$

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

Lemma 2.1 below describes an alternative (to Eq. (2,4a)) operator factorization of $\mathcal{M}$ as well as an expression for its inverse. Further discussion and the proof of the lemma can be found in references [8, 9].

Lemma 2.1: The operator expression for the innovations Factorization of the mass matrix M and its inverse are as follows:

$$M = [I + H\phi K]D[I + H\phi K]^* \quad (2.10a)$$

$$[I + H\phi K]^{-1} = [I - H\psi K] \quad (2.10b)$$

$$M^{-1} = [I - H\psi K]^* D^{-1}[I - H\psi K] \quad (2.10c)$$

∎

The factor $[I + H\phi K] \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is square while the factor D is block diagonal. Thus, the factorization in Lemma 2.1 can also be regarded as a closed-form LDL^* factorization of M. The closed form operator expression for the inverse of the factor $[I + H\phi K]$ is given by Eq. (2.10 b). It leads to the closed form operator expression for the inverse of the mass matrix M in Eq. (2.10c). This factorization can be regarded as a closed--form L^*DL factorization of M^{-1}.

2.2 Articulated Body Forward Dynamics Algorithm

Using Eq. (2,10c) in Eq. (2.3d) and after some simplifications [9] we obtain the following operator expression for generalized accelerations vector $\dot{\beta}$:

$$\dot{\beta} = [I - H\psi K]^* D^{-1}[T - H\psi(KT + Pa + b)] - K^*\psi^* H^* a \quad (2.11)$$

This expression forms the basis for the O(N) *articulated body inertia forward dynamics algorithm [7,8]* for manipulators. First we decompose Eq. (2.11) into the following sequence of sub-expressions:

$$z = \mathcal{E}_\phi z^+ + Pa + b \quad (2.12a)$$

$$z^+ = z + G\epsilon \quad (2.12b)$$

$$\epsilon = T - Hz \quad (2.12c)$$

$$\nu = D^{-1}\epsilon \quad (2.12d)$$

$$\alpha^+ = \mathcal{E}_\phi^* \alpha \quad (2.12e)$$

$$\dot{\beta} = \nu - G^*\alpha^+ \quad (2.12f)$$

$$\alpha = \alpha^+ + H^*\dot{\beta} + a \quad (2.12g)$$

The expressions in Eq. (2.12) map into the following computational algorithm:

Algorithm 3

$$\left|\begin{array}{l} \qquad z(0) = 0 \\ \textit{for } k = 1 \cdots n \\ \quad z(k) = \phi(k, k-1) z^{+}(k-1) + b(k) + P(k) a(k) \\ \quad \epsilon(k) = T(k) - H(k) z(k) \\ \quad z^{+}(k) = z(k) + G(k) \epsilon(k) \\ \quad \nu(k) = D^{-1} \epsilon(k) \\ \textit{end loop} \end{array}\right. \qquad (2.13a)$$

$$\left|\begin{array}{l} \qquad \alpha^{+}(n+1) = 0 \\ \textit{for } k = n \cdots 1 \\ \quad \alpha^{+}(k) = \phi^{*}(k+1, k) \alpha(k+1) \\ \quad \dot{\beta}(k) = \nu(k) - G^{*}(k) \alpha^{+}(k) \\ \quad \alpha(k) = \alpha^{+}(k) + H^{*}(k) \dot{\beta}(k) + a(k) \\ \textit{end loop} \end{array}\right. \qquad (2.13b)$$

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

1. Use the first base-to-tip recursion in Algorithm 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 to compute the articulated body inertias $P(k)$ etc.
3. Use the first tip-to-base recursion in Algorithm 3 to compute the residual forces $z(k)$ etc.
4. Use the latter base-to-tip recursion in Algorithm 3 to compute the link and joint accelerations $\alpha(k)$ and $\dot{\beta}(k)$ for all the links.

3 Free–Flying Space Manipulators

The above analysis is applicable to both ground-based as well as space manipulators. The model for the space manipulator remains the same as above – with bodies 1 through $n-1$ being the links of the manipulator, and body n denoting its base-body. The n^{th} hinge (between the base body and the inertial frame) has 6 degrees of freedom, and the hinge map matrix $H(n)$ for this hinge is the 6 x 6 identity matrix, i.e. $H(n) = I_6$. The joint velocities for this hinge are just the components of the 6-dimensional spatial velocity of the base body, that is, $\beta(n) = V(n)$.

The operator factorization and inversion result as well as the $O(\mathcal{N})$ articulated body inertia forward dynamics algorithm described earlier also extend to space manipulators as well. In Algorithm 2, the n^{th} recursion step (for the base-body) simplifies to

$$D(n) = P(n), \quad G(n) = I_6, \quad \tau(n) = I_6, \quad \bar{\tau}(n) = 0 \qquad (3.1)$$

For the residual force computations in Algorithm 3,

$$\epsilon(n) = T(n) - z(n), \quad \dot{\beta}(n) = \nu(n) = P^{-1}(n)\, \epsilon(n), \ \alpha(n) = \nu(n) + a(n) \qquad (3.2)$$

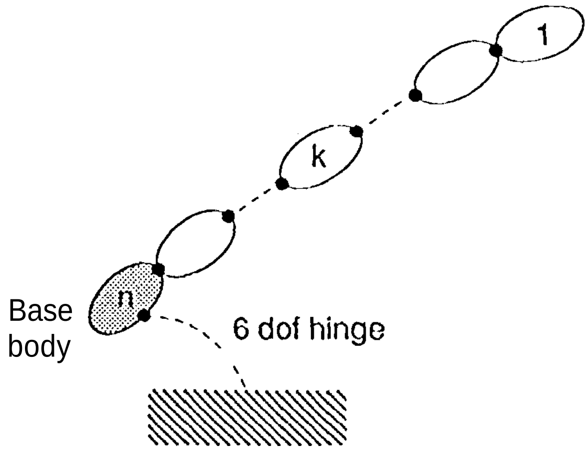

Figure 2: A space manipulator with a 6 degree of freedom hinge at the base-body

For a terrestrial manipulator, the choice for the base-body is unique and is the link attached to the ground. In contrast, one of the interesting aspects of space manipulators is an inherent symmetry that arises from the non-unique choice for base--bed y. There may well be a reason from physical considerations to prefer one of the bodies as base--body. However, analytically there is no inherent reason for selecting one body over the other as base -body. While, above we have designated link n as the base-body and the remaining links as forming the manipulator, from a modeling and algorithmic point of view we can in fact, choose other links as the base-body as well. Appendix A establishes this rigorously, and shows that the operator factorization results carry over completely even when links other than link n are used as the base–body, and thus so do the various algorithms. In the following section we look specifically at the case where the outermost link (link 1) is chosen as the "base" body. We regard and refer to this case as *dual* to the conventional case where link n serves as the base--body.

We henceforth use the subscript "p" for the articulated body inertia and residual force quantities defined in Section 2 where link n was used as the base-body The quantities associated with the dual approach (with link 1 as base body) will have the subscript "s". Thus the vectors β_p and β_s denote the generalized velocity vectors with the base--body being link n and link 1 respectively.

3.1 Dynamics Algorithm with Link 1 as Base Body

Figure 3 shows the configuration of the space manipulator with link 1 chosen as the base body. As discussed in Appendix A, the six components of the base body's spatial velocity vector form part of the generalized velocity coordinates for the manipulator. '1'bus, when the base -body is moved from link n to link 1, the components of $\beta(n)$ in β_p are replaced by the components $\beta(0) \triangleq V(1)$ to obtain the new generalized velocity coordinate vector β_s. Note that the $H_s, \phi_s, \mathcal{M}_s$ etc. operators and the mass matrix M, are all different in the dual formulation, Nevertheless, as discussed in Appendix A, the essential operator formulation and results including those about the operator factorization

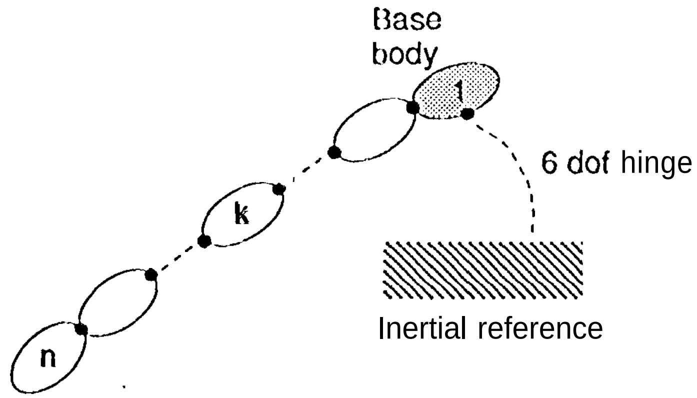

Figure 3: A space manipulator with the outer most link as base-body

and inversion of the new mass matrix still carry through and thus so does the articulated body inertia forward dynamics algorithm.

The difference in the articulated body inertia algorithm that arises within the dual formulation is that what was the tip before now is the base and vice versa. 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 $S^+(k)$, and the other dual quantities by the subscript s, i.e. D_s, G_s, $\tilde{\tau}_s$ etc. The quantity dual to P^+ is given by

$$S \triangleq \tilde{\tau}_s S^+$$

Note that the reversal in direction also reverses the sense of orientation of the internal hinge axes, and therefore all of their hinge map matrices H(.) reverse sign. The recursions corresponding to Eq. *(2.6) in* the dual articulated body inertia are as follows:

Algorithm 4

$$\begin{array}{l} S(n) = 0 \\ for\ k = n-1 \cdots 0 \\ \quad S^+(k) = \phi(k,k+1)[S(k+1) + M(k+1)]\phi^*(k,k+1) \\ \quad D_s(k) = H(k)S^+(k)H^*(k) \\ \quad G_s(k) = -S^+(k)H^*(k)D_s^{-1}(k) \\ \quad K_s(k-1,k) = \phi(k-1,k)G_s(k) \\ \quad \tilde{\tau}_s(k) = I_6 + G_s(k)H(k) \\ \quad S(k) = \tilde{\tau}_s(k)S^+(k) \\ \quad \psi_s(k-1,k) = \phi(k-1,k)\tilde{\tau}_s(k) \\ end\ loop \end{array} \tag{3.3}$$

The quantities $\phi(k-1, k)$ are defined as

$$\phi(k-1,k) = \phi^{-1}(k,k-1) = \begin{pmatrix} I_3 & -\tilde{l}(k,k-1) \\ 0 & I_3 \end{pmatrix} \tag{3.4}$$

The residual- forces recursion dual to Eq. (2.13) is as follows:

Algorithm 5

$$
\left|\begin{array}{lll}
 & z_s(n) = 0 & \\
\textbf{for } k & = n-1 \cdots 0 & \\
z_s^+(k) & = \phi(k,k+1)[z_s(k+1) + b(k+1) + \{S(k+1) + M(k+1)\}a_s(k+1)] & \\
\epsilon_s(k) & = T(k) + H(k)z_s(k) & \\
z_s(k) & = z_s^+(k) + G_s(k)\epsilon_s(k) & \\
\nu_s(k) & = D^{-1}\epsilon_s(k) & \\
\textbf{end } loop & &
\end{array}\right. \quad (3.5a)
$$

$$
\left\{\begin{array}{lll}
 & \alpha(0) = 0 & \\
\textbf{for } k & = 0 \cdots n-1 & \\
\alpha(k) & = \phi^*(k-1,k)\alpha^+(k-1) & \\
\beta(k) & = \nu_s(k) - G_s^*(k)\alpha(k) & \\
\alpha^+(k) & = \alpha(k) - H^*(k)\beta(k) + a_s(k) & \\
end\ loop & &
\end{array}\right. \quad (3.5b)
$$

Here the Coriolis acceleration $a_s(k)$ is given by

$$
a_s(k) \stackrel{\Delta}{=} -\phi^*(k+1,k)a_p(k+1) \quad (3.6)
$$

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 given in the following lemma,

***Lemma** 3.1: We have*

$$
f = P^+\alpha^+ + z_p^+ = P(\alpha - a_p) + z_p \quad (3.7)
$$

Proof: From Eq. (2.12) *it follows that*

$$
\begin{aligned}
\alpha &\stackrel{2.12f,2.12g}{=} \alpha^+ + H_p^*\nu_p - \tau_p^*\alpha^+ + a_p \stackrel{2.12e}{=} \mathcal{E}_{\psi_p}^*\alpha_p + H_p^*\nu_p + a_p \\
&\stackrel{2.9}{=} \psi_p^*(H_p^*\nu_p + a_p)
\end{aligned} \quad (3.8)
$$

Thus

$$
\tilde{\tau}_p^*\alpha_p^+ \stackrel{2.12e}{=} \mathcal{E}_{\psi_p}^*\alpha_p \stackrel{3.8}{=} \tilde{\psi}_p^*(H^*\nu_p + a_p) \quad (3.9)
$$

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

$$
z_p^+ \stackrel{2.12a,2.12b}{=} \mathcal{E}_{\phi_p} z_p^+ + G_p\epsilon_p + Pa_p + b_p \stackrel{2.5}{=} \phi_p^*(G_p\epsilon_p + Pa_p + b_p) \quad (3.10)
$$

Therefore it follows that,

$$
\begin{aligned}
f &\overset{2.3c}{=} \phi_p(M_p\alpha + b) \\
&\overset{3.8}{=} \phi_p P\psi_p(H_p^*\nu_p + a_p) + \phi_p b \\
&\overset{2.8}{=} \left[\phi_p P + P\tilde{\psi}_p^*\right](H_p^*\nu_p + a_p) + \phi_p b \\
&\overset{3.9}{=} P^+\alpha^+ + \phi_p P(H_p^*\nu_p + a_p) + \phi_p b \\
&\overset{2.7,2.12d,3.10}{=} P^+\alpha^+ + z^+
\end{aligned}
$$

The latter half of *Eq. (3. 7) can be proved similarly.* ∎

The dual form of Eq. (3.7) is given by:

$$-f = S^+(\alpha^+ - a_s) + z_s^+ = S\alpha + z_s \tag{3.11}$$

The above relationships give alternative ways of expressing the inter-link spatial force j 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 accelerations $\alpha(k)$ of the links are given by:*

$$
\begin{aligned}
\alpha(k) &= -[P(k)+S(k)]^{-1}[z_p(k) + z_s(k) - P(k)a_p(k)] \\
\alpha^+(k) &= -[P^+(k)+S^+(k)]^{-1}[z_p^+(k) + z_s^+(k) - S^+(k)a_s(k)]
\end{aligned}
\tag{3.12}
$$

Proof: Combine together Eq. (3.7] and Eq. (3. 11) to obtain the above. ∎

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 acceleration is given by the following pair of expressions:

$$\dot{\beta}(k) = \nu_p(k) - G_p^*(k)\,\alpha^+(k) = \nu_s(k) - G_s^*(k)\alpha(k) \tag{3.13}$$

We now discuss the physical interpretation of Eq. (3.1 2). If we return to the regular forward dynamics algorithm wc see that the key point in the algorithm is the computation of the 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 can be computed immediately followed by the rest of the accelerations.

Let us now consider the intermediate link k as the base-body as shown in Figure 4. For this choice, the components of the spatial velocity vector $V(k)$ for the k^{th} body now 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 [1 1]. 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

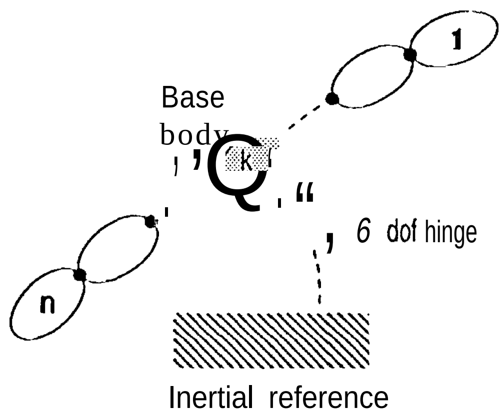

Figure 4: A space **manipulator** with an **intermediate** link as base--body

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* and proceeding independently towards link k. The first recursion is for the $P(.)$'s with link n as the base-body, while the second recursion is for the S(.)'s with link 1 as the base-body. These recursions come to a stop when the k^{th} link is reached.

The results from the two recursions along the two arms are "gathered" together at the k^{th} link to form the quantity $P(k)$ +- S(k). 'l'his quantity is nothing but the articulated body inertia of the whole manipulator as seen at frame O_k with the k^{th} link regarded as the base-body. Similarly $z_p(k)$ + z,(k) is the residual force at frame O_k with the k^{th} link being the base-body. The relationship with the superscript + quantities is basically the same but with reference frame O_k^+. Thus Eq. (3,12) is essentially telling us that one could 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 accelerations.

3.3 The Base–Invariant Decoupled Forward Dynamics Algorithm

The discussion in the previous section illustrated as to how each link in the manipulator could in principle be regarded as a base-body. This arbitrariness in the choice of the base-body reflects the inherent symmetry of space manipulators. However, while any link will do, the articulated body inertia forward dynamics algorithm requires a specific choice for the base- body. A specific choice for the base-body breaks this inherent symmetry. We show here that it is unnecessary and to break this symmetry. wc can reformulate the algorithm so as to preserve and take advantage of the symmetry. *The key to this is to treat every link in the manipulator as a base- body.* With this in mind, wc make usc of Lemma 3.2 to propose the following reformulated forward dynamics algorithm:

Algorithm 6 *1. Use the first part of Algorithm 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* $P(.)$ *etc.* and the residual *forces* $z_r(.)$ *in a recursion from* link n *to link 1 using Algorithms* 2 *and 3.* (b) *Simultaneously compute the dual articulated body quantities* $S^+(.)$ *etc. and the dual residual forces* $z_s(.)$ *in a* recursion *from* link 1 *to link n* using *Algorithms 4 and* 5.

3. *For the* k^{th} link, *compute the link spatial acceleration* $\alpha(k)$ *using Lemma 3.2, and the* hinge *acceleration* $\dot{\beta}$ **using** *Eq. (3.13). 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. Further simplification and decoupling of the algorithm is possible by choosing a new non-minimal set for the generalized coordinates for the manipulator and is discussed next.

3.3.1 Simplifications Using Non–Minimal Coordinates

We have so far used the hinge angles plus the six base--body positional and orientation coordinates as the generalized coordinates for the space 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 the decoupled dynamics algorithm.

Recall that 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 velocities 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. In fact, these steps can be entirely dispensed with.

We now regard each link as an independent rigid body system in its own right. The manipulator thus consists of a collection of these links connected together via hinges. For each link, we choose its six 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 and only $\mathcal{N}$ of them are truly independent. However, with the use these coordinates, transformations between the hinge and spatial coordinate domains are unnecessary. The modified decoupled dynamics algorithm is as follows:

Algorithm 7 1. *Use each links' generalized velocities* (i.e. *spatial velocity* $V(k)$) *to compute the Coriolis and gyroscopic terms a and b for* itself. *These can be computed completely independently for each link.*

2. *(a) Compute the articulated body quantities* $P(.)$ *etc. and the* residual *forces* $z_p(.)$ *in a recursion from link 1 to* link *n using Algorithms 2 and 3. (b) Simultaneously compute the dual*

articulated body quantities $S^+(.)$ *etc. and the dual residual forces* $z,(.)$ *in a* recursion *from* link *n to link* 1 using *Algorithms 4 and 5.*

s. *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 some of the computations in Algorithm 6. However, the price paid here with the use of redundant coordinates is that the integration method now involves a differential-algebraic equation rather t h an an ordinary differential equation.

3.3.2 Computational Issues

Like the articulated body forward dynamics algorithm, the decoupled dynamics algorithm is of $O(\mathcal{N})$ complexity. Since it involves a pair of articulated body recursions, it is computationally more expensive. However, since many of the computations are decoupled and independent of each other, it is useful for parallel implementation. In Algorithm 7, 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. With 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 and integrator to update the state of its link. Thus, when implemented on parallel hardware, there is a significant amount of computational speedup over the conventional articulated body inertia forward dynamics algorithm.

3.3.3 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-intcrvd 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 consists in part of a causal Kalman filter to obtain optimal filtered estimates of the state process. Once the filtered estimates arc obtained, an anti-causal smoothing recursion is used to compute the smoothed estimates. These causal and anti-causal recursions arc 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 the observations over the finite interval arc all available together. It consists of running two independent Kahnan 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 onc is based upon all the 'future" observations. The optimal smoothed estimate is 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 decoupled dynamics algorithm described here.

3.4 Extensions to Tree-Topology Manipulators

The extension of the conventional articulated body inertia forward dynamics algorithm in Section 2.2 to tree-topology systems has been described in reference [11]. The recursive computations now take on a gather/scatter structure. Thus recursions proceeding from the tips towards the base gather inputs together from the incoming branches as they progress. 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. Thus the conventional $O(\mathcal{N})$ articulated body inertia algorithm consists of sequential recursions in a manner similar to that for serial–chain systems.

As in the case of serial-chain space manipulators, tree-topologyj space manipulators also possess a symmetry arising from the non-unique choi cc for the base-bod y. The decoupled dynamics algorithm for tree-topology space manipulators is illustrated in Figure 5 and takes into account the fact there are more than two extremal bodies. Articulated body inertia recursions start from each

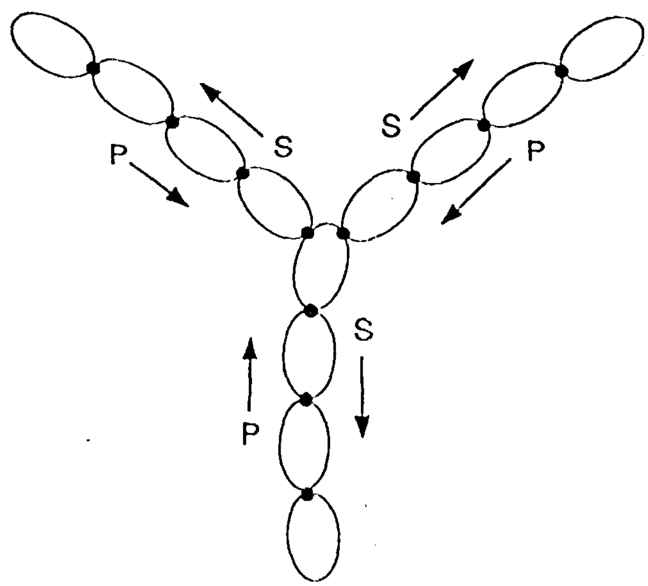

Figure 5: The structure of the decoupled dynamics algorithm for tree topology space manipulators

extremal body and proceed inwards. These recursions correspond to the P articulated body inertia recursions in the serial-manipulator case. As they approach branch points, their contributions are gathered together and accumulated by the link in common, and passed on to the other branches. There are also separate recursions corresponding to the S articulated body inertia recursions which treat every extremal body as a base body. These recursions proceed towards each of the extremal bodies accumulating the contributions from the branches as they go along. The overlap in the computations among these recursions is such that along any of the serial chain sub–branches there arc only precisely two recursions proceeding in opposite directions. Lemma 3.2 is still valid and is used to compute the spatial accelerations of each of the links.

The overall structure of the decoupled dynamics algorithm remains the same as in Algorithm 7. The only change is to the articulated body inertia computations in Step 2 as discussed above.

4 Base–Invariant Operational Space Inertia

We now look at the role of the *operational space inertia* [16, 17] in the dynamics of space manipulators. The operational space inertia inertia A(1) $\in \mathbb{R}^{6\times 6}$, 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

$$\text{A-'(1)} = J_p(1)\mathcal{M}_p^{-1}J_p^*(1) \tag{4.1}$$

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

$$J_p(1) = B^*(1)\,\phi^* H^*, \quad \text{where} \quad B(1) \triangleq \text{col}\Big\{I_6\delta(i,1)\Big\}_{i=1}^n \in \mathbb{R}^{6n\times 6}$$

with 6(., .) denoting the Kronecker delta function.

We generalize this notion of the operational space inertia to all links on the manipulator. Thus the operational space inertia for the k^{th} link, A(k), is the effective mass matrix of the manipulator as seen at the $\mathcal{O}_k$ frame. Analogous to the earlier definition, its inverse is given by the expression

$$\Lambda_p^{-1}(k) = J_p(k)\,\mathcal{M}_p^{-1}J_p^*(k) \tag{4.2}$$

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

$$J_p(k) = \text{B“(k)(\#)*H*}, \quad \text{where} \quad B(k) \triangleq \text{col}\Big\{I_6\delta(i,k)\Big\}_{i=1}^n \in \mathbb{R}^{6n\times 6} \tag{4.3}$$

It has been shown in [8] that

$$\phi_p^* H_p^* \mathcal{M}_p^{-1} H_p \phi_p = \psi_p^* H_p^* D_p^{-1} H_p \psi_p \triangleq \Omega_p \in \mathbb{R}^{6n\times 6n} \tag{4.4}$$

It follows therefore that

$$\Lambda_p^{-1}(k) \overset{4.3,4.4}{=} J_p(k)\mathcal{M}_p^{-1}J_p^*(k) = B^*(k)\Omega_p B(k) \tag{4.5}$$

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 $\mathcal{M}_p$ depend on the choice of base--body, Ω_p is in fact independent of this choice. To see this, let us usc the subscript k to denote the usc of the k^{th} link as the base body. We have from Lemma A.1 in Appendix A that

$$J_k(k) \overset{4.3}{=} B^*(k)\,\phi_p^*\,()H_p^*\mathcal{T}_{n,k}, \quad \text{and} \quad \mathcal{M}_k \overset{4.6}{=} \mathcal{T}_{n,k}^*\,\mathcal{M}_p\,\mathcal{T}_{n,k}$$

"..

Therefore,

$$\begin{aligned}
\Omega_k &\stackrel{\Delta}{=} \phi_k^* H_k^* \mathcal{M}_k^{-1} H_k \phi_k \\
&= B^*(k)\phi_p^* H_p^* \mathcal{T}_{n,k}[\mathcal{T}_{n,k}^* \mathcal{M}_p \mathcal{T}_{n,k}]^{-1} \mathcal{T}_{n,k}^* H_p \phi_p B(k) \\
&= \phi_p^* H_p^* \mathcal{M}_p^{-1} H_p \phi_p \\
&\stackrel{4.4}{=} \Omega_p
\end{aligned}$$

Since $\Omega_p \cdot \Omega_k$ for all k, this quantity is independent of the choice of the base body and wc drop the subscript from Ω altogether. This fact, taken together with Eq. (4.2) also establishes the invariance of the operational space inertia A(k) with respect to the choice of the base-body.

Using a new block diagonal operator $\Upsilon \in \mathbb{R}^{6n \times 6n}$, it has been shown in references [8; 17] that Ω can be decomposed as

$$\Omega = \Upsilon + \psi_p^* \Upsilon + \Upsilon \psi_p \tag{4.6}$$

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

$$\begin{cases}
\qquad \Upsilon^+(n) = 0 \\
\text{for } k = n \cdots 1 \\
\qquad \Upsilon(k) = \tau_p^*(k)\Upsilon^+(k)\tau_p(k) + H_p^*(k) D_p^{-1}(k) H_p(k) \\
\qquad \Upsilon^+(k-1) = \phi^*(k, k-1)\Upsilon(k)\phi(k, k-1) \\
\textbf{end loop}
\end{cases} \tag{4.7}$$

Superficially, it appears from Eq. *(4.7)* that wc should be using the subscript p on Υ and its components to indicate their dependence on the choice of link n *as* the base-body. However, we do not do so because Υ is in fact independent of the choice of the base-body. This fact is obvious once wc realize that the three terms on the right hand side of Eq. (4.6) arc block diagonal, block strictly upper–triangular and block strictly lower-triangular respectively. Since Ω is independent of the choice of base--body, therefore so also arc Y and its components. From the definition in Eq. (4.5) and the decomposition in Eq. (4.6), it follows that

$$\Lambda^{-1}(k) \stackrel{4.5,4.6}{=} B^*(k)[\Upsilon + \psi_p^*\Upsilon + \Upsilon\psi_p]B(k) = B^*(k)\Upsilon B(k) = \Upsilon(k) \tag{4,8}$$

That is, Y(k) is the inverse of the operational space inertia A(k). For ground-based manipulators, Y(k) **is** singular for the first 5 links connected to the base. The singularity reflects the fact that there arc directions along which spatial forces induce no motion in the manipulator. In contrast, at the base body (link *n)* of a space-manipulator

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

'1'bus, for space manipulators, $\Upsilon(n)$ is always invertible. Indeed, it can be shown that T(k) is invertible for all k. 'l'he invertibility property reflects the fact that any spatial force at any point on the space manipulator will cause a non-zero acceleration of the free flying manipulator.

Even though wc have seen that. the value of the $\Upsilon(.)$'s do not depend of the choice of a base body, the computational scheme in Eq. (4 .7) 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 T(k) which makes usc of the dual articulated body inertia quantities computed using Algorithm 4:

$$\left\{ \begin{array}{l} \qquad \mathrm{T}(0) = 0 \\ \textbf{for } k = 0 \cdots n-1 \\ \qquad \Upsilon^{+}(k) = \tilde{\tau}_{s}^{*}(k)\, \Upsilon(k) \tilde{\tau}_{s}(k) + H^{*}(k)\, D_{s}^{-1}(k) H(k) \\ \qquad \Upsilon(k+1) = \phi^{*}(k, k+1)\, \Upsilon^{+}(k)\, \phi(k, k+1) \\ \text{end loop} \end{array} \right. \tag{4.9}$$

Both Eq. *(4.7)* and Eq. (4.9) describe computational schemes consisting of a recursion one way to compute the articulated body inertia quantities, followed by a recursion in the opposite direction to compute the T(k) 's. The lemma below shows that the symmetry of space manipulators in fact allows us to dispense with these algorithms and express the $\Upsilon(k)$'s directly using the articulated body inertia $P(k)$ and its dual S+(k).

Lemma 4.1:

$$[\Upsilon(k)]^{-1} = P(k) + S(k) \quad (4.10a)$$
$$[\Upsilon^+(k)]^{-1} = P^+(k) + S^+(k) \quad (4.10b)$$

Proof: It follows from Eq. (2.6), Eq. (3.3) and Eq. (4.7) that Eq. (4.10a) is true for $k = i$ if and only if Eq. (4.10b) is true for $k = i-1$. We have that $\Upsilon^{-1}(n) = P(n)$. Since by definition $S(n) = 0$, this implies that Eq. (4.10a) holds for $k = n$. Thus Eq. (4.10b) holds for $k = n-1$. We use proof by induction to establish the general result.

Assume that Eq. (4.10b) holds for a certain k. Then from Eq. (2.6), Eq. (4.7) and that $S(k)H^*(k) = 0$, it follows that

$$\begin{aligned}
[P(k) + S(k)]\Upsilon(k) &\overset{4.7}{=} [P(k) + S(k)][\bar{\tau}_p^*(k)\Upsilon^+(k)\bar{\tau}_p(k) + H_p^*(k) D_p^{-1}(k)H_p(k)] \\
&\overset{2.6}{=} P^+(k)\Upsilon^+(k)\bar{\tau}_p^*(k) + \tau_p(k) + S(k)\bar{\tau}_p^*(k)\Upsilon^+(k)\bar{\tau}_p(k) \\
&= P^+(k)\Upsilon^+(k)\bar{\tau}_p(k) + \tau_p(k) + S(k)\Upsilon^+(k)\bar{\tau}_p(k) \\
&= I_6 - \tau_s(k)S^+(k)\Upsilon^+(k)\bar{\tau}_p(k) \\
&= I_6 - \tau_s(k)[I_6 - P^+(k)\Upsilon^+(k)]\bar{\tau}_p(k) \\
&= I_6 - \tau_s(k)\bar{\tau}_p(k)[I_6 - P^+(k)\Upsilon^+(k)\bar{\tau}_p(k)] \\
&= I_6
\end{aligned}$$

The last step follows from the fact that

$$\tau_s(k)\tau(k) = \tau_s(k) - G_s(k)H(k)G_p(k)H(k) = \tau_s(k) - G_s(k)H(k) = 0$$

Thus if Eq. (4.10b) is *true for a certain k*, Eq. (4.10a) is *also true for the same k. When combined with the* **earlier** *result, it implies that* Eq. (4.10a) is *also true for $k-1$.* This *establishes the induction* **process** since *we have seen that* Eq. (4.10a) *is in fact true for* $k = n$.

∎

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

$$\Lambda(k) \overset{4.8,4.10a}{=} P(k) + S(k) \quad (4.11)$$

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

Algorithm 8 *1. (a) Compute the articulated body quantities $P(.)$ recursively from link 1 to link n using Algorithm 2. (b) simultaneously compute the dual articulated body quantities $S^+(.)$ recursively from link n to link 1 using Algorithm 4.*

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

Unlike the algorithms in Eq. (4.7) and Eq. (4.9), Algorithm 8 has a decoupled structure arising from the symmetry of the space manipulators. The two sequential recursions in the earlier algorithms arc 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 space manipulator is simply obtained by summing up the P and S articulated body inertia terms at the link,

5 Conclusions

Space manipulators possess a symmetry not normally 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 was to treat 'every link" as a base body. It has been shown that key dynamical quantities can be obtained by combining results from independent articulated body inertia computations. 'l'he usc of non–minimal coordinates to further decouple the forward dynamics algorithm has also been discussed. The extension of the decoupled algorithm to tree-topology space manipulators is straightforward. 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

[I] }1. 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, 011), 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] A. Jain and G. Rodriguez, "An Analysis of the Kinematics and Dynamics of Under-Actuated Manipulators," *IEEE Transactions on Robotics and Automation*, 1992, In press.

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

[8] 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.

[9] 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.

[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] 0. 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.

Appendix A: Transformation of the Base Body to Link k

The spatial velocity of the base body contributes six of the generalized velocity coordinates for the manipulator, The generalized velocities vector β with link n as the base body consists of

$\{\beta(1), \cdots, \beta(n-1), V(n)$ where we have used the fact that

$$\beta(n) = V(n)$$

In this section we will use the base-body index as a subscript to denote the choice of the base-body. Thus β above will now be denoted β_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 $\beta_k \in \mathbb{R}^{\mathcal{N}}$ are given by

$$\beta_k \triangleq \begin{pmatrix} \beta(1) \\ \vdots \\ \beta(n-1) \\ V(k) \end{pmatrix} \tag{A1}$$

Lemma A.1 below defines the nonlinear transformation $\mathcal{T}(.,.)$ which transforms between the β_k and β_n coordinates. First we rewrite H_n in the following partitioned form

$$H_n = \begin{pmatrix} \mathcal{H} & 0 \\ 0 & I_6 \end{pmatrix}, \quad \text{where} \quad \mathcal{H} \triangleq \text{diag}\{H(i)\}_{i=1}^{n-1} \in \mathbb{R}^{(\mathcal{N}-6)\times 6(n-1)} \tag{A.2}$$

Lemma A1: ***The transformation*** *map* $\mathcal{T}_{k,n}$ *is such that*

$$\beta_k = \mathcal{T}_{k,n}\beta_n, \quad where \quad \mathcal{T}_{k,n} = \begin{pmatrix} I_{n-1} & 0 \\ X_k\mathcal{H}^* & \phi^*(n,k) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}} \tag{A.3}$$

with

$$X_k \triangleq [0, \cdots, I_6, \phi^*(k+1,k), \ldots, \phi^*(n-1,k)] \in \mathbb{R}^{6\times n-1} \tag{A.4}$$

The *inverse transformation* $\mathcal{T}_{n,k}$ *such that* $\beta_n = \mathcal{T}_{n,k}\beta_k$ *is given by*

$$\mathcal{T}_{n,k} \triangleq \mathcal{T}_{k,n}^{-1} = \begin{pmatrix} I_{n-1} & 0 \\ -\phi^*(k,n)X_k\mathcal{H}^* & \phi^*(k,n) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}} \tag{A.5}$$

Proof: ***From*** *Eq.* (2.3a) ***it*** *follows that*

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

From this follows the expression *for* $\mathcal{T}_{k,n}$ ***in*** *Eq. (A .3). The expression for its inverse,* $\mathcal{T}_{n,k}$, ***follows*** *quite simply* ***from matrix manipulation.*** ∎

The kinetic energy is given by

$$\frac{1}{2}\beta_n^*\mathcal{M}_n\beta_n = \frac{1}{2}\beta_k^*\mathcal{T}_{n,k}^*\mathcal{M}_n\mathcal{T}_{n,k}\beta_k$$

therefore, the mass matrix $\mathcal{M}_k$ in the β_k set of coordinates is given by

$$\mathcal{M}_k = \mathcal{T}^*_{n,k}\mathcal{M}_n\mathcal{T}_{n,k} = \mathcal{T}^*_{n,k}H_n\phi_n M_n\phi^*_n H^*_n\mathcal{T}_{n,k} \tag{A.6}$$

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 wc look closely at the derivation of the factorization and inversion results for the mass matrix and the articulated body inertia forward dynamics algorithm, wc see that the kcy properties in the derivation were that the mass matrix has a Newton-Euler operator factorization as in Eq. (2.4a) and that H^*_n and M_n arc diagonal, and ϕ_n has the form

$$\phi_n = [I - \mathcal{E}_{\phi_n}]^{-1}$$

where $\mathcal{E}_{\phi_n}$ is a nilpotent matrix. 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 wc need to define a ncw velocity coordinates vector β°_k obtained by reordering the components of β_k as follows:

$$\beta^\circ_k = \mathcal{P}\beta_k = \begin{pmatrix} \beta(1) \\ \vdots \\ \beta(k-1) \\ V(k) \\ \beta(k) \\ \vdots \\ \beta(n-1) \end{pmatrix}, \quad \text{where} \quad \mathcal{P} \triangleq \begin{pmatrix} I_{(k-1)} & 0 \\ 0 & \begin{pmatrix} 0 & I_6 \\ I_{6(n-k-1)} & 0 \end{pmatrix} \end{pmatrix} \tag{A.7}$$

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

$$\mathcal{M}^\circ_k = \mathcal{P}\mathcal{M}_k\mathcal{P}^* \overset{A.6}{=} \mathcal{P}\mathcal{T}^*_{n,k}\mathcal{M}_n\mathcal{T}_{n,k}\mathcal{P}^* = \mathcal{P}\mathcal{T}^*_{n,k}H_n\phi_n M_n\phi^*_n H^*_n\mathcal{T}_{n,k}\mathcal{P}^* \tag{A.8}$$

Lemma A.2: *The mass matrix $\mathcal{M}^\circ_k$ has the following operator factorization:*

$$\mathcal{M}_k = H_k\phi_k M_k\phi^*_k H^*_k \tag{A.9}$$

where

$$\begin{aligned} H_k &\triangleq \mathcal{P}H\mathcal{P}^* \\ \phi_k &\triangleq [I - \mathcal{E}_{\phi_k}]^{-1} \\ M_k &\triangleq \Delta_\phi M_n\Delta_\phi^* \\ \Delta_\phi &\triangleq \begin{pmatrix} I_{6k} & 0 \\ 0 & \operatorname{diag}\Big\{\phi(i-1,i)\Big\}_{i=k+1}^{n} \end{pmatrix} \in \mathbb{R}^{6n\times 6n} \end{aligned} \tag{A.10}$$

where

$$\mathcal{E}_{\phi_k} \triangleq \begin{pmatrix} Y_1 & Y_3 \\ 0 & Y_2 \end{pmatrix} \tag{A.11}$$

is nilpotent with $Y_1 \in \mathbb{R}^{6k\times 6k}$, $Y_2 \in \mathbb{R}^{6(n-k)\times 6(n-k)}$ *and* $Y_3 \in \mathbb{R}^{6k\times 6(n-k)}$ *defined as*

$$Y^* \triangleq \begin{pmatrix} \mathbf{0} & 0 & 0 & 0 & 0 \\ \phi(2,1) & 0 & \cdot & \cdot & \cdot & 0 & \mathbf{0} \\ 0 & \phi(3,2) & \cdot & \cdot & \cdot & 0 & 0 \\ \vdots & \vdots & \ddots & & \cdot & \vdots \\ 0 & 0 & \cdots & \phi(k,k-1) & \mathbf{0} \end{pmatrix}$$

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

and

$$Y_3 \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ I_6 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

Proof: *We have*

$$H^*\mathcal{T}_{n,k} \overset{A.5}{=} \begin{pmatrix} \mathcal{H}^* & \mathbf{0} \\ -\phi^*(k,n)\,X_k\mathcal{H}^* & \phi^*(k,\ n) \end{pmatrix}$$

$$= \quad QH^*, \quad \text{where } Q \triangleq \begin{pmatrix} I_{6(n-1)} & \mathbf{0} \\ -\phi^*(k,n)X_k & \phi^*(k,\ n) \end{pmatrix} \in \mathbb{R}^{6n\times 6n} \tag{A.12}$$

Let

$$e_k \triangleq \left[\operatorname{col}\left\{I_6\delta(i,k)\right\}_{i=1}^{n-1}\right]^* = [0,\cdots,I_6,\cdots,0], \quad \text{and} \quad \hat{I} \triangleq [I_{6(n-1)},\mathbf{0}]$$

where 6(., .) *denotes the* ***Kronecker*** *delta function. We have that*

$$[X_k,\ \phi^*(n,\ k)] = e_k\phi_n^* \tag{A.13}$$

Therefore,

$$Q^{-1} = \begin{pmatrix} I_{6(n-1)} & \mathbf{0} \\ X_k & \phi^*(n,k) \end{pmatrix} = \begin{pmatrix} \hat{I} \\ e_k\phi_n^* \end{pmatrix} \tag{A.14}$$

$$\phi_n^* Q \quad = \quad \left[Q^{-1}\phi_n^{-*}\right]^{-1} \overset{A.14}{=} \begin{pmatrix} \hat{I}\phi_n^* \\ e_k \end{pmatrix}^{-1} = \begin{pmatrix} \hat{I} - \hat{I}\mathcal{E}_\phi^* \\ e_k \end{pmatrix}^{-1} \tag{A.15}$$

At the ***component level,***

$$\left(\begin{matrix}\hat{I} - \hat{I}\mathcal{E}_\phi^* \\ e_k\end{matrix}\right) = \begin{pmatrix} I_6 & -4"(2,1) & o & 0 & . & . & . & 0 & . & . & . & 0 & 0 \\ 0 & I_6 & -\phi^*(3, & 2) & 0 & \cdots & 0 & \cdots & 0 & & & & 0 \\ \vdots & . & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \cdots & I & & & & & -\phi^*(n, n-1) \\ 0 & 0 & 0 & 0 & \cdots & I_6 & \cdots & 0 & & & & & 0 \end{pmatrix}$$

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

$$\mathcal{P}\left(\begin{matrix}\hat{I} - \hat{\mathcal{E}_\phi}^* \\ e_k\end{matrix}\right)\Delta_\phi^* = [I - \mathcal{E}_{\phi_k^*}] \tag{A.16}$$

In the *above, the permutation matrix* $\mathcal{P}$ is *used to* ***transform*** (' - $ ' ") *into a* ***tri-diagonal matrix form, while*** Δ_ϕ^* ***normalizes the terms along the diagonal to*** 1_6***. It*** is easy ***to verify that*** $\mathcal{E}_{\phi k}$ ***is nilpotent, and hence (I –*** $\mathcal{E}_{\phi_k}$***) is invertible.*** *We denote this inverse as* $\phi(k + 1, k)$. *Thus*

$$\left(\begin{matrix}\hat{I} - \hat{\mathcal{E}_\phi}^* \\ e_k\end{matrix}\right)^{-1} \overset{A.16}{=} \Delta_\phi^* \phi_k \mathcal{P} \tag{A.17}$$

Therefore we *have that*

$$\phi^* H^* \mathcal{T}_{n,k} \mathcal{P}^* \overset{A.12}{=} \phi^* \mathcal{Q} H^* \mathcal{P}^* \overset{A.15,A.17}{=} \Delta_\phi^* \phi_k^* \mathcal{P} H^* \mathcal{P}^* = \Delta_\phi^* \phi_k^* H_k^* \tag{A.18}$$

Thus

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

This ***establishes the result.*** ∎

Note that ϕ_k is no longer fully lower triangular, but 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 wc now scc that $\mathcal{M}_k^o$ has the necessary Newton–Euler operator factorization, the operator inversion results corresponding to Lemma *2.1* can bc obtained here as well. It can be shown that

$$\{\mathcal{M}_k^o\}^{-1} = [I - H_k \psi_k K_k]^* D_k^{-1} [I - H_k \psi_k K_k] \tag{A.19}$$