JPL Publication 88-11

Recursive Mass Matrix Factorization and Inversion

An Operator Approach to Open- and Closed-Chain Multibody Dynamics

G. Rodriguez
K. Kreutz

March 15, 1988

NASA
National Aeronautics and Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL Publication 88-11

Recursive Mass Matrix Factorization and Inversion

An Operator Approach to Open- and Closed-Chain Multibody Dynamics

G. Rodriguez
K. Kreutz

March 15, 1988

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

ABSTRACT

This report advances a linear operator approach for analyzing the dynamics of systems of joint-connected rigid bodies. It is established that the mass matrix $\mathcal{M}$ for such a system can be factored as $\mathcal{M} = (I + H\Phi L)D(I + H\Phi L)^T$. This yields an immediate inversion $\mathcal{M}^{-1} = (I - H\Psi L)^T D^{-1}(I - H\Psi L)$, where H and Φ are given by known link geometric parameters, and L, Ψ and D are obtained recursively by a spatial discrete-step Kalman filter and by the corresponding Riccati equation associated with this filter. The factors $(I + H\Phi L)$ and $(I - H\Psi L)$ are lower triangular matrices which are inverses of each other, and D is a diagonal matrix. This factorization and inversion of the mass matrix leads to recursive algorithms for forward dynamics based on spatially recursive filtering and smoothing. The primary motivation for advancing the operator approach is to provide a better means to formulate, analyze and understand spatial recursions in multibody dynamics. This is achieved because the linear operator notation allows manipulation of the equations of motion using a very high-level analytical framework (a spatial operator algebra) that is easy to understand and use. Detailed lower-level recursive algorithms can readily be obtained by inspection from the expressions involving spatial operators. The report consists of two main sections. In Part I, the problem of serial-chain manipulators is analyzed and solved. Extensions to a closed-chain system formed by multiple manipulators moving a common task object are contained in Part II. To retain ease of exposition in the report, only these two types of multibody systems are considered. However, the same methods can be easily applied to arbitrary multibody systems formed by a collection of joint-connected rigid bodies.

CONTENTS

Tables

Figures

Part I

RECURSIVE MASS MATRIX FACTORIZATION AND INVERSION FOR OPEN-CHAIN SYSTEMS

ABSTRACT

In this paper, some of the power of an operator formulation for manipulator dynamics is demonstrated by readily obtaining several distinct recursive forward dynamics algorithms. Starting from the Newton-Euler Formulation of the equations of motion for an N-link serial manipulator with a fixed base, the manipulator dynamics are re-expressed in a linear operator form. Using an operator factorization and inversion technique, applied to the manipulator mass matrix $\mathcal{M}$, an $O(N)$ iterative forward dynamics algorithm is then obtained. Additional $O(N)$ algorithms are derived by the use of certain operator identities.

The extension to a fully mobile base is discussed in an appendix. The manipulator mass matrix is shown to have a factorization $\mathcal{M} = (I + H\Phi L)D(I + H\Phi L)^T$ yielding an immediate inversion $\mathcal{M}^{-1} = (I - H\Psi L)^T \, D^{-1} \, (I - H\Psi L)$, where H and Φ are given by known link geometric parameters, and L, Ψ and D are obtained via a discrete-step Riccati equation driven by the link masses. The factors $(I + H\Phi L)$ and $(I - H\Psi L)$ are lower triangular matrices which are inverses of each other, and D is a diagonal matrix.

1. INTRODUCTION

In this paper, manipulator dynamics are given an operator interpretation which greatly facilitates the ability to manipulate and solve expressions arising from considerations involving robot kinematics, dynamics and control. Furthermore, the analytical results of such manipulations can often be mapped directly onto an equivalent recursive algorithm. To illustrate this, the robot dynamical equations will be given an operator formulation, after which several recursive forward dynamics algorithms will be derived. In the body of this paper, the assumption is made that the base is fixed and that link joints are one degree of freedom revolute or sliding joints. In an Appendix B, the base is allowed to be fully mobile, and it is discussed how three degree-of-freedom spherical ball joints can be incorporated into the dynamics.

The ability to quickly and iteratively compute the forward dynamics of an N-link serial manipulator enables efficient simulation of arbitrary robot arms from knowledge only of the individual link geometric and mass properties and the nature of the link interconnections. An explicit analytical expression describing manipulator dynamics is not needed [1]. An $O(N)$ iterative algorithm for forward dynamics has been proposed in [2] and [3] by R. Featherstone. This algorithm is derived utilizing the Spatial Algebra, Kinematics, and Dynamics developed in [3] (which is related to the Motor and Screw Calculus). The algorithm of [2] and [3] first computes bias terms due to coriolis, centrifugal, gravity, and contact forces by means of the $O(N)$ iterative Newton-Euler algorithm for inverse dynamics [4,5]. It then subtracts this bias from the input joint moments (in the manner previously suggested by [1]) to obtain a "bias- free" robot dynamics equation. After this step, the algorithm proceeds to obtain joint-angle accelerations by an additional $O(N)$ iteration.

This paper extends the recent work of [6,7] to show that $O(N)$ iterative forward dynamics algorithms can be easily obtained by the use of techniques for solving linear operator equations by operator factorization [8−14]. After the recursive Newton- Euler formulation [4,5] of manipulator dynamics is restated in an operator form, concepts developed in [6], which extend the classical signal filtering theory of [8-14] to mechanical systems, are used to achieve the operator factorization. These algorithms are developed in the context of classical mechanics, and thereby avoid the need to learn the nonstandard Spatial Dynamics of [2,3]. In this paper, the focus is on the open chain serial link case. The extension to the closed-chain case is reported in the companion paper [15].

A forward dynamics algorithm is first developed which computes and subtracts out the coriolis, centrifugal, gravity, and contact force bias terms, exactly as discussed above for the algorithms in [1,3]. After this step, as a consequence of an operator factorization and inversion of the manipulator mass matrix, joint-angle accelerations are obtained in $O(N)$ iterations. This two-step algorithm is examined because it focuses on the key issues of mass matrix factorization and inversion. Alternative algorithms are then found by the use of certain operator identities. These include an algorithm which avoids the need for a preliminary bias computation and subtraction.

The operator formulation of manipulator dynamics developed here has great conceptual and practical power. A conceptual strength of this approach is that abstract terms or expressions involved in manipulator dynamics and control, such as those which arise from a Lagrangian analysis, often have operator interpretations which make them implicitly equivalent to tip-to-base or base-to-tip recursions. For example, the manipulator mass matrix, $\mathcal{M}$, will be shown to represent an unstated base-to-tip followed by a tip-to-base recursion. In fact, manipulator dynamics expressed in the Lagrangian formulation have an operator interpretation which results in exactly the Newton-Euler recursions, and the well-known equivalence of the Lagrangian and Newton-Euler formulations [16] is directly seen.

Abstract analytical expressions, involving quantities such as the mass matrix $\mathcal{M}$, can be manipulated with greater ease when viewed from the operator perspective. This is because the operator interpretation enhances the ability to manipulate such expressions by the use of operator identities which are given and derived in this paper. The abstract forms resulting from such manipulations can be interpreted as operator expressions which have equivalent recursive forms relating interlink forces and accelerations. This in turn can give insight into the physical meaning of the abstract expressions.

The practical value of the operator algebra results from the ability to take the recursive equivalent of any operator-interpreted expression (which is coordinate-free) and immediately obtain an implementable recursive algorithm by projecting the recursion into appropriate coordinate frames attached to the links or to an inertial reference point.

Although this paper focuses on obtaining forward dynamics algorithms, this by no means exhausts the applications of the operator approach. For example, [6,7] derive efficient iterative methods for obtaining the entire robot mass matrix and its inverse, as well as producing exact symbolic expressions for these matrices. In [15,17], which develop forward dynamics algorithms for closed-chain systems, it is shown how $J\mathcal{M}^{-1}J^T$, with J the manipulator Jacobian, can be recursively derived. The inverse matrix, i.e., $[J\mathcal{M}^{-1}J^T]^{-1}$, is exactly the operational space manipulator mass matrix of [18]. In a similar way, recursions for computing other terms involved in the operational space formulation of manipulator dynamics can be obtained. Additionally, it is anticipated that manipulator control laws under development [19] can be shown to have recursive implementations once an appropriate operator reformulation is performed.

For clarity and brevity of expression, the algorithms of this paper are given in coordinate-free form. Of course, in actual applications the algorithm calculations can be performed in fixed link frames utilizing link-to-link transformations as is done in [1,4].

2. PROBLEM FORMULATION AND STATEMENT

Consider a rigid N-link serial manipulator as illustrated in Figs. 1.1 and 1.2, with the quantities defined in Table 1.1. The links and joints are numbered in an increasing order that goes from the tip of the system toward the base. Joint N is the last in the sequence, and it connects link N to a base. In the body of the paper, the base is assumed to be immobile. The extension to a mobile base may be found in Appendix B. The base is referred to as "link N + 1". Joint k in the sequence connects links k and $k+1$. The point O_o can be selected to be any arbitrary point in link 1. Note that link and joint numbers increase toward the base of the system. This differs from the more common numbering approach in which the numbers increase toward the tip.

This ordering allows one to think of sequentially moving from joint 1 to joint N as going "forward" and moving from joint N to joint 1 as going "backward". In this paper, an algorithm which processes link data by iterating from $k = 1$ to $k = N$ is then called "causal" while an algorithm which iterates from $k = N$ to $k = 1$ is called "anticausal". A complete tip- to-base causal iteration or a complete base-to-tip anti-causal iteration is called a sweep.

The external environment is viewed as "link 0", and the arm can contact the environment at point O_o. Although $\theta(k)$ is defined to be rotational, the extension to the sliding joint case is simple and will be discussed later.

Note that axis $h(k)$ is associated with angle $\theta(k)$ and both are associated with link k. This feature is not true for the standard Denavit-Hartenberg formulation for numbering links, angle and axes discussed in [4]. This feature is true for the modified D-H formulation described in [2], and the scheme used in this paper can be considered to be that of [2] but with reversed link numbering.

By standard classical kinematical and dynamical considerations [4,5], the equations relating link velocities and accelerations and the link force/torque balance equations are readily obtained resulting in the following "recursive Newton-Euler" (RNE) equations. The "$\dot{}$" indicates time differentiation taken in an inertial frame. Also $\dot{v}(N+1) = (9.8m/s^2)e_3$ may be taken as a means to incorporate the effect of a uniform gravity loading on the links when determining link forces and moments, in which case $\dot{v}(k)$ is not the true link k linear acceleration. e_3 is defined as $e_3 = (0,0,1)^T$.

frame i [illegible] link i

RECURSIVE NEWTON-EULER EQUATIONS.

frame i [illegible] link i

z0

$$\omega(N+1) = 0; \quad v(N+1) = 0; \quad \dot{\omega}(N+1) = 0; \quad \dot{v}(N+1) = 0 \text{ or } (9.8m/2)e_3; \tag{2.1}$$

for $k = N, \ldots, 1$ loop

For sliding

kth

$$\omega(k) = \omega(k+1) + h(k)\dot{\theta}(k) \qquad \omega(k) = \omega(k+1) \tag{2.2}$$

(k+1) th

$$v(k) = v(k+1) + \omega(k+1) \times \ell(k+1,k) \qquad v(k) = v(k+1) + \omega(k+1) \times \ell(k+1,k) + R(k)\dot{\theta}(k) \tag{2.3}$$

$R\ \omega_k(k+1)$

$$\dot{\omega}(k) = \dot{\omega}(k+1) + h(k)\ddot{\theta}(k) + \omega(k+1) \times h(k)\dot{\theta}(k) \tag{2.4}$$

$$\dot{v}(k) = \dot{v}(k+1) + \dot{\omega}(k+1) \times \ell(k+1,k) + \omega(k+1) \times [\omega(k+1) \times \ell(k+1,k)] \tag{2.5}$$

$$\dot{v}^c(k) = \dot{v}(k) + \dot{\omega}(k) \times p(k) + \omega(k) \times [\omega(k) \times p(k)] \tag{2.6}$$

end loop;

$$\omega(0) = \omega(1); \quad v(0) = v(1) + \omega(1) \times \ell(1,0); \quad \dot{\omega}(0) = \dot{\omega}(1);$$

$$\dot{v}(0) = \dot{v}(1) + \dot{\omega}(1) \times \ell(1,0) + \omega(1) \times [\omega(1) \times \ell(1,0)] \tag{2.7}$$

$$F(0) = F_{ext}; \quad N(0) = N_{ext} \tag{2.8}$$

for $k = 1, \ldots, N$ loop;

$$N^c(k) = I^c(k)\dot{\omega}(k) + \omega(k) \times I^c(k) \cdot \omega(k); \tag{2.9}$$

$$F^c(k) = m(k)\dot{v}^c(k) \tag{2.10}$$

$$F(k) = F(k-1) + F^c(k) \tag{2.11}$$

$$N(k) = N(k-1) + \underline{\ell(k,k-1)} \times F(k-1) + p(k) \times F^c(k) + N^c(k) \tag{2.12}$$

$$T(k) = h^T(k) N(k) \tag{2.13}$$

end loop;

$$N_{k-1}(k-1) = N_{k}(k-2) + \ell_{k-1}(k-1,k-2) \times F_{k-1}(k-2) + p(k-2) \times F^c(k-2) + N^c(k-2)$$

The quantities referred to as "spatial velocity" $V(k)$, "spatial force" $f(k)$, "spatial acceleration" $\alpha(k)$, and "spatial inertia" $M(k)$ are now defined. ($\tilde{v}y \equiv v \times y$).

$$V(k) = \begin{pmatrix} \omega(k) \\ v(k) \end{pmatrix}; \quad \alpha(k) = \begin{pmatrix} \dot{\omega}(k) \\ \dot{v}(k) \end{pmatrix}; \quad f(k) = \begin{pmatrix} N(k) \\ F(k) \end{pmatrix}; \tag{2.14}$$

$$M(k) = \begin{pmatrix} I(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)I \end{pmatrix}. \tag{2.15}$$

I denotes the identity operator. Note that the spatial quantities defined by (2.14) and (2.15) are not those given in [2,3], although they are quite similar. This distinction is quite significant because it implies that only the rules of ordinary matrix algebra are needed here. Thus the nonstandard Featherstone spatial algebra of [2] is not used in this paper. Define also

$$\phi(k+1,k) = \begin{pmatrix} I & \tilde{\ell}(k+1,k) \\ 0 & I \end{pmatrix}; \quad H^T(k) = \begin{pmatrix} h(k) \\ 0 \end{pmatrix}. \tag{2.16}$$

With the above definitions (2.1)-(2.6) become

$$V(N+1) = 0$$

$$V(k) = \phi^T(k+1,k)V(k+1) + H^T(k)\dot{\theta}(k) \tag{2.17}$$

and

$$\alpha(N+1) = 0$$

$$\alpha(k) = \phi^T(k+1,k)\alpha(k+1) + H^T(k)\ddot{\theta}(k) + a(k) \tag{2.18}$$

where $a(k)$, the "bias acceleration", is given by

$$a(k) = \begin{pmatrix} \omega(k+1) \times h(k)\dot{\theta}(k) \\ \omega(k+1) \times [\omega(k+1) \times \ell(k+1,k)] \end{pmatrix} = \begin{pmatrix} \omega(k+1) \times \omega(k) \\ \omega(k+1) \times [\omega(k+1) \times \ell(k+1,k)] \end{pmatrix}. \tag{2.19}$$

Note from (2.17) that $\phi^T(k,k-1)$ is precisely the link k Jacobian operator which relates rates at point $O(k)$ to rates at point $O(k-1)$ [5].

Equations (2.8)−(2.13) can be recast as

$$\begin{pmatrix} N(k) \\ F(k) \end{pmatrix} = \begin{pmatrix} I & \tilde{\ell}(k,k-1) \\ 0 & I \end{pmatrix} \begin{pmatrix} N(k-1) \\ F(k-1) \end{pmatrix} + \begin{pmatrix} I & \tilde{p}(k) \\ 0 & I \end{pmatrix} \begin{pmatrix} N^c(k) \\ F^c(k) \end{pmatrix} \tag{2.20}$$

and

$$\begin{pmatrix} N^c(k) \\ F^c(k) \end{pmatrix} = \begin{pmatrix} I^c(k) & 0 \\ -m(k)\widetilde{p}(k) & m(k)I \end{pmatrix} \begin{pmatrix} \dot{\omega}(k) \\ \dot{v}(k) \end{pmatrix} + \begin{pmatrix} \omega(k) \times I^c(k) \cdot \omega(k) \\ m(k)\omega(k) \times [\omega(k) \times p(k)] \end{pmatrix}. \tag{2.21}$$

Inserting (2.21) into (2.20) then results in

$$f(k) = \phi(k, k-1)f(k-1) + M(k)\alpha(k) + b(k) \tag{2.22}$$

$$f(0) = f_{ext}; \quad T(k) = H(k)f(k)$$

where $b(k)$, the "bias spatial force", is given by

$$b(k) = \begin{pmatrix} \omega(k) \times I(k) \cdot \omega(k) \\ m(k)\omega(k) \times [\omega(k) \times p(k)] \end{pmatrix}. \tag{2.23}$$

To arrive at (2.22)–(2.23), the facts that $I(k) = I^c(k) - m(k)\widetilde{p}^2(k)$ and $\omega(k) \times I^c(k) \cdot \omega(k) + m(k)p(k) \times [\omega(k) \times [\omega(k) \times p(k)]] = \omega(k) \times I(k)\omega(k)$ were used. Assuming, for now, no gravity loading, equations (2.17),(2.18)-(2.19), and (2.22)–(2.23) result in the following iterative equations.

RECURSIVE SPATIAL DYNAMICS (RSD).

$$V(N+1) = 0; \quad \alpha(N+1) = 0; \tag{2.24}$$

for $k = N, \ldots, 1$ loop

$$V(k) = \phi^T(k+1, k)V(k+1) + H^T(k)\dot{\theta}(k) \tag{2.25}$$

$$\alpha(k) = \phi^T(k+1, k)\alpha(k+1) + H^T(k)\ddot{\theta}(k) + a(k) \tag{2.26}$$

end loop

$\alpha(k-1) = \phi^T(k, k-1)\alpha(k) + H^T(k-1)\ddot{\theta}(k-1) + a(k-1)$

$$V(0) = \phi^T(1,0)V(1); \quad \alpha(0) = \phi^T(1,0)\alpha(1) + a(0); \tag{2.27}$$

$$f(0) = f_{ext} \tag{2.28}$$

for $k = 1, \ldots, N$ loop

$$f(k) = \phi(k, k-1)f(k-1) + M(k)\alpha(k) + b(k) \tag{2.29}$$

$$T(k) = H(k)f(k); \tag{2.30}$$

$+ \phi(k, k-1) \cdot m(k-1)\alpha(k-1)$

end loop;

Note that $V(0)$ and $\alpha(0)$ are the manipulator tip spatial velocity and acceleration and $f(0)$ is the tip spatial force acting on the external environment. Note also that $a(k) = a[V(k+1), V(k)]$ and $b(k) = b[V(k)]$ so that at the k^{th} iteration $a(k)$ and $b(k)$ can be computed from available quantities.

$f(k) = \phi(k, k-1)\,(f(k-1) + m(k-1)\alpha(k-1)$
$+ m(k)\alpha(k)) + b(k)$

If a joint k is a sliding joint rather than rotational, the RSD equations should be modified by taking

$$H^T(k) = \begin{pmatrix} 0 \\ h(k) \end{pmatrix} \tag{2.31}$$

instead of (2.16) and taking

$$a(k) = \begin{pmatrix} 0 \\ \omega(k+1) \times [\omega(k+1) \times \ell(k+1,k) + h(k)\dot{\theta}(k)] \end{pmatrix} =$$

$$\begin{pmatrix} 0 \\ \omega(k+1) \times [\omega(k+1) \times \ell(k+1,k) + \omega(k)] \end{pmatrix} \tag{2.32}$$

instead of (2.19). Here $\theta(k)$ now denotes a sliding degree of freedom. Thus (2.14)–(2.16), (2.19), (2.23), (2.24)–(2.30), (2.31) and (2.32) completely describe the kinematics and dynamics of an arm with both sliding and rotational joints.

Assume henceforth that $[\theta(k), \dot{\theta}(k)]$ are known. Given this knowledge, the forward dynamics problem is to obtain $\ddot{\theta}(k)$ from known inputs $T(k)$ and the RSD equations (2.24)–(2.30).

3. PROBLEM STATEMENT IN OPERATOR FORM

Define $\ell(i,j) = O(j) - O(i)$ to be the vector from $O(i)$ to $O(j)$. Define also

$$\phi(i,j) = \begin{pmatrix} I & \tilde{\ell}(i,j) \\ 0 & I \end{pmatrix}. \tag{3.1}$$

The matrix $\phi^T(i,j)$ is the Jacobian which relates the spatial velocity at point i to spatial velocity at point j. It is known [6] that $\phi(i,j)$ obeys the "group properties"

$$\phi(i,i) = I; \quad \phi^{-1}(i,j) = \phi(j,i); \quad \phi(i,k)\phi(k,j) = \phi(i,j). \tag{3.2}$$

These are also referred to as the state transition operator properties [20]. Note in particular that

$$\phi(i,k) = \phi(i,i-1)\cdots\phi(k+1,k). \tag{3.3}$$

Define

$$\phi = \begin{pmatrix} \phi(1,1) & 0 & \dots & 0 \\ \phi(2,1) & \phi(2,2) & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \phi(N,1) & \phi(N,2) & \dots & \phi(N,N) \end{pmatrix}, \tag{3.4}$$

$$H = diag[H(1), \dots, H(N)] \tag{3.5}$$

$$M = diag[M(1), \ldots, M(N)] \tag{3.6}$$

$$B^T = [\phi^T(1,0), 0, \ldots, 0] \tag{3.7}$$

and

$$E = [0, \ldots, 0, \phi^{\cancel{*}}(N+1, N)]^{T}. \tag{3.8}$$

Also, let

$$V = \begin{pmatrix} V(1) \\ \vdots \\ V(N) \end{pmatrix}; \quad \theta = \begin{pmatrix} \theta(1) \\ \vdots \\ \theta(N) \end{pmatrix}; \quad T = \begin{pmatrix} T(1) \\ \vdots \\ T(N) \end{pmatrix} \tag{3.9}$$

$$a = \begin{pmatrix} a(1) \\ \vdots \\ a(N) \end{pmatrix}; \quad \alpha = \begin{pmatrix} \alpha(1) \\ \vdots \\ \alpha(N) \end{pmatrix}; \quad f = \begin{pmatrix} f(1) \\ \vdots \\ f(N) \end{pmatrix}; \quad b = \begin{pmatrix} b(1) \\ \vdots \\ b(N) \end{pmatrix}. \tag{3.10}$$

Note that the block diagonal of ϕ is filled by the identity operator since $\phi(i,i) = I$.

From (2.25),

$$V(k) = \sum_{i=k}^{N} \phi^T(i,k) H^T(i) \dot{\theta}(i) \tag{3.11}$$

or, using (3.1)-(3.9),

$$V = \phi^T H^T \dot{\theta}. \tag{3.12}$$

Since (3.12) is just a restatement in "operator form" of (3.11), $\phi^T H^T$ must be an anticausal operator. That $\phi^T H^T$ is anticausal is equivalent to $\phi^T H^T$ being upper block triangular. Actually, ϕ^T is anticausal (upper block triangular) while H (being block diagonal) is "memoryless". This makes the product $\phi^T H^T$ anticausal. Recall that (3.12) being anticausal just means that V is obtained from an outward iteration from the base to the tip.

Since $V(0) = \phi^T(1,0)V(1)$,

$$V(0) = J\dot{\theta}$$

$$J = B^T \phi^T H^T. \tag{3.13}$$

J is nothing more than the "Jacobian Operator" which relates joint rates to the tip spatial velocity. J is seen to be an anticausal (*i.e.*, outward sweeping) operator. In (3.13) the dependence of the Jacobian Operator J upon the interlink Jacobians $\phi^T(i,j)$ is made explicit.

The entire manipulator dynamics can be expressed in a variety of operator forms useful for solving the forward dynamics problem by operator factorization techniques.

From (2.26),

$$\alpha(k) = \sum_{i=k}^{N} \phi^T(i,k)[H^T(i)\ddot{\theta}(i) + a(i)]. \tag{3.14}$$

With (3.1)-(3.10), this becomes

$$\alpha = \phi^T H^T \ddot{\theta} + \phi^T a. \tag{3.15}$$

The accelerations α are seen to be the result of the joint accelerations $H^T\ddot{\theta}$ and the bias accelerations a propagated from the base to the tip of the manipulator under the influence of the anticausal operator of interlink Jacobians, ϕ^T.

From (2.29) and (2.30)

$$f(k) = \sum_{j=1}^{k} \phi(k,j)[M(j)\alpha(j) + b(j) + \delta(j,1)\phi(1,0)f(0)] \tag{3.16}$$

$$T(k) = H(k)f(k)$$

or, with (3.15),

$$f = \phi(M\alpha + b + Bf(0)) \tag{3.17}$$

$$T = Hf; \quad \alpha = \phi^T H^T \ddot{\theta} + \phi^T a.$$

f is seen to be due to the causal propagation (due to the action of the lower block triangular operator ϕ) of the D'Alembert forces $M\alpha$, the bias forces b, and the tip forces $f(0)$, from the tip to the base. T is seen to be the memoryless projection of f onto the joint axis. From (2.27), $\alpha(0) = B^T\phi^T H^T\ddot{\theta} + B^T\phi^T a + a(0)$, and, since $\alpha(0) = J\ddot{\theta} + \dot{J}\dot{\theta}$, then $\dot{J}\dot{\theta} = B^T\phi^T a + a(0)$.

Equation (3.17) can also be written as

$$T = H\phi[M\alpha + b + Bf(0)] \tag{3.18}$$

$$\alpha = \phi^T H^T \ddot{\theta} + \phi^T a$$

where $M\alpha$ are the D'Alembert forces. Alternatively,

$$T = H\phi M\phi^T H^T\ddot{\theta} + H\phi[M\phi^T a + b + Bf(0)]$$

$$= H\phi[M\phi^T(H^T\ddot{\theta} + a) + b + Bf(0)] \tag{3.19}$$

or

$$
\begin{aligned}
T &= \mathcal{M}\ddot{\theta} + \mathcal{C} + J^T f(0) \\
\mathcal{M} &= H\phi M \phi^T H^T \\
\mathcal{C} &= H\phi(M\phi^T a + b) \\
J^T &= H\phi B.
\end{aligned} \tag{3.20}
$$

Eqs. (3.17)−(3.20) assume no uniform gravity loading. The effect of a uniform gravity loading on the manipulator is found by applying a pseudo-spatial acceleration to the base frame in the standard way [4]. This results in the reassignment

$$
a \rightarrow a + Eg, \quad g^T = (0^T, 9.8e_3^T) \tag{3.21}
$$

where E is given by (3.8).

Equations (3.17)-(3.20) give the operator formulations of the manipulator dynamics. In particular the operator form (3.17) says that T is obtained by an anticausal operation on $\ddot{\theta}$ and a, followed by a causal operation on α, b, and $f(0)$. This is, of course, precisely the recursive Newton-Euler algorithm of [4]. The fact that Eq. (3.20), which is of the form obtainable from a Lagrangian dynamical analysis, has an operator interpretation which results in the recursive Newton-Euler formulation reflects the well-known equivalence between Lagrangian dynamics and Newton-Euler dynamics [16].

The operator $\mathcal{M}$ will be called the "mass operator". The mass operator is seen in (3.20) to have a natural causal-memoryless-anticausal factorization. Note that $\mathcal{M}$ cannot be inverted by inverting the individual factors since $H\phi$ is nonsquare. In the next section, it will be shown that $\mathcal{M}$ has an alternative causal-memoryless-anticausal factorization for which the individual factors are invertible.

Equation (3.20) can be written as

$$
\mathcal{M}\ddot{\theta} = T' \tag{3.22}
$$

in which $T' = T - \widehat{T}$, with the "bias torques" $\widehat{T}$ given by

$$
\widehat{T} = H\phi[M\phi^T a + b + Bf(0)] = \mathcal{C} + J^T f(0). \tag{3.23}
$$

Computing $\widehat{T}$ recursively via the algorithm implicit in (3.23) allows one to work with the simpler system (3.22). Obtaining $\widehat{T}$ from (3.23) is equivalent to using the RNE algorithm implicit in (3.17), but with $\ddot{\theta} = 0$. This approach to simplifying from (3.20) to (3.22) is the standard one and is used in [1−3]. A choice exists then, to solve the forward dynamics problem by either solving (3.20) directly for $\ddot{\theta}$, or by solving the simpler system (3.22) for which the bias torques have been removed. The second choice is considered in the next section, where the focus is on operator factorization and inversion techniques appropriate for inverting the mass operator. The algorithmic alternative represented by (3.20) is developed and presented in Section 5.

4. A FORWARD DYNAMICS ALGORITHM BY MASS MATRIX/OPERATOR FACTORIZATION

Note that the operator $\mathcal{M}$ in the bias-free robot dynamics equations (3.22) is symmetric positive definite. This follows from (3.20). If such an operator can be modeled as the covariance of an output from a known, causal, and finite-dimensional linear system driven by white noise, then the operator can be factored and inverted efficiently by the use of standard techniques from filtering, detection, and estimation theory [8-14]. This factorization and inversion leads to a solution of (3.22). With $\mathcal{M} = H\phi M\phi^T H^T$, such a model is immediately at hand. Indeed, taking

$$Y = H\phi W \tag{4.1}$$

$$E(W) = 0; \quad E(WW^T) = M,$$

where $Y^T = [y_1^T, \ldots, y_N^T]$ and $W^T = [w_1^T, \ldots, w_N^T]$, results in $E(Y) = 0$ and

$$E(YY^T) = H\phi M\phi^T H^T = \mathcal{M}. \tag{4.2}$$

Here, $E(\cdot)$ is the statistical expectation operator [14]. Note that (4.1) is just a succinct way of saying

$$z(0) = 0;$$

for $k = 1, \ldots, N$ loop

$$z(k) = \phi(k, k-1)z(k-1) + w(k); \tag{4.3}$$

$$y(k) = H(k)z(k);$$

end loop;

where $E[w(k)] = 0$ and $E[w(k)w^T(k)] = M(k)$. The operator formulation of (4.1) and the state space (or algorithmic) formulation of (4.3) are entirely equivalent.

We have just seen that the model (4.1) results in the particular factorization $\mathcal{M} = H\phi M\phi^T H^T$. In fact, there are an infinity of possible factorizations for $\mathcal{M}$, each one associated with a particular model. This is discussed in Chapter 9 of [14]. All such models for $\mathcal{M}$ are related and form an equivalence class. One member of this class, in particular, is viewed as canonical. This is referred to as the "innovations model" or the "innovations representation". Under reasonably mild technical assumptions , which are met here, the innovations model for $\mathcal{M}$ is obtainable from any other available model for $\mathcal{M}$, such as the model (4.1). The factorization of $\mathcal{M}$ associated with the innovations model is a primary goal of this section.

The quantities $P(k)$, $D(k)$ and $L(k)$, $k = 1, \ldots, N$, are now defined by the following iteration

$$P(1) = M(1), \quad D(1) = H(1)P(1)H^T(1), \quad L(1) = \Phi(2,1)P(1)H^T(1)D^{-1}(1); \tag{4.4}$$

for $k = 2, \ldots, N-1$ loop

$$P(k) = \phi(k,k-1)[P(k-1) - P(k-1)H^T(k-1)D^{-1}(k-1)H(k-1)P(k-1)]\phi^T(k,k-1) + M(k); \tag{4.5}$$

$$\begin{aligned} D(k) &= H(k)P(k)H^T(k); \\ L(k) &= \phi(k+1,k)P(k)H^T(k)D^{-1}(k); \end{aligned} \tag{4.6}$$

end loop;

It can be shown that $P(k) = P^T(k) > 0$ for all k. Hence, the scalar $D(k)$ is always nonzero, and $D^{-1}(k) = 1.0/D(k)$ is guaranteed to exist.

Define $\psi(i,j)$ for $i \geq j$ by

$$\begin{aligned} \psi(k,k) &= I \\ \psi(k,k-1) &= \phi(k,k-1) - L(k-1)H(k-1) \\ \psi(i,j) &= \psi(i,i-1)\psi(i-1,i-2)\cdots\psi(j+1,j); \quad i \geq j \end{aligned} \tag{4.7}$$

Also define the memoryless operators

$$P = diag[P(1), \ldots, P(N)]; \quad D = diag[D(1), \ldots, D(N)]; \quad L = diag[L(1), \ldots, L(N)] \tag{4.8}$$

and the causal operators

$$\Phi = \begin{pmatrix} 0 & & & & & 0 \\ \phi(2,2) & 0 & & & & 0 \\ \phi(3,2) & \phi(3,3) & 0 & & & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \phi(N,2) & \phi(N,3) & \phi(N,4) & \cdots & \phi(N,N) & 0 \end{pmatrix} \tag{4.9}$$

$$\Psi = \begin{pmatrix} 0 & & & & & 0 \\ \psi(2,2) & 0 & & & & 0 \\ \psi(3,2) & \psi(3,3) & 0 & & & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ \psi(N,2) & \psi(N,3) & \psi(N,4) & \cdots & \psi(N,N) & 0 \end{pmatrix} \tag{4.10}$$

Note that Φ of (4.9) is distinct from ϕ of (3.4). Note also that: Φ and Ψ are strictly lower block-triangular (strictly causal); that D is symmetric positive definite; and that the block subdiagonals of Φ and Ψ are filled by the identity operator, since $\psi(k,k) = \psi(k,k) = I$.

It is now necessary to establish a few preliminary facts. Proof of the following Fact 1 is given in Appendix A. Alternative derivations can be found in [6],[7],[13] and [14]. Furthermore, the physical and statistical meanings of many of the quantities defined and used in this section and in Appendix A are discussed in [6,7].

FACT 4.1. *An alternative factorization of $\mathcal{M} = H\phi\mathcal{M}\phi^T H^T$ is*

$$\mathcal{M} = (I + H\Phi L)D(I + H\Phi L)^T \tag{4.11}$$

where $I + H\Phi L$ is causal (lower triangular) and D is memoryless and invertible.

<u>*Proof*</u> : See Appendix A.

The model for $\mathcal{M}$ which results in (4.11) is the innovations representation and is given by the following causal operation on the "innovations" $\epsilon^T = [\epsilon^T(1), \ldots, \epsilon^T(N)]$:

$E[Y Y^T] = M$

$$\boxed{Y = (I + H\Phi L)\epsilon}$$

$$E(\epsilon) = 0; \quad E(\epsilon\epsilon^T) = D \tag{4.12}$$

$$E(Y) = 0; \quad E(YY^T) = \mathcal{M}.$$

$W' \rightarrow M^{\frac{1}{2}} \rightarrow W' \rightarrow H\phi \rightarrow \varepsilon \rightarrow (I + H\Phi L) \rightarrow \varepsilon$

$Y = (I + H\Phi L)\epsilon$ can be restated in equivalent state space (or algorithmic) form as

$E[W W^T] = M$

$$\hat{z}(0) = 0; \quad \epsilon(0) = 0;$$

for $k = 1, \ldots, N$ loop

$= E[M^{\frac{1}{2}} W' W'^T M^{\frac{T}{2}}] = M^{\frac{1}{2}} =$

$$\hat{z}(k) = \phi(k, k-1)\hat{z}(k-1) + L(k-1)\epsilon(k-1); \tag{4.13}$$

$Y = H\phi M^{\frac{1}{2}} W'$

$= M$

$$y(k) = H(k)\hat{z}(k) + \epsilon(k);$$

$E[W W'^T] = I$

end loop;

It is known that the inverse of the innovations representation is a Kalman filter, viewed as a whitening filter. This statement will be clarified after the statement of another needed fact.

$E[Y Y^T] = E[H\phi W' W'^T \phi^T Y] = H\phi W'$

$= H\phi\Phi^T$

$Y = (I + H\Phi L)\varepsilon$

FACT 4.2. *The lower triangular operators $I + H\Phi L$ and $I - H\Psi L$ are mutually reciprocal*

$$(I + H\Phi L)^{-1} = I - H\Psi L. \tag{4.14}$$

<u>*Proof*</u> : See Appendix A or the comment below.

The relationship $\epsilon = (I + H\Phi L)^{-1}y = (I - H\Psi L)y$ has the state space form (taking $L(0)H(0) \equiv 0$

$$\hat{z}(0) = 0, \quad y(0) = 0;$$

for $k = 1, \ldots, N$ loop

$$\hat{z}(k) = [\phi(k, k-1) - L(k-1)H(k-1)]\hat{z}(k-1) + L(k-1)y(k-1); \tag{4.15}$$

$$\epsilon(k) = -H(k)\hat{z}(k) + y(k);$$

end loop;

Equation (4.15) is the Kalman filter associated with the model (4.3), written as a whitening filter which produces the "white-noise" innovations sequence $\epsilon(k)$. Hence, $\hat{z}(k) = E[z(k)/y(1), \ldots, y(k-1)]$, if $w(k)$ is gaussian. Also note that $\hat{z}(k)$ of (4.13) and (4.15) are identical, *i.e.*, $\hat{z}(k)$ of (4.13) is also a Kalman filter state. Comment: Note that since (4.15) follows from (4.13) by a simple rearrangement of terms, Fact 4.2 is easily shown to be true.

From Facts 4.1 and 4.2 immediately follows the key result of this paper.

THEOREM 4.1. *The operator $\mathcal{M}^{-1}$ can be factored as*

$$\mathcal{M}^{-1} = (I - H\Psi L)^T D^{-1}(I - H\Psi L). \tag{4.16}$$

Theorem 1 and the bias-free robot dynamics (3.22) immediately imply the following forward dynamics algorithm.

ALGORITHM FD1.

$$T' = T - H\phi[M\phi^T a + b + Bf(0)] \tag{4.17}$$

$$\ddot{\theta} = (I - H\Psi L)^T D^{-1}(I - H\Psi L)T'. \tag{4.18}$$

As discussed in the last section, (4.17) involves the $O(N)$ recursive Newton-Euler algorithm for $\ddot{\theta} = 0$. Equation (4.18) is given by a causal (tip-to-base) sweep to produce the normalized innovations $\nu = D^{-1}\epsilon$ with $\epsilon = (I - H\Psi L)T'$, followed by an anticausal (base-to-tip) sweep to produce the joint accelerations $\ddot{\theta}$. Since (4.17) additionally involves an anticausal sweep followed by a causal sweep, we call algorithm $FD1$ a "4-sweep" algorithm.

In state-space (algorithmic) form, (4.18) becomes

i) Causal Filtering of Bias Free Joint Moments:

$$\hat{z}(0) = 0, \quad T'(0) = 0; \tag{4.19}$$

for $k = 1, \ldots, N$ loop

$$\hat{z}(k) = [\phi(k, k-1) - L(k-1)H(k-1)]\hat{z}(k-1) + L(k-1)T'(k-1); \tag{4.20}$$

$$\epsilon(k) = T(k) - H(k)\hat{z}(k);$$

$$\nu(k) = D^{-1}(k)\epsilon(k);$$

end loop;

ii) Anticausal Smoothing of Weighted Residuals:

$$\lambda(N+1) = 0; \tag{4.21}$$

for $k = N, \ldots, 1$ loop

$\lambda(N) = H^T(N)\nu(N)$

$$\lambda(k) = [\phi(k+1, k) - L(k)H(k)]^T\lambda(k+1) + H^T(k)\nu(k); \tag{4.22}$$

$\ddot{\theta}(N) = \nu(N)$

$$\ddot{\theta}(k) = \nu(k) - L^T(k)\lambda(k+1);$$

$\ddot{\theta}(N-1) = \nu(N-1) - L^T(N-1)H^T(N)\nu(N)$

end loop;

Note that the algorithm (4.19)–(4.22) obviously has an operations count which is $O(N)$. What has not been made explicit in the above algorithm is the need to recursively construct $V, a, b, P, D,$ and L. For notational simplicity the need for these recursions is usually not stated, it being understood that they will be performed during appropriate sweeps of the algorithm. For example, V and a can be constructed during the 1[st] (anticausal) sweep of (4.17) indicated by $M\phi^T a$, while $b, P, D,$ and L can be constructed during the 2[nd] (causal) sweep of (4.17) indicated by $H\phi[\cdots]$.

5. ADDITIONAL FORWARD DYNAMICS ALGORITHMS FROM THE USE OF OPERATOR IDENTITIES

Various alternative formulations of the forward dynamics algorithms can be obtained by the use of operator identities which are derived in Appendix A. We will first show how a slight modification of the 4-Sweep algorithm of the last section leads to a 4-Sweep algorithm which produces the link spatial accelerations α and the tip spatial acceleration $\alpha(0)$. Define

$$\psi = \begin{pmatrix} \psi(1,1) & & & & 0 \\ \psi(2,1) & \psi(2,2) & & & 0 \\ \psi(3,1) & \psi(3,2) & \psi(3,3) & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ \psi(N,1) & \psi(N,2) & \psi(N,3) & \dots & \psi(N,N) \end{pmatrix} \tag{5.1}$$

$$S = \begin{pmatrix} 0 & & & 0 \\ I & 0 & & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{pmatrix}. \tag{5.2}$$

and note that

$$\Psi = \psi S. \tag{5.3}$$

IDENTITY 5.1. *For ϕ given by (3.4), Ψ given by (4.10), and ψ given by (5.1),*

$$(I - H\Psi L)H\phi = H\psi. \tag{5.4}$$

$H(I - \Psi L)\phi$

Proof : The proof may be found following the set of identities A.4 given in Appendix A. However, the following alternative proof is more transparent: A state-space representation of $X = H\phi W$ and $Y = (I - H\Psi L)X = (I - H\Psi L)H\phi W$ is

$$z(k) = \phi(k, k-1)z(k-1) + w(k); \quad x(k) = H(k)z(k);$$

$$\varsigma(k) = [\phi(k, k-1) - L(k-1)H(k-1)]\varsigma(k-1) + L(k-1)x(k-1);$$

$$y(k) = -H(k)\varsigma(k) + x(k);$$

yielding

$$z(k) - \varsigma(k) = [\phi(k, k-1) - L(k-1)H(k-1)][z(k-1) - \varsigma(k-1)] + w(k);$$

$$y(k) = H(k)[z(k) - \varsigma(k)];$$

or

$$Y = H\psi W.$$

This identity is now used to obtain the following 4-Sweep algorithm for link spatial accelerations and joint accelerations.

ALGORITHM FD2.

$$T' = T - H\phi[M\phi^T a + b + Bf(0)]; \tag{5.5}$$

$$\alpha = \psi^T H^T D^{-1}(I - H\Psi L)T' + \phi^T a; \tag{5.6}$$

$$\alpha(0) = B^T \alpha + a(0); \tag{5.7}$$

Proof : From (3.15), $\alpha = \phi^T H^T \ddot{\theta} + \phi^T a$. With (4.18), this gives

$$\alpha = \phi^T H^T (I - H\Psi L)^T D^{-1}(I - H\Psi L)T' + \phi^T a$$

$$= \psi^T H^T D^{-1}(I - H\Psi L)T' + \phi^T a.$$

The last step follows from (5.4). Eq. (5.7) follows from (3.7) and (2.27).

Note that λ of (4.22) obeys $\lambda = \psi^T H^T \nu$, $\nu = D^{-1}\epsilon$, $\epsilon = (I - H\Psi L)T'$. This gives $\lambda = \psi^T H^T D^{-1}(I - H\Psi L)T'$. Hence, with (4.18) and (5.3), $\ddot{\theta} = (I - L^T S^T \psi^t H^T)\nu = \nu - L^T S^T \lambda$. Also, from (5.6), $\alpha = \lambda + \phi^T a$. This means that a slight modification of the forward dynamics algorithm (4.19)–(4.22) allows the computation of the manipulator tip acceleration, $\alpha(0)$, in addition to $\ddot{\theta}$. This occurs by changing (4.21) and (4.22) to

$$\lambda(N+1) = 0; \quad \varsigma(N+1) = 0 \tag{4.21b}$$

for $k = N, \ldots, 1$ loop

$$\lambda(k) = [\phi(k+1,k) - L(k)H(k)]^T \lambda(k+1) + H^T(k)\nu(k); \tag{4.22b}$$

$$\ddot{\theta}(k) = \nu(k) - L^T(k)\lambda(k+1);$$

$$\varsigma(k) = \phi^T(k+1,k)\varsigma(k+1) + a(k);$$

$$\alpha(k) = \lambda(k) + \varsigma(k);$$

end loop;

$$\alpha(0) = \phi^T(1,0)\alpha(1) + a(0). \tag{4.23}$$

Algorithms $FD1$ and $FD2$ are both "4-Sweep Algorithms", 2 sweeps being required to compute the biases, followed by 2 sweeps to complete the computation of joint rates or spatial accelerations. The next two algorithms are 3-Sweep Algorithms.

ALGORITHM FD3.

$$\varsigma = \phi^T a; \tag{5.8}$$

$$\epsilon = T - H\psi[SLT + M\varsigma + b + Bf(0)]; \tag{5.9}$$

$$\ddot{\theta} = (I - H\Psi L)^T D^{-1}\epsilon; \tag{5.10}$$

Proof: From (3.19)

$$H\phi M\phi^T H^T\ddot{\theta} = T - H\phi[M\varsigma + b + Bf(0)].$$

Note that, see (5.3),

$$(I - H\Psi L)T = (I - H\psi SL)T = T - H\psi SLT.$$

Also note (5.4). With Theorem 4.1 the above yields

$$\ddot{\theta} = (I - H\Psi L)^{-1}D^{-1}[T - H\psi(SLT + M\varsigma + b + Bf(0)].$$

Algorithm $FD3$ requires anticausal-causal-anticausal sweeps to obtain $\ddot{\theta}$ as indicated respectively by (5.8), (5.9) and (5.10). In $FD3$, V and a can be computed during the anticausal sweep (5.8), and b, P, D, and L during the causal sweep (5.9). In the same way that $FD2$ was derived from $FD1$, algorithm $FD3$ can be modified to compute link and tip spatial accelerations although this is not done here. When obtaining the recursive forms of algorithms expressed operationally, such as the algorithms (5.8)- (5.10), the identities $A.1$ given in Appendix A are useful.

Note that Eq. (5.8) reflects a need to have a preliminary step to compute the bias accelerations a. The last algorithm given in this paper removes this requirement, although the need for a preliminary anticausal sweep for the purposes of computing V remains. We first define

$$\tilde{\psi} = \psi - I \tag{5.11}$$

and

$$G = PH^T D^{-1}. \tag{5.12}$$

We have

ALGORITHM FD4.

$$V = B^T \phi^T H^T \dot{\theta}; \tag{5.13}$$

$$\nu = D^{-1}(I - H\Psi L)T - D^{-1}H\widetilde{\psi}Pa - D^{-1}H\psi[b + Bf(0)]; \tag{5.14}$$

$$\ddot{\theta} = (I - H\Psi L)^T \nu - L^T \Psi^T (I - GH)^T a - G^T a; \tag{5.15}$$

<u>*Proof*</u>: See Appendix A.

Writing (5.14) and (5.15) as

$$z = \Psi LT + \widetilde{\psi}Pa + \psi[b + Bf(0)];$$

$$\epsilon = T - Hz;$$

$$\nu = D^{-1}\epsilon;$$

$$\lambda = \psi^T H^T \nu + \psi^T (I - GH)^T a;$$

$$\ddot{\theta} = \nu - L^T S^T \lambda - G^T a;$$

shows that (5.14) and (5.15) can be implemented as

$$z(0) = f(0); \quad P(0) = 0; \quad T(0) = 0;$$

for $k = 1, \ldots, N$ loop

$$z(k) = \psi(k, k-1)z(k-1) + L(k-1)T(k-1) + \psi(k, k-1)P(k-1)a(k-1) + b(k); \tag{5.16}$$

$$\epsilon(k) = T(k) - H(k)z(k);$$

$$\nu(k) = D^{-1}(k)e(k);$$

end loop;

for $k = N, \ldots, 1$ loop

$$\lambda(k) = \psi^T(k+1, k)\lambda(k+1) + H^T(k)\nu(k) + [I - G(k)H(k)]^T a(k); \tag{5.17}$$

$$\ddot{\theta}(k) = \nu(k) - L^T(k)\lambda(k+1) - G^T(k)a(k);$$

end loop;

The 3-Sweep algorithm (5.16)-(5.17) can be found in [6], where, viewing the RSD equations as describing a two-point boundary-value problem, it was derived by applying what [6] refers to as the "sweep" method to solve for $\ddot{\theta}(k)$.

It is possible to apply the tools of this paper to obtain a "2-Sweep" forward dynamics algorithm, avoiding the sweep contained in (5.8), although at the expense of much greater algorithmic complexity. The development is somewhat lengthy, however, and will not be given here.

6. CONCLUSION AND DISCUSSION

This paper has shown the power of a linear operator approach to solve the manipulator forward dynamics problem. This approach is based on techniques for operator factorization and inversion that were originally developed in the context of filtering and estimation theory and that have recently been extended to the analysis of multibody mechanisms [6,7]. Other results of [6,7] not discussed here include the derivation of efficient iterative methods for obtaining the entire robot mass matrix and its inverse, as well as the derivation of exact symbolic expressions for these quantities.

As a demonstration of the capabilities of this approach, several $O(N)$ iterative algorithms for computing the forward dynamics of an $N-$link rigid manipulator were developed. The first algorithm derived in this paper requires that the operator form of the robot dynamics be first placed in a bias-free form (3.21). Alternative algorithms which relax this requirement were then developed by using various operator identities. The ability to obtain alternative forward dynamics algorithms by the use of operator identities potentially allows for algorithmic modifications leading to greater computational efficiencies.

In the companion paper [15], efficient iterative algorithms for solving the forward dynamics of a closed-chain system (comprised of several arms grasping a commonly held object) are given which further show the power of the approach presented here.

7. ACKNOWLEDGMENT

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

8. REFERENCES

1. M. W. Walker and D. E. Orin, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *J. Dyn. Sys. Meas. and Control*, Vol. 104, pp. 205-211, 1982.

2. R. Featherstone, "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *Int. J.*

Robotics Research, Vol. 2, pp. 13-30, 1983.

3. R. Featherstone, *Robot Dynamics Algorithms*, Ph.D. Thesis, Univ. of Edinburgh, 1984.

4. J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "On-Line Computational Scheme for Mechanical Manipulators," *J. Dyn. Sys. Meas. and Control*, Vol. 102, pp. 69-76, 1980.

5. J. J. Craig, *Introduction to Robotics*, Addison-Wesley, Pub. Co., Reading, MA, 1986.

6. G. Rodriguez, *Kalman Filtering, Smoothing, and Recursive Robot Arm Forward and Inverse Dynamics*, JPL Publication 86-48, 1986. Also in *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 6, Dec. 1987.

7. G. Rodriguez, "Spatially Random Models, Estimation Theory, and Robot Arm Dynamics," *Proceedings of Workshop on Space Telerobotics*, JPL Publication 87-13, 1987.

8. A. E. Baggeroer, "A State-Variable Technique for the Solution of Fredholm Integral Equations," *IEEE Trans. Inf. Theory*, Vol. IT-15, pp 557-570, 1969.

9. A. Schumitzky, "On the Equivalence Between Riccati Equations and Fredholm Resolvents," *J. Computer Sys. Sci.*, Vol. 2, pp. 76-87, 1968.

10. T. Kailath, "Fredholm Resolvents, Wiener-Hopf Equations, and Riccati Differential Equations," *IEEE Trans. Inf. Theory*, Vol. IT-15, pp. 665-672, 1969.

11. T. Kailath, "The Innovations Approach to Detection and Estimation Theory," *Proc. IEEE*, Vol. 58, pp. 680-695, 1970.

12. T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Inf. Theory*, Vol. IT-20, pp. 147-181, 1974.

13. M. Gevers and T. Kailath, "An Innovations Approach to Least-Squares Estimation - Part VI: Discrete-Time Innovations Representations and Recursive Estimation," *IEEE Trans. Aut. Control*, Vol. AC-18, pp. 588-600, 1973.

14. B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall Inc., Englewood Cliffs, NJ., 1979.

15. G. Rodriguez and K. Kreutz, "Closed-Chain Dynamics for Multiple Robot Arms Moving a Common Object," *JPL Publication 88-11, Part II*, March 1988.

16. D. B. Silver, "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", *Int. J. of Robotics Research*, Vol. 1, 1982.

17. G. Rodriguez, "Recursive Forward Dynamics for Multiple Robot Arms Moving a Common Task Object," *JPL Publication 88-6*, Jan. 1988.

18. O. Khatib, "The Operational Space Formulation in the Analysis, Design, and Control of Manipulators," 3^{rd} *Int. Symp. Robotics Research*, Paris, 1985, pp. 103-110.

19. J. T. Wen and D. S. Bayard, "A New Class of Control Laws for Robotic Manipulators: Part I. Nonadaptive Case" *Int. J. Control*, 1988, in press.

20. W. L. Brogan, *Modern Control Theory*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1985.

21. P. C. Hughes, *Spacecraft Attitude Dynamics,*, Wiley, New York, NY, 1986.

22. J. T. Wen and K. Kreutz, "Globally Stable Tracking Control Laws for the Attitude Maneuver Problem," *1988 Automatic Control Conference*, Atlanta, GA, 1988.

23. J. Wittenburg, *Dynamics of Systems of Rigid Bodies*, B. G. Teubner, Stuttgart, 1977.

APPENDIX A: OPERATOR IDENTITIES

In this appendix, Facts 4.1, 4.2 and Algorithm $FD4$ are derived following the development of various useful and important operator identities. The derivations given here are "mechanical" and do not depend upon the insights available from the theory given in [8–14]. These insights greatly facilitate the understanding of the algorithms developed in this paper, and the results proven in this appendix can also be obtained by use of the methods of [13] and Chapter 9 of [14]. See also [6,7].

Recall that ϕ is given by (3.4), Φ by (4.9), ψ by (5.1), Ψ by (4.10), $\tilde{\psi}$, by (5.11), and S by (5.2). Also recall that P, D, and L are given by (4.4)–(4.8) and G by (5.12), giving

$$D = HPH^T \tag{a.1}$$

$$G = PH^T D^{-1} \tag{a.2}$$

We also define

$$\tilde{\phi} = \phi - I \tag{a.3}$$

and

$$\Delta_\phi = diag[\phi(2,1), \ldots, \phi(N+1, N)] \tag{a.4}$$

$$\Delta_\psi = diag[\psi(2,1), \ldots, \psi(N+1, N)] \tag{a.5}$$

Our first block of identities is given by

IDENTITIES A.1.

$$\Phi = \phi S \tag{a.6}$$

$$\Psi = \psi S \tag{a.7}$$

$$\Psi \Delta_\psi = \tilde{\psi} \tag{a.8}$$

$$\Phi \Delta_\phi = \tilde{\phi} \tag{a.9}$$

$$\psi^{-1} = I - S\Delta_\psi \tag{a.10}$$

$$\phi^{-1} = I - S\Delta_\phi \tag{a.11}$$

$$L = \Delta_\phi G \tag{a.12}$$

$$\tilde{\phi} G = \Phi L \tag{a.13}$$

Proof: Equations (a.6)-(a.9) and (a.12) are straightforward. Equations (a.10) and (a.11) follow from (a.6)–(a.9) since $\tilde{\psi} = \psi - I$ and $\tilde{\phi} = \phi - I$. Equation (a.13) follows from (a.12) and (a.9).

A second block of identities is given by

IDENTITIES A.2.

$$\phi = \psi(I + SLH\phi) \tag{a.14}$$

$$\psi = (I - \Psi LH)\phi \tag{a.15}$$

$$\Phi = \Psi(I + LH\Phi) \tag{a.16}$$

$$\Psi = (I - \Psi LH)\Phi \tag{a.17}$$

Proof: Equations (a.16) and (a.17) follow from (a.14) and (a.15) from postmultiplying by S and using (a.6) and (a.7). Equations (a.14) and (a.15) are both equivalent to

$$\phi - \psi = \Psi LH\phi = \psi SLH\phi \tag{a.18}$$

To prove (a.18) consider $T(k,m)$ defined by

$$T(k,m) = \sum_{i=m}^{k-1} [\psi(k,i+1)\phi(i+1,m) - \psi(k,i)\phi(i,m)] \quad \text{for} \quad k > m \tag{a.19}$$

with $T(k,m) = 0$ for $k \leq m$.

We have that

$$T(k,m) = \psi(k,k)\phi(k,m) - \psi(k,m)\phi(m,m) = \phi(k,m) - \psi(k,m) \tag{a.20}$$

On the other hand

$$T(k,m) = \sum_{i=m}^{k-1} [\psi(k,i+1)\phi(i+1,m) - \psi(k,i+1)\psi(i+1,i)\phi(i,m)]$$

$$= \sum_{i=m}^{k-1} \psi(k,i+1)[\phi(i+1,m) - \psi(i+1,i)\phi(i,m)]$$

$$= \sum_{i=m}^{k-1} \psi(k,i+1)[\phi(i+1,m) - [\phi(i+1,i) - L(i)H(i)]\phi(i,m)]$$

or

$$T(k,m) = \sum_{i=m}^{k-1} \psi(k,i+1)L(i)H(i)\phi(i,m) \tag{a.21}$$

Equations (a.20) and (a.21) yield

$$\phi(k,m) - \psi(k,m) = \sum_{i=m}^{k-1} \psi(k,i+1)L(i)H(i)\phi(i,m)$$

for $k > m$, thereby proving (a.18).

A third block of useful identities is given by

IDENTITIES A.3.

$$\phi = (I + \Phi LH)\psi \tag{a.22}$$

$$\psi = \phi(I - SLH\psi) \tag{a.23}$$

$$\Phi = (I + \Phi LH)\Psi \tag{a.24}$$

$$\Psi = \Phi(I - LH\Psi) \tag{a.25}$$

$$\Phi LH\psi = \Psi LH\phi \tag{a.26}$$

$$\Phi LH\Psi = \Psi LH\Phi \tag{a.27}$$

Proof: Equation (a.24) and (a.25) follow from (a.22) and (a.23) from post-multiplying by S and using (a.6) and (a.7). Equation (a.26) follows from (a.22) and (a.14). Equation (a.27) follows from (a.26) by postmultiplying by S. Equations (a.22) and (a.23) are both equivalent to

$$\phi - \psi = \Phi LH\psi = \phi SLH\psi \tag{a.28}$$

To prove (a.28) consider $T'(k,m)$ defined by

$$T'(k,m) = \sum_{i=m}^{k-1} [\phi(k,i+1)\psi(i+1,m) - \phi(k,i)\psi(i,m)] \quad \text{for} \quad k > m \tag{a.29}$$

with $T'(k,m) = 0$ for $k \leq m$.

We have

$$T'(k,m) = \phi(k,k)\psi(k,m) - \phi(k,m)\psi(m,m) = \psi(k,m) - \phi(k,m) \tag{a.30}$$

On the other hand,

$$\begin{aligned} T'(k,m) &= \sum_{i=m}^{k-1} [\phi(k,i+1)\psi(i+1,m) - \phi(k,i+1)\phi(i+1,i)\psi(i,m)] \\ &= \sum_{i=m}^{k-1} \phi(k,i+1)[\psi(i+1,m) - \phi(i+1,i)\psi(i,m)] \\ &= \sum_{i=m}^{k-1} \phi(k,i+1)[\psi(i+1,i) - \phi(i+1,i)]\psi(i,m) \end{aligned}$$

or

$$T'(k,m) = -\sum_{i=m}^{k-1} \phi(k,i+1)L(i)H(i)\psi(i,m). \tag{a.31}$$

Equations (a.30) and (a.31) yield

$$\phi(k,m) - \psi(k,m) = \sum_{i=m}^{k-1} \phi(k,i+1)L(i)H(i)\psi(i,m)$$

for $k > m$, thereby proving (a.28).

Yet another block of useful identities is given by

IDENTITIES A.4.

$$H\phi = (I + H\Phi L)H\psi \tag{a.32}$$

$$H\Phi = (I + H\Phi L)H\Psi \tag{a.33}$$

$$H\Psi = H\Phi(I - LH\Psi) \tag{a.34}$$

$$H\psi = (I - H\Psi L)H\phi \tag{a.35}$$

$$H\Psi = (I - H\Psi L)H\Phi \tag{a.36}$$

$$H\Phi = H\Psi(I + LH\Phi) \tag{a.37}$$

$$H\Phi LH\Psi = H\Psi LH\Phi \tag{a.38}$$

$$\Phi L = \Psi L(I + H\Phi L) \tag{a.39}$$

$$\Psi L = (I - \Psi LH)\Phi L \tag{a.40}$$

$$\Psi L = \Phi L(I - H\Psi L) \tag{a.41}$$

$$\Phi L = (I + \Phi LH)\Psi L \tag{a.42}$$

$$\Psi LH\Phi L = \Phi LH\Psi L \tag{a.43}$$

Proof: Equations (a.32)-(a.38) follow from (a.22), (a.24), (a.25), (a.15), (a.16), (a.17) and (a.27) by premultiplication by H. Equation (a.39)-(a.42) follow from (a.16), (a.17), (a.25) and (a.24) by postmultiplication by L. Equation (a.43) follows from (a.40) and (a.41).

LEMMA A.1. *For* $\dot{r} = diag[r(1), \ldots, r(N)]$ *where*

$$r(0) = 0$$

$$r(k+1) = \psi(k+1,k)r(k)\phi^T(k+1,k) + M(k+1) \tag{a.44}$$

we *have*

$$\psi M \phi^T = r + \tilde{\psi} r + r \tilde{\phi}^T. \tag{a.45}$$

Proof: It is an exercise in algebra to show that for $R = \psi M \phi^T$, $R(k,k) = r(k)$ where

$$r(k) = \sum_{j=1}^{k} \phi(k,j) M(j) \psi^T(k,j)$$

obeys (a.44) and

$$R(k,l) = \psi(k,l) r(l) \quad \text{for} \quad k \geq l \quad \text{and} \quad R(k,l) = r(k) \phi^T(l,k) \quad \text{for} \quad k \leq l.$$

To show this, exploit the fact that M is block diagonal and ψ and ϕ are both causal (lower block triangular) with elements which obey the semi-group properties $\psi(j,k) = \psi(j,l)\psi(l,k)$, $\phi(j,k) = \phi(j,l)\phi(l,k)$, and $\phi(j,j) = \psi(j,j) = I$. Thus $R = r + \tilde{\psi} r + r \tilde{\phi}^T$.

COROLLARY A.1. *: LEMMA A.1 holds for $\psi = \phi$.*

Proof: The only restrictions on ψ and ϕ are that they be causal with elements which obey the semi-group properties.

Since $\tilde{\psi}$ is strictly causal (strictly lower block triangular), $\tilde{\phi}^T$ is strictly anticausal (strictly upper block triangular), and r is memoryless (block diagonal), LEMMA A.1 is a statement about how the operator $\psi M \phi^T$ can be decomposed into the sum of a memoryless term, a strictly causal term, and a strictly anti-causal term.

IDENTITIES A.5.

$$\psi M \phi^T = P + \tilde{\psi} P + P \tilde{\phi}^T \tag{a.46}$$

$$\psi M \psi^T = P + \tilde{\psi} P + P \tilde{\psi}^T. \tag{a.47}$$

Proof: *$P = diag[P(1), \ldots, P(N)]$ with $P(k)$ given by (See Equations (4.4)-(4.6))*

$$P(0) = 0;$$

$$\begin{aligned} P(k+1) &= \phi(k+1,k)[P(k) - P(k)H^T(k)D^{-1}(k)H(k)P(k)]\phi^T(k+1,k) + M(k+1) \\ &= \phi(k+1,k)[I - G(k)H(k)]P(k)\phi^T(k+1,k) + M(k+1) \\ &= [\phi(k+1,k) - L(k)H(k)]P(k)\phi^T(k+1,k) + M(k+1) \end{aligned}$$

$$= \psi(k+1,k)P(k)\phi^T(k+1,k) + M(k+1).$$

Equation (a.46) now follows from LEMMA A.1. Now note that

$$[I - G(k)H(k)]P(k)H^T(k)G^T(k) = [P(k)H^T(k) - P(k)H^T(k)D^{-1}(k)H(k)P(k)H^T(k)]G^T(k)$$

$$= [P(k)H^T(k) - P(k)H^T(k)]G^T(k) = 0$$

Thus

$$P(k+1) = \phi(k+1,k)[I - G(k)H(k)]P(k)[I - G(k)H(k)]^T\phi^T(k+1,k) + M(k+1)$$

$$= \psi(k+1,k)P(k)\psi^T(k+1,k) + M(k+1)$$

and (a.47) follows from Corollary A1.

IDENTITY A.6.

$$\phi M\phi^T = P + \tilde{\phi}P + P\tilde{\phi}^T + \tilde{\phi}PH^TD^{-1}HP\tilde{\phi}^T. \qquad (a.48)$$

<u>*Proof*</u>: $P = diag[P(1), \ldots, P(N)]$ with $P(k)$ given by

$$P(0) = 0;$$

$$P(k+1) = \phi(k+1,k)[P(k) - P(k)H^T(k)D^{-1}(k)H(k)P(k)]\phi^T(k+1,k) + M(k+1)$$

$$= \phi(k+1,k)P(k)\phi^T(k+1,k) + \pi(k+1); \qquad (a.49)$$

where

$$\pi(k+1) = M(k+1) - \Theta(k+1);$$

$$\Theta(k+1) = \phi(k+1,k)P(k)H^T(k)D^{-1}(k)H(k)P(k)\phi^T(k+1,k).$$

Define $\pi = diag[\pi(1), \ldots, \pi(N)]$ and $\Theta = diag[\Theta(1), \ldots, \Theta(N)]$. Then $\pi = M - \Theta$. From (a.49) and Corollary (A.1),

$$\phi\pi\phi^T = P + \tilde{\phi}P + P\tilde{\phi}^T$$

or

$$\phi M\phi^T = P + \tilde{\phi}P + P\tilde{\phi}^T + \phi\Theta\phi^T.$$

Algebraic manipulations show that

$$\phi\Theta\phi^T = \tilde{\phi}PH^TD^{-1}HP\tilde{\phi}^T$$

proving our result.

IDENTITY A.7.

$$(I - H\Psi L)^T D^{-1} H\psi M\phi^T$$

$$= (I - H\Psi L)^T D^{-1} H\tilde{\psi}P + L^T\Psi^T(I - GH)^T + G^T. \qquad (a.50)$$

Proof:

$$(I - H\Psi L)^T D^{-1} H(\psi M\phi^T)$$

$$= (I - H\Psi L)^T D^{-1} H(P + \tilde{\psi}P + P\phi^T) \quad \text{from (a.45)}$$

$$= (I - H\Psi L)^T D^{-1} H\tilde{\psi}P + (I - H\Psi L)^T D^{-1} HP + (I - H\Psi L)^T D^{-1} HP\tilde{\phi}^T.$$

Comparing the last expression to (a.50), we see that (a.50) is true provided that

$$(I - H\Psi L)^T D^{-1} HP + (I - H\Psi L)^T D^{-1} HP\tilde{\phi}^T$$

$$= L^T\Psi^T(I - GH)^T + G^T \qquad (a.51)$$

We have that

$$(I - H\Psi L)^T D^{-1} HP + (I - H\Psi L)^T D^{-1} HP\tilde{\phi}^T$$

$$= (I - H\Psi L)^T G^T + (I - H\Psi L)^T G^T\tilde{\phi}^T$$

$$= (I - H\Psi L)^T G^T + (I - H\Psi L)^T L^T\Phi^T \quad \text{from (a.13)}$$

$$= (I - H\Psi L)^T G^T + [\Phi(I - LH\Psi)L]^T$$

$$= (I - H\Psi L)^T G^T + L^T\Psi^T \quad \text{from (a.25)}$$

$$= L^T\Psi^T(I - GH)^T + G^T.$$

This proves (a.51) and hence (a.50).

PROOF OF FACT 4.1.

With (a.13), it is enough to show that

$$H\phi M\phi^T H^T = (I + H\tilde{\phi}G)D(I + H\tilde{\phi}G)^T \quad \text{where} \quad G = PH^T D^{-1}.$$

We have that

$$(I + H\tilde{\phi}G)D(I + H\tilde{\phi}G)^T$$

$$= HPH^T + H\tilde{\phi}PH^T + HP\tilde{\phi}^T H^T + H\tilde{\phi}PH^T D^{-1}HP\tilde{\phi}^T H^T$$

$$= H(P + \tilde{\phi}P + P\tilde{\phi}^T + \tilde{\phi}PH^T D^{-1}HP\tilde{\phi}^T)H^T$$

$$= H\phi M\phi^T H^T$$

where the last step follows from (a.48).

PROOF OF FACT 4.2.

$$(I + H\Phi L)(I - H\Psi L) = I + H(\Phi - \Psi - \Psi LH\Phi)L = I \quad \text{from (a.16)}.$$

PROOF OF ALGORITHM FD4.

From (3.19),

$$H\phi M\phi^T H^T \ddot{\theta} = T - H\phi[M\phi^T a + b + Bf(0)]$$

Theorem 4.1 and (a.35) are used to obtain

$$\ddot{\theta} = (I - H\Psi L)^T D^{-1}(I - H\Psi L)T - (I - H\Psi L)^T D^{-1} H\psi[b + Bf(0)]$$

$$-(I - H\Psi L)^T D^{-1} H\psi M\phi^T a.$$

Note that ALGORITHM FD4 can be written as

$$\ddot{\theta} = (I - H\Psi L)^T D^{-1}(I - H\Psi L)T - (I - H\Psi L)^T D^{-1} H\psi[b + Bf(0)]$$

$$-[(I - H\Psi L)^T D^{-1} H\tilde{\psi}P + L^T\Psi^T(I - GH)^T + G^T]a.$$

The proof follows from (a.50).

APPENDIX B: EXTENSION TO A MOBILE BASE

The extension here is to the case where the base is completely free to move throughout Euclidean 3-space. In this case, the natural configuration space for describing the translation and orientation of the base is $R^3 \times SO(3)$, where $SO(3)$ is the rotation group [21,22]. This results in no loss of generality as the procedure shown here for incorporating configuration, velocity, and acceleration parameters appropriate for describing base behavior into the manipulator equations has an obvious analogue for a choice of configuration space other than $R^3 \times SO(3)$. In any event, every other possible configuration space for the base must be a submanifold of $R^3 \times SO(3)$ (e.g., R^2 for a nonrotating base constrained to the plane) and an appropriate restriction of base configurations, velocities, accelerations and forces in the general solution given here will result in the solution for any possible base configuration space. A restriction of $R^3 \times SO(3)$ to $SO(3)$ is particularly easy and interesting as it is equivalent to making the base a 3 degrees-of-freedom (dof) spherical (ball-in-socket) joint, enabling the arm to have the anthropomorphic quality of a 3 dof spherical "shoulder-like" base. In fact, the procedure shown here for admitting $SO(3)$ as a configuration space for the base can be used to model any arm joint as a 3-dof spherical joint.

The first step is to now rename link N to be the base, whereas link $N+1$ will now be called the "station". In terms of the notation used in the paper, the arm would be an N-dof mechanism, only now there must be added an extra 5 degrees of freedom, since $R^3 \times SO(3)$ is a 6-dimensional manifold. For now, however, continue to think of the base link N as having a 1 dof joint.

The configuration parameter for joint N has been taken to be $\theta(N) \in R^1$ with a velocity $\dot{\theta}(N) = d\theta/dt$. There is a need to extend $\theta(N)$ to be a vector parameterization of base location and orientation (i.e., of $R^3 \times SO(3)$), and to allow $\theta(N)$ to have an associated velocity which is not $\dot{\theta}(N)$. For this reason, define $W(k) = \dot{\theta}(k)$. With this change, the manipulator equations (3.20) become

$$T = \mathcal{M}\dot{W} + \mathcal{C} + J^T f(0). \qquad (b.1)$$

Equivalently, the RSD equations (2.24)-(2.30) hold with $\ddot{\theta}(k)$ replaced by $\dot{W}(k)$. The dynamical behavior of link N is given by the modified RSD equations for $k = N$. The velocity of the base link N is found from (2.25) to be

$$V(N) = \phi^T(N+1, N)V(N+1) + H^T(N)W(N).$$

Take $\mathcal{O}(N)$ to be any point on the base link N and take the station velocity $V(N+1) = 0$. This gives

$$V(N) = H^T(N)W(N).$$

Now the appropriate velocity for describing the base link N is precisely $W(N) = V(N)$, since $V(N) = col[\omega(N), v(N)]$, where $v(N) = \dot{\ell}(N+1, N)$, and $\omega(N)$ gives the angular rate of change of the attitude of the base with respect to the station [21,23]. Thus

$$V(N) = H^T(N)W(N)$$

$$H^T(N) = I \in R^{6\times 6}. \tag{b.2}$$

Appropriate configuration parameters for the base link N must be chosen to describe the base in $R^3 \times SO(3)$. Take $\theta(N) \in R^7$, where $\theta(N) = col[q, \ell(N+1, N)]$ with $\ell(N+1, N) \in R^3$ being the base location with respect to the station. The Euler parameters (unit quaternions) $q \in R^4$ give the attitude of the base with respect to the station [21,23]. The Euler parameters are a globally nonsingular representation of $SO(3)$, unlike the Euler angles which can become singular. The relationship between $\dot{q}$ and $\omega(N)$ is given by

$$\dot{q} = \overset{4\times 3}{\Lambda}(q)\omega(N)$$

$$\omega = \pi(q)\dot{q} \tag{b.3}$$

where $\Omega^{-1}(q) = \pi(q)$, and $\Lambda(q)$ are always nonsingular and can be found in [21-23]. This gives

$$\dot{\theta}(N) = \Omega[\theta(N)]W(N) \qquad W(N) = \begin{bmatrix} \omega(N) \\ \dot{\ell}(N+1,N) \end{bmatrix}$$

$$\Omega[\theta(N)] = \begin{pmatrix} \Lambda(q) & 0 \\ 0 & I \end{pmatrix}. \tag{b.4}$$

With θ having elements $\theta(1), \ldots, \theta(N-1) \in R^1$, and $\theta(N) \in R^7$, then $\theta \in R^{N+6}$, where N is the number of links and $N+5$ is the total number of arm degrees of freedom. Now define $\Omega[\theta(k)] = 1$ for $k = 1, \ldots, N-1$, $\Omega[\theta(N)]$ by (b.4), and $\Omega = diag[\Omega(k)]$. Then,

$$\dot{\theta} = \Omega W. \tag{b.5}$$

Differentiation of (b.2) gives

$$\alpha(N) = H^T(N)\dot{W}(N) \equiv \dot{W}(N), = \begin{bmatrix} \ddot{q} \\ \dot{\omega} \end{bmatrix} \tag{b.6}$$

with the station velocity $\alpha(N+1) = 0$, since gravity is being ignored. Eq. (b.6) shows that now

$$a(N) = 0. \tag{b.7}$$

Eq. (2.29) for $k = N$ is unchanged, but (2.30) is now

$$T(\underset{N}{k}) = H(\underset{N}{k})f(\underset{N}{k}) = f(\underset{N}{k}) \in R^6. \tag{b.8}$$

The extension is now complete and (b.1)-(b.8) succinctly result in

$$T = \mathcal{M}\dot{W} + \mathcal{C} + J^T f(0)$$

$$\dot{\theta} = \Omega W. \tag{b.9}$$

The operator forms and interpretations of $\mathcal{M}$, $\mathcal{C}$, and J^T still hold and in particular

$$\dot{W} = \mathcal{M}^{-1}[T - \mathcal{C} - J^T f(0)] \tag{b.10}$$

can be obtained recursively by the operator factorization and inversion given in this paper. The only change is in the dimensions of the operators. All such changes are conformal and hence harmless.

Table 1.1 SYMBOL DEFINITIONS

$h(k)$ = unit vector along joint axis k

$\theta(k)$ = angle of link k with respect to link $k+1$ about joint axis k

$O(k)$ = fixed point on joint axis k, which can be viewed as the origin of a frame fixed in link k

$\ell(k, k-1)$ = vector from $O(k)$ to $O(k-1)$

$F(k)$ = constraint force on link k at point $O(k)$ of joint k

$N(k)$ = constraint moment on link k at joint k

$F^c(k)$ = net force on link k at link k mass center

$CM(k)$ = mass center of link k

$p(k)$ = vector from $O(k)$ to $CM(k)$

$v(k)$ = velocity of link k at point $O(k)$ of joint k

$\omega(k)$ = angular velocity of link k

$v^c(k)$ = velocity of link k at link k mass center

$m(k)$ = mass of link k

$I^c(k)$ = inertia tensor of link k at point $CM(k)$

$I(k)$ = inertia tensor of link k at point $O(k)$

$T(k)$ = actuated torque at joint k

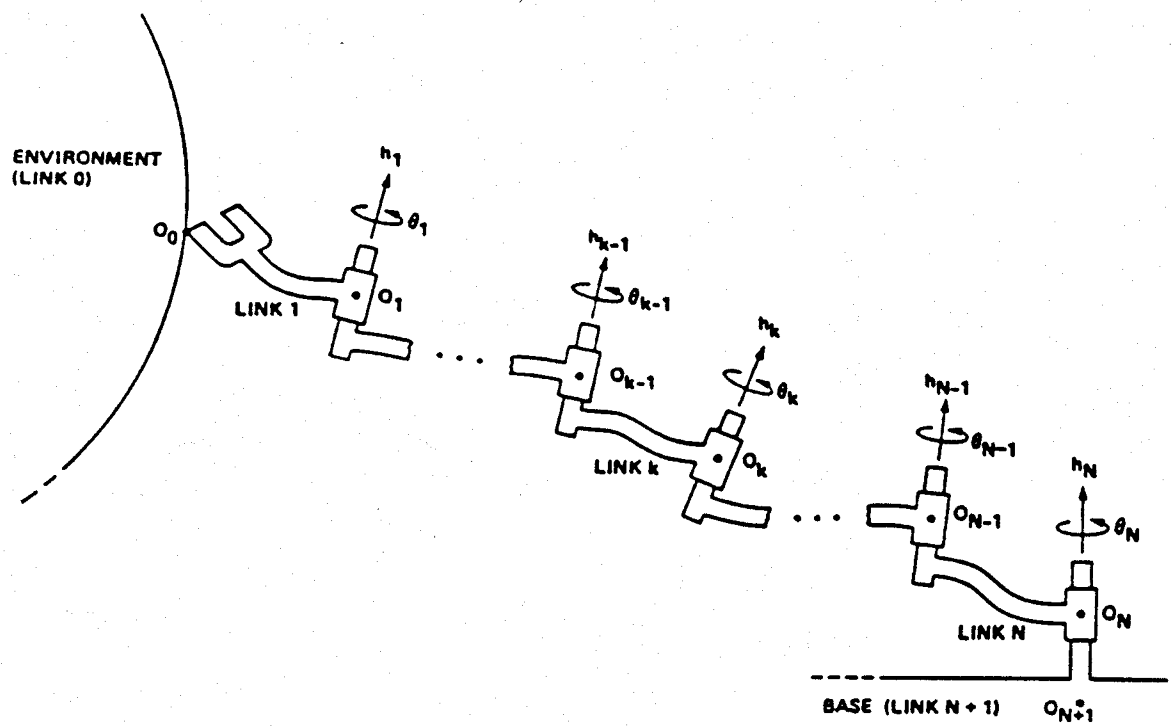

Fig. 1.1 N-Link Serial Manipulator

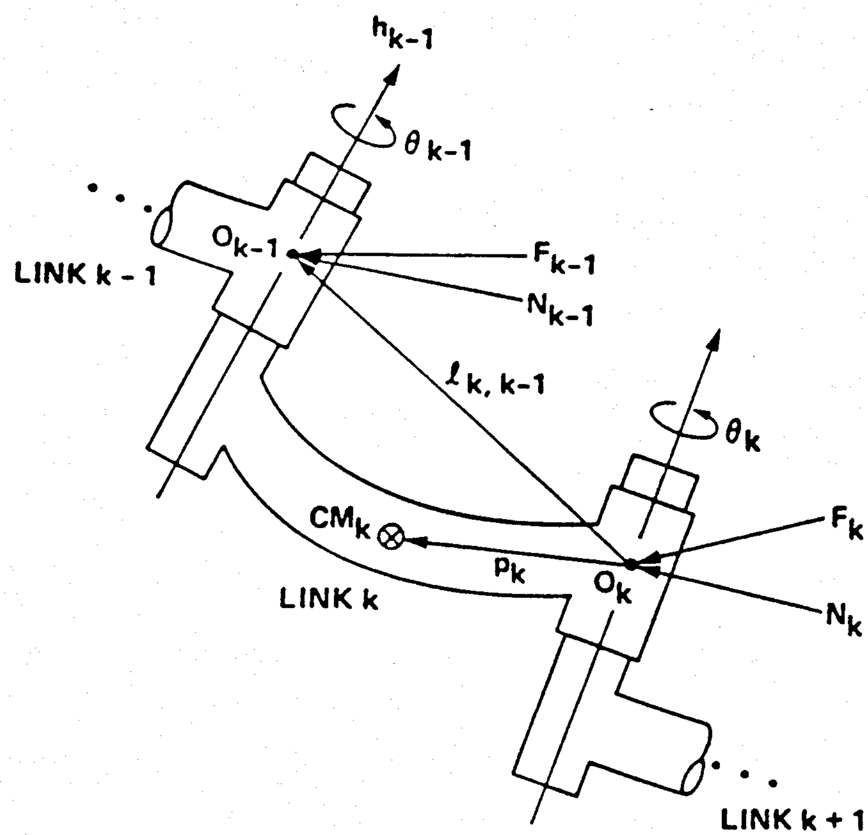

Fig. 1.2 Relationship of Defined Quantities to Link k