Journal of Guidance, Control and Dynamics, vol. 14, pp. 531-542, May-June 1991

Unified Formulation of Dynamics for Serial Rigid Multibody Systems

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

Abstract

There has been a growing interest in the development of new and efficient algorithms for multibody dynamics in recent years. Serial rigid multibody systems form the basic subcomponents of general multibody systems and a variety of algorithms to solve the serial chain forward dynamics problem have been proposed. In this paper, the economy of representation and analysis tools provided by the spatial operator algebra are used to clarify the inherent structure of these algorithms, to identify those that are similar, and to study the relationship among the ones that are distinct. For the purposes of this study, the algorithms are categorized into three classes: algorithms that require the explicit computation of the mass matrix, algorithms that are completely recursive in nature, and algorithms of intermediate complexity. In addition, alternative factorizations for the mass matrix and closed form expressions for its inverse are derived. These results provide a unifying perspective, within which these diverse dynamics algorithms arise naturally as a consequence of a progressive exploitation of the structure of the mass matrix.

Unified Formulation of Dynamics for Serial Rigid Multibody Systems

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

1 Introduction

The increase in size and complexity of spacecraft and robotic systems has led to a growing interest in the development of new and efficient algorithms for multibody dynamics. Serial rigid multibody systems form the basic subcomponents of general multibody systems, and a variety of algorithms to solve the serial chain forward dynamics problem have been proposed in recent years. In this paper, these algorithms are presented in one common setting in order to compare them and to analyze their interrelationships. A comprehensive analysis of the algorithms for serial chain dynamics will aid in the development of dynamics algorithms for more complex systems.

The serial rigid multibody system studied here is assumed to contain n links and possess $\mathcal{N}$ motion degrees of freedom (dof). The *inverse dynamics* problem requires the computation of the joint forces needed to obtain prescribed joint accelerations. There exist efficient $O(\mathcal{N})^\dagger$ Newton-Euler recursive algorithms to carry out the necessary computations. The *forward dynamics* is the converse problem and requires the computation of the joint accelerations generated by a given set of joint forces. It is the more difficult problem and is the primary focus of this paper. A variety of algorithms for its solution are studied here, and they are classified into the three categories of $O(\mathcal{N}^3)$, $O(\mathcal{N}^2)$, and $O(\mathcal{N})$ algorithms, based on their algorithmic complexity.

There are two stages in the derivation of computational dynamics algorithms. The first stage is to formulate solution algorithms. The focus of this paper is entirely on this first stage. Its goal is to clarify the inherent structure of the forward dynamics algorithms, to identify those that are similar, and to study the relationship among the ones that are distinct. This is considerably simplified using coordinate free *spatial notation* rather than minimal dimensional notation. In the second stage, these algorithm formulations are used to develop optimized computational algorithms for implementation. Here, unlike in the first stage, it is necessary to work with equations of minimal dimension.

Rather than reproducing the original rationale that led to the development of these forward dynamics algorithms, we use techniques from the *spatial operator algebra* framework for multibody dynamics (see Refs. 1 – 3) to derive the algorithms and study the relationships among them. The reasons for using the spatial operator algebra approach are as follows: (a) the use of spatial operators from the framework makes it possible to describe and analyze several algorithms concisely

*Member of Technical Staff, Machine Intelligence Applications Group, Robotics and Automation Section
†An $O(a^b)$ algorithm is one whose operations count is given by a polynomial in a of order at most b

within a single paper; (b) they reveal new relationships among the classes of the $O(\mathcal{N}^3)$, $O(\mathcal{N}^2)$ and $O(\mathcal{N})$ algorithms; (c) results such as the factorization and inversion of the mass matrix can be derived; (d) it provides a unifying perspective, within which, diverse dynamics algorithms arise naturally as a consequence of a progressive exploitation of the structure of the mass matrix; (e) due to the intimate relationship of the spatial operator algebra to the area of optimal filtering and estimation, techniques from this latter field can be applied to the domain of multibody dynamics.

In Section 2, the dynamical equations of motion of a serial rigid multibody system with general multiple dof joints are developed. Section 3 is a discussion of the $O(\mathcal{N}^3)$ algorithms in Ref. 4 for the solution of the forward dynamics problem. The $O(\mathcal{N}^2)$ methods in Refs. 1, 4 – 6 are studied in Section 4. In Section 5, we derive the $O(\mathcal{N})$ Articulated Body algorithm and use it to compare the $O(\mathcal{N})$ algorithms described in Refs. 7 – 13. Section 6 contains a derivation of operator factorizations and inversion of the mass matrix using spatial algebra techniques. Section 7 summarizes the conclusions of this paper.

Notation

In the stacked notation used in this paper, indices are used to identify quantities pertinent to a specific link. Thus for instance, V denotes the vector of the spatial velocities of all the links in the serial chain, and $V(k)$ denotes the spatial velocity vector of the k^{th} link.

Serial Chain:

n	number of links in the serial chain
$r_p(k)$	number of positional dofs of the k^{th} joint
$r_v(k)$	number of motion dofs of the k^{th} joint
$\mathcal{N}$	total number of motion dofs for the serial chain
$\mathcal{M}$	$\in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$, mass matrix for the serial chain

Link/Joint properties:

$\mathcal{O}_k^+$	reference location of the k^{th} joint on the $(k+1)^{th}$ link
$\mathcal{O}_k$	reference location of the k^{th} joint on the k^{th} link
$l(k, k-1)$	vector from $\mathcal{O}_k$ to $\mathcal{O}_{k-1}$
$\theta(k)$	$\in \mathcal{R}^{r_p(k)}$, configuration variables for k^{th} joint
$\beta(k)$	$\in \mathcal{R}^{r_v(k)}$, joint motion parameters for k^{th} joint
$H^*(k)$	$\in \mathcal{R}^{6\times r_v(k)}$, joint matrix for the k^{th} joint
$m(k)$	mass of the k^{th} link
$\mathcal{J}(k)$	$\in \mathcal{R}^{3\times3}$, moment of inertia of the k^{th} link about $\mathcal{O}_k$
$p(k)$	$\in \mathcal{R}^3$, vector from $\mathcal{O}_k$ to the center of mass of the k^{th} link
$M(k)$	$\in \mathcal{R}^{6\times6}$, spatial inertia of the k^{th} link about $\mathcal{O}_k$

Forces and velocities:

$v(k)$	$\in \mathcal{R}^3$, linear velocity of the k^{th} link at $\mathcal{O}_k$
$\omega(k)$	$\in \mathcal{R}^3$, angular velocity of the k^{th} link
$V(k)$	$\in \mathcal{R}^6$, spatial velocity of the k^{th} link at $\mathcal{O}_k$
$\alpha(k)$	$\in \mathcal{R}^6$, spatial acceleration of the k^{th} link at $\mathcal{O}_k$

$T(k) \quad \in \mathcal{R}^{r_v(k)}$, joint force at the k^{th} joint
$\bar{T}(k) \quad \in \mathcal{R}^{r_v(k)}$, part of the joint force $T(k)$ free of velocity dependent terms
$F(k) \quad \in \mathcal{R}^3$, linear force of interaction between the $(k+1)^{th}$ and k^{th} links at $\mathcal{O}_k$
$N(k) \quad \in \mathcal{R}^3$, moment of interaction between the $(k+1)^{th}$ and k^{th} links about $\mathcal{O}_k$
$f(k) \quad \in \mathcal{R}^6$, spatial force of interaction between the $(k+1)^{th}$ and k^{th} links at $\mathcal{O}_k$
$a(k) \quad \in \mathcal{R}^6$, Coriolis and centrifugal spatial acceleration at $\mathcal{O}_k$
$b(k) \quad \in \mathcal{R}^6$, gyroscopic spatial force at $\mathcal{O}_k$

Composite rigid body terms:

$\phi(k+1,k) \quad \in \mathcal{R}^{6\times 6}$, composite body transformation operator from $\mathcal{O}_k$ to $\mathcal{O}_{k+1}$
$\phi, \mathcal{E}_\phi \quad \in \mathcal{R}^{6n\times 6n}$, composite body transformation operators for the full serial chain
$R(k) \quad \in \mathcal{R}^{6\times 6}$, composite rigid body spatial inertia of links $k \ \cdots \ 1$ about $\mathcal{O}_k$

Articulated body terms:

$\psi(k+1,k) \quad \in \mathcal{R}^{6\times 6}$, articulated body transformation operator from $\mathcal{O}_k$ to $\mathcal{O}_{k+1}$
$\psi, \mathcal{E}_\psi \quad \in \mathcal{R}^{6n\times 6n}$, articulated body transformation operators for the full serial chain
$P(k) \quad \in \mathcal{R}^{6\times 6}$, articulated body spatial inertia of links $k \ \cdots \ 1$ about $\mathcal{O}_k$
$z(k) \quad \in \mathcal{R}^6$, articulated body correction force at $\mathcal{O}_k$
$\nu(k) \quad \in \mathcal{R}^6$, articulated body relative joint acceleration at the k^{th} joint
$\epsilon(k) \quad \in \mathcal{R}^6$, effective joint force generating joint accelerations $\nu(k)$ at the k^{th} joint
$\overline{\tau}(k), \tau(k) \quad \in \mathcal{R}^{6\times 6}$, articulated body projection operators for the k^{th} joint
$D(k) \quad \in \mathcal{R}^{r_v(k)\times r_v(k)}$, joint inertia at the k^{th} joint

2 Equations of Motion

We begin by developing the equations of motion of a single rigid body and in the process introduce the coordinate-free spatial notation used throughout this paper. We then derive the Newton-Euler equations of motion for a serial rigid multibody system and introduce spatial operators to recast them in a concise form.

Dynamics of a Single Rigid Body

Let O and C be two points on a rigid body and $l(OC)$ the vector from O to C. Let $v(C)$ and $\omega(C)$ be the linear and angular velocities of the body at C, and $F(C)$ and $N(C)$ the force and moment at and about C. Then the forces and velocities at the points C and O are related to each other as follows:

$$\omega(C) = \omega(O), \quad v(C) = v(O) + \omega(O) \times l(OC), \quad F(O) = F(C), \quad N(O) = N(C) + l(OC) \times F(C) \tag{1}$$

We obtain a more compact form for (1) by using *spatial notation.* The *spatial velocity, spatial acceleration* and *spatial force* at the point C are defined as

$$V(C) \triangleq \begin{pmatrix} \omega(C) \\ v(C) \end{pmatrix}, \quad \alpha(C) \triangleq \begin{pmatrix} \dot{\omega}(C) \\ \dot{v}(C) \end{pmatrix}, \quad f(C) \triangleq \begin{pmatrix} N(C) \\ F(C) \end{pmatrix} \tag{2}$$

The *composite (rigid) body transformation operator* $\phi(.)$ relates the spatial force/velocity quantities between two points on a rigid body, and is a function of the vector joining the points. For a given vector l, the operator $\phi(l)$ is defined as:

$$\phi(l) \stackrel{\triangle}{=} \begin{pmatrix} I & \tilde{l} \\ 0 & I \end{pmatrix} \tag{3}$$

where $\tilde{l}$ denotes the skew-symmetric cross-product matrix associated with the vector l, which acts on another vector to produce the cross-product of l with the vector. Using Eqs. (2) and (3), (1) can be reexpressed in the form:

$$\begin{aligned} V(C) &= \phi^*\Big(l(OC)\Big)V(O), \quad f(O) = \phi\Big(l(OC)\Big)f(C) \\ \alpha(C) &= \phi^*\Big(l(OC)\Big)\alpha(O) + a(O) \quad \text{where} \quad a(O) \stackrel{\triangle}{=} \dot{\phi}^*\Big(l(OC)\Big)V(O) = \begin{pmatrix} 0 \\ \tilde{\omega}(O)\tilde{\omega}(O)l(OC) \end{pmatrix} \end{aligned} \tag{4}$$

$a(O)$ is the velocity dependent Coriolis and centrifugal acceleration term at O, and a dot above a symbol denotes the time derivative in an inertial frame. Equation (4) is a succinct way of describing the equations of motion of rigid bodies and motivates the use of the spatial notation. It also illustrates the duality between the transformation properties of spatial forces and the spatial accelerations/velocities. Note also that

$$\phi^{-1}\Big(l(OC)\Big) = \phi\Big(l(CO)\Big) \quad \text{and} \quad \phi\Big(l(OQ)\Big) = \phi\Big(l(OC)\Big)\phi\Big(l(CQ)\Big) \tag{5}$$

where Q is a third point on the body. Thus the $\phi(.)$ operators satisfy the *semigroup property*.

Let us assume now that C is the center of mass of the rigid body and that the body has mass m and moment of inertia $\mathcal{J}(C)$ about C. The *spatial inertia* $M(C)$ and the *spatial momentum* $\Pi(C)$ of the body about C are defined as

$$M(C) \stackrel{\triangle}{=} \begin{pmatrix} \mathcal{J}(C) & 0 \\ 0 & mI \end{pmatrix} \quad \text{and} \quad \Pi(C) \stackrel{\triangle}{=} M(C)V(C) \tag{6}$$

About the center of mass C, the equations of motion for the rigid body are given by

$$f(C) = \dot{\Pi}(C) = M(C)\alpha(C) + \dot{M}(C)V(C) = M(C)\alpha(C) + b(C) \tag{7}$$

where

$$b(C) \stackrel{\triangle}{=} \dot{M}(C)V(C) = \begin{pmatrix} \tilde{\omega}(C)\mathcal{J}(C)\omega(C) \\ 0 \end{pmatrix} \tag{8}$$

$b(C)$ is the velocity dependent gyroscopic spatial force at C.

For studying the dynamics of multibody systems, it is useful to develop the equations of motion of a rigid body about points other than the center of mass. Using (4) and (7) we obtain the following equations of motion about O.

$$\begin{aligned} f(O) &= \phi\Big(l(OC)\Big)f(C) = \phi\Big(l(OC)\Big)[M(C)\alpha(C) + b(C)] \\ &= \phi\Big(l(OC)\Big)\Big[M(C)\Big\{\phi^*\Big(l(OC)\Big)\alpha(O) + a(O)\Big\} + b(C)\Big] = M(O)\alpha(O) + b(O) \end{aligned} \tag{9}$$

where $\mathcal{J}(O) \stackrel{\triangle}{=} \mathcal{J}(C) - m\tilde{l}(OC)\tilde{l}(OC)$,

$$M(O) \stackrel{\triangle}{=} \phi\Big(l(OC)\Big)M(C)\phi^*\Big(l(OC)\Big) = \begin{pmatrix} \mathcal{J}(O) & m\tilde{l}(OC) \\ -m\tilde{l}(OC) & mI \end{pmatrix} \tag{10}$$

and

$$\begin{aligned} b(O) &\stackrel{\triangle}{=} \phi\Big(l(OC)\Big)[M(C)a(O) + b(C)] = \phi\Big(l(OC)\Big)[M(C)\ \phi^*\Big(l(\dot{OC})\Big)\ V(O) + \dot{M}\ (C)V(C)] \\ &= \dot{M}\ (O)V(O) - \phi\Big(l(\dot{OC})\Big)\ M(O)V(O) = \begin{pmatrix} \tilde{\omega}(O)\mathcal{J}(O)\omega(O) \\ m\tilde{\omega}(O)\tilde{\omega}(O)l(OC) \end{pmatrix} \end{aligned} \tag{11}$$

We have used the fact that $\phi\Big(l(OC)\Big)\dot{\phi}\Big(l(CO)\Big) = \dot{\phi}\Big(l(CO)\Big) = -\dot{\phi}\Big(l(OC)\Big)$. $M(O)$ is the spatial inertia, $\mathcal{J}(O)$ the moment of inertia, and $b(O)$ the gyroscopic spatial force about O. Note that (10) is precisely the extension of the well known *parallel axis theorem* for moments of inertias to the case of spatial inertias.

Dynamics of a Serial Rigid Multibody System

In this section we derive the equations of motion for a serial multibody system composed of n rigid links connected together by multiple dof joints. The links are numbered 1 through n from tip to base. The term *outboard* (*inboard*) link is used to refer to a link on the path towards the tip (base) respectively. We allow for the use of quasicoordinates as well as joints with holonomic and nonholonomic constraints. For notational convenience, all the joint constraints are assumed to be homogeneous (i.e., catastatic). References 14, 15 contain a detailed discussion of the kinematical properties of general joints.

As shown in Fig. 8, the reference location of the k^{th} joint on the k^{th} and the $(k+1)^{th}$ links, are denoted $\mathcal{O}_k$ and $\mathcal{O}_k^+$ respectively. The k^{th} joint is assumed to possess $r_p(k)$ positional and $r_v(k)$ $(\leq r_p(k))$ motion dofs. The vector of configuration variables (of dimension at least $r_p(k)$) is denoted $\theta(k)$, and the $r_v(k)$ dimensional parameter vector for the joint velocity is denoted $\beta(k)$. The *kinematical equations* which relate $\dot{\theta}(k)$ to $\beta(k)$ depend on the specific nature of the joint. $H(k)$ is defined such that $H^*(k)$ is the $6 \times r_v(k)$ *joint matrix* for the k^{th} joint, and it maps $\beta(k)$ into the relative spatial velocity $\Delta_V(k)$ across the joint, i.e., $\Delta_V(k) = H^*(k)\beta(k)$. The complexity of the dynamics algorithms for the serial chain is determined by the number of overall motion dofs for the chain which is given by

$$\mathcal{N} \stackrel{\triangle}{=} \sum_{k=1}^{n} r_v(k) \tag{12}$$

The state of the multibody system is assumed known and is defined by the collection of vectors $[\theta(.), \beta(.)]$ for all the joints. To derive the equations of motion of the serial chain, we first focus on the k^{th} link. Each link being rigid, it suffices to develop the equations of motion at a single reference point on the link, which for the k^{th} link we take to be the k^{th} joint location $\mathcal{O}_k$. The spatial velocity $V(k)$ of the k^{th} link at $\mathcal{O}_k$ consists of the sum of the contribution from the motion of the $(k+1)^{th}$ link plus the additional relative spatial velocity $\Delta_V(k)$ between the two links generated by the joint velocity $\beta(k)$. Thus,

$$\begin{aligned} V(k) &= \phi^*(k+1,k)V(k+1) + H^*(k)\beta(k) \\ \alpha(k) &= \dot{V}(k) = \phi^*(k+1,k)\alpha(k+1) + H^*(k)\dot{\beta}(k) + a(k) \end{aligned} \tag{13}$$

where the velocity dependent Coriolis and centrifugal acceleration term

$$a(k) \stackrel{\triangle}{=} \dot{\phi}^*(k+1,k)V(k+1) + \dot{H}^*(k)\beta(k) \tag{14}$$

The suggestive notation $\phi(k+1,k)$ is used to denote the link transformation operator $\phi\Big(l(k+1,k)\Big)$. Note that $l(k+1,k)$ and hence $\phi(k+1,k)$ will depend on the configuration variable $\theta(k)$ in case the k^{th} joint contains a prismatic component.

Here, $f(k)$ will denote the spatial force of interaction between the $(k+1)^{th}$ and the k^{th} links at $\mathcal{O}_k$. The total external spatial force due to interactions with the inboard and outboard adjoining links acting on the k^{th} link about the point $\mathcal{O}_k$ is $[f(k) - \phi(k,k-1)f(k-1)]$. With $M(k)$ denoting the spatial inertia of the k^{th} link about the point $\mathcal{O}_k$, the equations of motion of the k^{th} link about $\mathcal{O}_k$ are:

$$f(k) = \phi(k,k-1)f(k-1) + M(k)\alpha(k) + b(k), \quad T(k) = H(k)f(k) \tag{15}$$

$$\text{where } b(k) \stackrel{\triangle}{=} \dot{M}(k)V(k) - \dot{\phi}\Big(p(k)\Big)M(k)V(k) \tag{16}$$

$T(k)$ is the joint force at the k^{th} joint, $b(k)$ is the velocity dependent gyroscopic force (see (11)), and $p(k)$ denotes the vector from the joint location $\mathcal{O}_k$ to the center of mass of the k^{th} link. Putting together Eqs. (13)–(15), leads to the following Newton-Euler recursive equations of motion for the serial multibody chain:

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

$$\left\{ \begin{aligned} & \qquad f(0) = 0 \\ & \textbf{for } \boldsymbol{k = 1 \cdots n} \\ & \quad f(k) = \phi(k,k-1)f(k-1) + M(k)\alpha(k) + b(k) \\ & \quad T(k) = H(k)f(k) \\ & \textbf{end loop} \end{aligned} \right. \tag{17}$$

But for the use of spatial notation here, these equations of motion are of the same form as those in Ref. 16. Without losing any generality, we have made the simplifying assumption that the tip force $f(0)$ is zero. Attaching a full 6 dof joint between the base and the inertial frame takes care of the mobile base situation. For the inverse dynamics problem, the joint accelerations $\dot{\beta}$ are known, and (17) represents an $O(\mathcal{N})$ algorithm involving a base-to-tip recursion sequence to compute the velocities and accelerations, followed by a tip-to-base recursion to compute the joint forces.

The *stacked notation* allows the equations of motion given by (17) to be expressed in a more concise form. In this notation, the $V(k)$'s, $\alpha(k)$'s, etc., are viewed as components of larger vectors V, α, etc. Thus for instance, $V \stackrel{\triangle}{=} [V^*(1) \ \cdots \ V^*(n)]^*$. Now (17) can be written as:

$$\begin{aligned} V &= \mathcal{E}_\phi^* V + H^*\beta, & \alpha &= \mathcal{E}_\phi^*\alpha + H^*\dot{\beta} + a \\ f &= \mathcal{E}_\phi f + M\alpha + b, & T &= Hf \end{aligned} \tag{18}$$

where the M and H are block diagonal matrices are defined as $M = diag[M(1) \ \cdots \ M(n)]$ and $H = diag[H(1) \ \cdots \ H(n)]$. Also,

$$\mathcal{E}_\phi \stackrel{\triangle}{=} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \phi(2,1) & 0 & \ldots & 0 & 0 \\ 0 & \phi(3,2) & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \phi(n,n-1) & 0 \end{pmatrix} \tag{19}$$

However, $\mathcal{E}_\phi$ is nilpotent (because $\mathcal{E}_\phi^n = 0$), and so

$$\phi \stackrel{\triangle}{=} (I - \mathcal{E}_\phi)^{-1} = I + \mathcal{E}_\phi + \mathcal{E}_\phi^2 + \ \cdots \ + \mathcal{E}_\phi^{n-1} = \begin{pmatrix} I & 0 & \ldots & 0 \\ \phi(2,1) & I & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi(n,1) & \phi(n,2) & \ldots & I \end{pmatrix} \tag{20}$$

where,

$$\phi(i,j) \stackrel{\triangle}{=} \phi(i,i-1) \ \cdots \ \phi(j+1,j) \quad \text{for} \quad i > j$$

Thus (18) can be rewritten in the form,

$$\begin{aligned} V &= \phi^* H^* \beta \\ \alpha &= \phi^*(H^*\dot{\beta} + a) \\ f &= \phi(M\alpha + b) = \phi M \phi^* H^* \dot{\beta} + \phi(M\phi^* a + b) \\ T &= Hf = H\phi M\phi^* H^*\dot{\beta} + H\phi(M\phi^* a + b) \\ &= \mathcal{M}\dot{\beta} + \mathcal{C}, \quad \text{where} \quad \mathcal{M} \stackrel{\triangle}{=} H\phi M \phi^* H^* \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}, \quad \text{and} \quad \mathcal{C} \stackrel{\triangle}{=} H\phi(M\phi^* a + b) \in \mathcal{R}^{\mathcal{N}} \end{aligned} \tag{21}$$

Here, $\mathcal{M}$ is the *mass matrix* of the serial chain and $\mathcal{C}$ consists of the velocity dependent Coriolis, centrifugal and gyroscopic joint forces. This is only a conceptual statement of the serial chain dynamics since $\mathcal{M}$ and $\mathcal{C}$ are typically not explicitly available, and may in general not even need to be computed. In the terminology of Kane's method (see Ref. 17), β are the *generalized speeds* and the elements of $\phi^* H^*$ are the *partial (spatial) velocities*.

The *spatial operators* $\mathcal{E}_\phi$, ϕ, H, and M, are the first we encounter. Recursive dynamical algorithms can be derived naturally by exploiting the special *state transition* properties of the elements of spatial operators such as $\mathcal{E}_\phi$, ϕ, etc. For instance, given a vector y, the evaluation of the matrix-vector product ϕy *does not require an $O(n^2)$ matrix-vector product computation, and not even the explicit computation of the elements* of ϕ, rather, it can be evaluated using an $O(n)$ recursive algorithm involving only the elements of $\mathcal{E}_\phi$ and y. This is precisely the correspondence between the concise operator based high-level description of the equations of motion in (21) and the recursive algorithmic description in (17).

The forward dynamics problem consists of solving (21) for $\dot{\beta}$ for a given state (θ, β) of the system and applied joint forces T. The $O(\mathcal{N}^3)$ and $O(\mathcal{N}^2)$ forward dynamics algorithms require the precomputation of the velocity dependent joint force term $\mathcal{C}$ to obtain a "bias free" joint force vector $\bar{T} \stackrel{\triangle}{=} T - \mathcal{C}$. The forward dynamics problem then reduces to solving the *linear matrix equation* given by

$$\mathcal{M}\dot{\beta} = \bar{T} \tag{22}$$

for $\dot{\beta}$. For a multibody system at rest, the velocity dependent terms a, b and $\mathcal{C}$ are all zero, thus working with $\bar{T}$ is conceptually equivalent to assuming that the multibody system is at rest. Initially, for ease of exposition, we derive even the $O(\mathcal{N})$ AB forward dynamics algorithm by assuming that $\bar{T}$ is available. Later we develop simple extensions to the AB algorithm, so that it works directly with T, and does not require the precomputation of $\mathcal{C}$ at all. Unless explicitly stated otherwise, from now on, α and f in (21) will denote quantities for the multibody system at rest (i.e. with a and b set to zero).

In the next sections, we discuss a variety of $O(\mathcal{N}^3)$, $O(\mathcal{N}^2)$ or $O(\mathcal{N})$ algorithms that have been proposed for solving the forward dynamics problem.

3 $O(\mathcal{N}^3)$ Forward Dynamics Algorithms

Forward dynamics algorithms based on the explicit computation of the mass matrix $\mathcal{M}$, followed by the $O(\mathcal{N}^3)$ process of solving the linear matrix equation in (22) form the class of $O(\mathcal{N}^3)$ algorithms. The various $O(\mathcal{N}^3)$ algorithms differ only in the method used to compute $\mathcal{M}$, and in this section we discuss three $O(\mathcal{N}^2)$ procedures proposed in Ref. 4 for computing the mass matrix.

Direct Method

The direct method relies on the fact that the k^{th} column of $\mathcal{M}$ is precisely $\mathcal{M}\dot{\beta}$ for a vector of joint accelerations $\dot{\beta}$ given by

$$\dot{\beta}(i) = \begin{cases} 0 & \text{for} \quad i \neq k \\ 1 & \text{for} \quad i = k \end{cases} \tag{23}$$

Thus, the k^{th} column of the mass matrix can be obtained by using the $O(\mathcal{N})$ Newton-Euler recursive inverse dynamics procedure to compute $\bar{T}$ for this set of joint accelerations. Carrying out this step $\mathcal{N}$ times for $k \in 1 \cdots \mathcal{N}$ leads to the computation of the $\mathcal{N}$ columns and the whole of $\mathcal{M}$. However, this method does not advantage of the structure of $\mathcal{M}$, and is not very efficient.

Symmetry Based Method

As discussed in the previous method, the k^{th} column of $\mathcal{M}$ is the vector of joint forces corresponding to the joint acceleration vector $\dot{\beta}$ in (23). However, for such a $\dot{\beta}$, the links from n through $(k+1)$ are stationary, and so the links k through 1 may be viewed as a new serial chain with k links and with unit joint acceleration at the k^{th} joint and zero acceleration at the other joints. Thus, the use of the $O(k)$ Newton-Euler recursive inverse dynamics procedure for this smaller chain of k links yields the k^{th} column of the *upper triangular half* of $\mathcal{M}$. Carrying out this step for $k \in 1 \cdots \mathcal{N}$ leads to the computation of the upper triangular half of $\mathcal{M}$ (and consequently all of $\mathcal{M}$, due to its symmetry). In the previously described direct method, the $\mathcal{N}$ inverse dynamics computations were of fixed complexity $O(\mathcal{N})$, and so the symmetry based algorithm is more efficient, since the complexity of the inverse dynamics computations here varies between $O(1)$ to $O(\mathcal{N})$.

Composite Rigid Body (CRB) Method

The composite rigid body (CRB) method takes advantage of the structure of the mass matrix to develop one of the most efficient methods for computing $\mathcal{M}$. The *composite rigid body inertia* $R(k)$ at the k^{th} joint is defined as the effective spatial inertia at $\mathcal{O}_k$ of the outboard links $k \ \cdots \ 1$, assuming that they form a composite rigid (augmented) body. For such a composite body, the following relationships hold for arbitrary $\alpha(k)$:

$$\begin{aligned} f(k) &= R(k)\alpha(k) = \phi(k,k-1)f(k-1) + M(k)\alpha(k) \\ f(k-1) &= R(k-1)\alpha(k-1), \quad \alpha(k-1) = \phi^*(k,k-1)\alpha(k) \end{aligned}$$

It follows then that the composite body inertias $R(k)$ and $R(k-1)$ at the k^{th} and $(k+1)^{th}$ adjacent joints are related as follows:

$$R(k) = \phi(k,k-1)R(k-1)\phi^*(k,k-1) + M(k) \tag{24}$$

This leads to the following recursive scheme for the computation of the CRB inertias:

$$\left\{ \begin{array}{l} \qquad R(0) = 0 \\ \textbf{for } \boldsymbol{k \ = \ 1 \cdots n} \\ \quad R(k) \ = \ \phi(k,k-1)R(k-1)\phi^*(k,k-1) + M(k) \\ \textbf{end loop} \end{array} \right. \tag{25}$$

The *composite rigid body inertia matrix* R is defined as the block diagonal matrix with the k^{th} block diagonal entry $R(k)$. The following lemma reveals the structure of the mass marix $\mathcal{M}$ and leads to the CRB algorithm for computing $\mathcal{M}$.

Lemma 3.1 *We have that*

$$M = R - \mathcal{E}_\phi R \mathcal{E}_\phi^* \tag{26}$$

$$\textit{and} \quad \phi M \phi^* = R + \tilde{\phi} R + R\tilde{\phi}^* \qquad \textit{where} \quad \tilde{\phi} \stackrel{\triangle}{=} \phi - I \tag{27}$$

Proof: It is a straightforward exercise to verify that (26) is merely a restatement in stacked notation of the recursions in (25). Moreover, from (20), $\phi\mathcal{E}_\phi = \mathcal{E}_\phi\phi = \phi - I = \tilde{\phi}$. Multiplying (26) from the left and right by ϕ and ϕ^* respectively leads to

$$\phi M\phi^* = \phi R\phi^* - \phi\mathcal{E}_\phi R\mathcal{E}_\phi^*\phi^* = (\tilde{\phi}+I)R(\tilde{\phi}+I)^* - \tilde{\phi}R\tilde{\phi}^* = R + \tilde{\phi}R + R\tilde{\phi}^* \qquad \blacksquare$$

Equation (26) is identical in form to the discrete Lyapunov Equations which arise in the theory of linear discrete-time systems. Note that the three terms on the right in (27) can be computed from just the knowledge of R. Moreover, the terms are block diagonal, strictly lowertriangular and strictly uppertriangular respectively. Due to its symmetry, the computation of $\phi M\phi^*$ thus requires the computation of R and just one of the two terms on the right in (27). Noting from (21) that $\mathcal{M} = H\phi M\phi^* H^*$ and putting together (25) and (27), leads to the following efficient CRB algorithm for computing $\mathcal{M}$:

$$\begin{cases} \quad R(0) = 0 \\ \textbf{for } \boldsymbol{k = 1 \cdots n} \\ \quad R(k) = \phi(k,k-1)R(k-1)\phi^*(k,k-1) + M(k) \\ \quad \mathcal{M}(k,k) = H(k)X(k) \\ \quad \begin{cases} X(k) = R(k)H^*(k) \\ \textbf{for } \boldsymbol{j = (k+1) \cdots n} \\ \quad X(j) = \phi(j,j-1)X(j-1) \\ \quad \mathcal{M}(j,k) = \mathcal{M}^*(k,j) = H(j)X(j) \\ \textbf{end loop} \end{cases} \\ \textbf{end loop} \end{cases} \tag{28}$$

Composite Rigid Body Inertias Based Inverse Dynamics

In this section, we use the composite body inertias to introduce the notion of alternative force decompositions, and also develop an alternative inverse dynamics algorithm. We begin with the following lemma.

Lemma 3.2 $\quad f = R\alpha + y \quad \textit{where} \quad y \stackrel{\triangle}{=} \tilde{\phi}RH^*\dot{\beta}$

Proof: Ignoring the velocity dependent Coriolis, etc., bias terms a and b, (21) and (27) imply that

$$f = \phi M\alpha = \phi M\phi^* H^*\dot{\beta} = [R\phi^* + \tilde{\phi}R]H^*\dot{\beta} = R\alpha + \tilde{\phi}RH^*\dot{\beta} \qquad \blacksquare$$

As a consequence of the direct mapping that exists between high-level spatial operator expressions and recursive algorithms, Lemma 3.2 leads to the following composite body inertias based $O(\mathcal{N})$ recursive inverse dynamics algorithm:

$$\left\{\begin{array}{rcl} & y(1) = 0 & \\ \textbf{for } \boldsymbol{k} & = & \mathbf{2 \cdots \boldsymbol{n}} \\ y(k) & = & \phi(k,k-1)\ [y(k-1) + R(k-1)H^*(k-1)\dot{\beta}(k-1)] \\ \textbf{end loop} & & \end{array}\right.$$

$$\left\{\begin{array}{rcl} & \alpha(n+1) = 0 & \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{n} \cdots \mathbf{1} \\ \alpha(k) & = & \phi^*(k+1,k)\alpha(k+1) + H^*(k)\dot{\beta}(k) \\ f(k) & = & R(k)\alpha(k) + y(k) \\ \bar{T}(k) & = & H(k)f(k) \\ \textbf{end loop} & & \end{array}\right. \tag{29}$$

In this scheme, the tip-to-base recursion for y and the base-to-tip recursion for α are completely decoupled from each other and can be carried out in parallel (Ref. 5). The spatial forces $f(k)$ are then obtained by simply an extra addition and the joint forces $\bar{T}(k)$ via a projection operation.

Lemma 3.2 decomposes the spatial force $f(k)$ at the k^{th} joint into the sum of the two components $R(k)\alpha(k)$ and $y(k)$, with the component terms having the following physical interpretations. The component $R(k)\alpha(k)$ arises from assuming that the outboard links form a composite body, i.e., the outboard joints have zero joint acceleration. This component is thus dependent only on the spatial acceleration at the k^{th} joint, and hence only on the joint accelerations of the inboard joints $\dot{\beta}(n)\cdots\dot{\beta}(k)$. The second component, $y(k)$, is a correction term which accounts for the presence of nonzero joint accelerations at the outboard joints and is a function of only these joint accelerations, $\dot{\beta}(k+1)\cdots\dot{\beta}(1)$. Note that this decomposition may be contrasted with the decomposition in (15), wherein the correction term $\phi(k,k-1)f(k-1)$ is a function of all the joint accelerations. The rationale of decomposing the spatial force at a joint into a linear combination of one term independent of the outboard known quantities (the joint accelerations in this case) and a second term which is a correction term, and purely a function of these outboard quantities, also forms the basis for the articulated body $O(\mathcal{N})$ forward dynamics algorithm (discussed in Section 5), where the known quantities are the joint forces.

4 $O(\mathcal{N}^2)$ Forward Dynamics Algorithms

The two general types of $O(\mathcal{N}^2)$ methods are the *conjugate gradient* based methods and a method based on solving in succession a set of *triangular equations*.

Conjugate Gradient Based Methods

A conjugate gradient based $O(\mathcal{N}^2)$ method for the solution of the forward dynamics algorithm for serial chains was originally proposed in Ref. 4. In general, the conjugate gradient method is an iterative procedure for the solution of a linear matrix equation $Ax = B$ when the coefficient matrix A is symmetric. The direct implementation of this method requires the computation of an $O(\mathcal{N}^2)$ matrix-vector product of the form Ay during each of $\mathcal{N}$ iterations, and this makes the conjugate gradient method an $O(\mathcal{N}^3)$ algorithm for solving symmetric matrix equations. However, in the forward dynamics context, it was recognized in Ref. 4 that even though the mass matrix $\mathcal{M}$ is not available, the product $\mathcal{M}y$ can be computed via an $O(\mathcal{N})$ inverse dynamics algorithm. This then converts the conjugate gradient algorithm into an $O(\mathcal{N}^2)$ algorithm for solving the forward dynamics problem.

The rate of convergence of the conjugate gradient procedure is a function of the condition number of the coefficient matrix, and the use of a preconditioning matrix Λ can be used to increase the convergence rate. In Ref. 5, the authors propose the use of the diagonal elements of the mass matrix as a preconditioning matrix. The choice of the diagonal elements is motivated by the following facts: (a) For many serial multibody systems the mass matrix is diagonally dominant and (b) The matrix Λ formed from the diagonal elements of the mass matrix is nothing but HRH^*, and it can be efficiently computed using the recursions in (25) for the composite rigid body inertia matrix R. The general conjugate gradient method from Ref. 18, for solving (22) and using a preconditioning matrix Λ is given below:

$$\left\{\begin{array}{rcl} & \dot\beta_0 = 0,\ r_0 = \bar{T},\ \gamma_1 = 0,\ p_0 = 0 & \\ \textbf{for}\ \boldsymbol{k} & \boldsymbol{=} & \boldsymbol{1\cdots n} \\ \text{if}\quad r_{k-1} & = & 0, \quad \text{then}\quad \dot\beta = \dot\beta_{k-1} \qquad \text{QUIT} \\ \text{Solve}\quad r_{k-1} & = & \Lambda z_{k-1}\ \text{ for }\ z_{k-1} \\ \gamma_k & = & z^*_{k-1} z_{k-1} / r^*_{k-2} r_{k-2} \\ p_k & = & z_{k-1} + \gamma_k p_{k-1} \\ s_k & = & z^*_{k-1} r_{k-1} / p^*_k \mathcal{M} p_k \\ \dot\beta_k & = & \dot\beta_{k-1} + s_k p_k \\ r_k & = & r_{k-1} - s_k \mathcal{M} p_k \\ \textbf{end loop} & & \end{array}\right. \tag{30}$$

$$\dot\beta = \dot\beta_n$$

When no preconditioning matrix is used, $\Lambda = I$ and this leads to the algorithm proposed in Ref. 4. With $\Lambda = HRH^*$ we obtain the algorithm in Ref. 5. Note that the choice of a block diagonal matrix for Λ implies that the solution of the linear equation $r_{k-1} = \Lambda z_{k-1}$ during each iteration can be carried out at low computational cost.

Triangularized Equations Method

Using general numerical techniques, the process of obtaining the LDU decomposition of $\mathcal{M}$ is an $O(\mathcal{N}^3)$ process. Based on the non-square factorization $\mathcal{M} = H\phi M\phi^* H^*$ of the mass matrix in (21), an alternative *square* factorization of $\mathcal{M}$ using spatial operator identities, is derived in Ref.

1 (and rederived in this paper in Lemma 6.2). This factorization is of the form

$$\mathcal{M} = [I + H\phi K]D[I + H\phi K]^*$$

and is known as the *innovations factorization* because of its relationship to the innovations approach to filtering and prediction theory (see Ref. 19). The factor $[I + H\phi K]$ is lower triangular, while D is block diagonal, and thus this factorization may be regarded as a block LDU factorization of $\mathcal{M}$. The elements of K and D can be computed recursively via an $O(\mathcal{N})$ process (shown in Sections 5 and 6), and the computation of $[I + H\phi K]$ is an $O(\mathcal{N}^2)$ process. Once the LDU factors are available, the solution of the linear equation for solving the forward dynamics is only an $O(\mathcal{N}^2)$ process. This factorization thus forms the basis for an $O(\mathcal{N}^2)$ forward dynamics procedure involving the successive solution of triangular systems of equations as outlined in the following two steps:

(a) Solve $[I + H\phi K]Dx = \bar{T}$ for x (b) Solve $[I + H\phi K]^*\dot{\beta} = x$ for $\dot{\beta}$

Reference 6 descibes an $O(\mathcal{N}^2)$ forward dynamics algorithm based on reducing the equations of motion to a triangular system for a serial chain with 1–dof joints. Kane's method was used to develop the equations of motion. The forward dynamics algorithm requires using an elimination process similar to the Gaussian elimination procedure to obtain a square triangular system of equations which is then solved by back substitution. Closer examination reveals that the various steps and computed quantities in this algorithm are similar to those that arise in the factorization based $O(\mathcal{N}^2)$ algorithm described above. The m_{kk} terms in Ref. 6 correspond to $D^{-1}(k)$ here, and the general m_{kj}/m_{kk} terms are precisely the elements of $[I + H\phi K]$.

5 $O(\mathcal{N})$ Forward Dynamics Algorithms

In this section, $O(\mathcal{N})$ algorithms for the computation of the forward dynamics of serial chains are discussed. We show that, aside from some minor differences, all the $O(\mathcal{N})$ algorithms are closely related and have the same inherent structure. To avoid repetitiousness, we first develop the $O(\mathcal{N})$*Articulated Body* (AB) forward dynamics algorithm based on first principles, and use it subsequently to discuss the other algorithms. For ease of exposition, we assume that $\mathcal{C}$ has been precomputed and work with $\bar{T}$. In the next section, this algorithm is shown to follow as a natural consequence of operator factorizations of $\mathcal{M}$ and $\mathcal{M}^{-1}$, and we also describe there simple extensions which allow the algorithm to work directly with T, and avoid the precomputation of $\mathcal{C}$.

Articulated Body (AB) Algorithm

The $O(\mathcal{N})$ AB algorithm is based on the use of the *articulated body inertias* of the serial chain. The forward dynamics problem is the converse of the inverse dynamics problem, with the known quantities being the joint forces $\bar{T}$, and the quantities to be computed the joint accelerations $\dot{\beta}$. Taking a cue from the discussion in Section 3 for the composite rigid body inertias based inverse dynamics algorithm, we look for a decomposition of the following form,

$$f(k) = P(k)\alpha(k) + z(k) \tag{31}$$

analogous to the one in Lemma 3.2, for the spatial force $f(k)$. Keeping in mind that now the joint forces are the known quantities, we require the first component $P(k)\alpha(k)$ to be independent of the outboard joint forces (i.e., as if these joint forces were zero), and the correction force $z(k)$ to be a function of only the outboard joint forces $\bar{T}(k-1)\cdots\bar{T}(1)$. Here $P(k)$ then is the effective spatial inertia at the k^{th} joint of the outboard links regarded as forming an articulated body (i.e., as if they formed a "floppy" serial chain with zero joint forces), and is referred to as the *articulated body inertia* at the k^{th} joint. On the other hand, $z(k)$ is the correction term which accounts for the existence of non-zero joint forces at the outboard joints. The roles of $P(k)$ and $z(k)$ are dual to those of $R(k)$ and $y(k)$ respectively in the inverse dynamics algorithm in Section 3. It is seen later, that $P(k)$ and $z(k)$ can be computed via $O(\mathcal{N})$ recursive algorithms. For now, let us assume that the $P(k)$'s and $z(k)$'s have been computed and are available, and we show how they lead to the $O(\mathcal{N})$ forward dynamics algorithm. From Eqs. (17) and (31) it follows that

$$\begin{aligned}
T(k) &= H(k)f(k) = H(k)P(k)\alpha(k) + H(k)z(k) \\
\Longrightarrow \epsilon(k) &\stackrel{\triangle}{=} T(k) - H(k)z(k) = D(k)\dot{\beta}(k) + H(k)P(k)\phi^*(k+1,k)\alpha(k+1) \\
& \qquad\qquad \text{where} \quad D(k) \stackrel{\triangle}{=} H(k)P(k)H^*(k) \\
\Longrightarrow \dot{\beta}(k) &= \nu(k) - K^*(k+1,k)\alpha(k+1)
\end{aligned} \tag{32}$$

where

$$\nu(k) \stackrel{\triangle}{=} D^{-1}(k)\epsilon(k), \quad K(k+1,k) \stackrel{\triangle}{=} \phi(k+1,k)G(k), \quad G(k) \stackrel{\triangle}{=} P(k)H^*(k)D^{-1}(k) \tag{33}$$

First, the $P(k)$'s and $z(k)$'s are used to compute $\nu(k)$'s and $K(k+1,k)$'s in (33) for all k. Recall from (17) that

$$\alpha(k) = \phi^*(k+1,k)\alpha(k+1) + H^*(k)\dot{\beta}(k) \tag{34}$$

and that $\alpha(n+1) = 0$. Now use (32) to compute $\dot{\beta}(n)$, then use (34) to compute $\alpha(n)$, then go back to (32) to compute $\dot{\beta}(n-1)$, and so on, recursively, until all the elements of $\dot{\beta}$ are computed. We now derive recursive algorithms for computing the $P(k)$'s and the $z(k)$'s.

Recursive Computation of $P(k)$'s

An induction based argument is used to derive recursive expressions for the articulated body inertias $P(k)$'s. Thus, we assume that the articulated body inertia $P(k)$ at $\mathcal{O}_k$ is known, and we proceed to derive an expression for $P(k+1)$. Since link 1 is a rigid body with no outboard links, $P(1) = M(1)$, and this suffices to start the induction argument. Assume that the links $k\cdots 1$ form an articulated body, and so $\bar{T}(k) = \cdots = \bar{T}(1) = 0$. Denoting the spatial acceleration on the + side of $\mathcal{O}_k$ (i.e., at $\mathcal{O}_k^+$, which is on the $(k+1)^{th}$ link side of the k^{th} joint) by $\alpha^+(k)$, it follows from (17) that

$$\alpha^+(k) = \phi^*(k+1,k)\alpha(k+1), \qquad \alpha(k) = \alpha^+(k) + H^*(k)\gamma(k) \tag{35}$$

$$f(k) = P(k)\alpha(k), \qquad \bar{T}(k) = H(k)f(k) = 0 \tag{36}$$

$$f(k+1) = \phi(k+1,k)f(k) + M(k+1)\alpha(k+1) \tag{37}$$

The missing equation in (37) is the relationship between $\alpha^+(k)$ and $\alpha(k)$, since the joint acceleration $\gamma(k)$ is not yet known. If the k^{th} joint were frozen, $\alpha^+(k) = \alpha(k)$, However, due to the presence of a joint, only a part of $\alpha^+(k)$ gets transmitted across the k^{th} joint as $\alpha(k)$. Equation (36) requires that $\alpha(k) \in Null[H(k)P(k)]$, and thus from (35) it follows that that $\alpha(k)$ is the *projection* of $\alpha^+(k)$ *along* $Range[H^*(k)]$ *onto* $Null[H(k)P(k)]$. That is, there exists a projection operator [‡] $\overline{\tau}^*(k)$ such that:

$$\alpha(k) = \overline{\tau}^*(k)\alpha^+(k) \quad \text{with} \quad \overline{\tau}^*(k) : \begin{cases} \overline{\tau}^*(k)H^*(k) &= 0 \\ H(k)P(k)\overline{\tau}^*(k) &= 0 \end{cases} \tag{38}$$

The specification of the left and right nullspaces of a projection operator uniquely specify it. That is, if Q is a projection operator whose left nullspace if specified by the columns of X, i.e., $X^*Q = 0$, and whose right nullspace is specified by the columns of Y, i.e., $QY = 0$, then Q is uniquely defined by

$$Q = I - Y(X^*Y)^{-1}X^*$$

The two conditions on the right in (38) are thus sufficient to uniquely define the *projection operator* $\overline{\tau}^*(k)$, and it is given by

$$\overline{\tau}^*(k) \quad = \quad I - H^*(k)[H(k)P(k)H^*(k)]^{-1}H(k)P(k) = I - H^*(k)G^*(k) \tag{39}$$

where $G(k)$ and $D(k)$ are as defined in (33). It is easy to verify that $\overline{\tau}(k) = I - G(k)H(k)$, and $\tau(k) = I - \overline{\tau}(k) = G(k)H(k)$ are also projection operators. Putting together (35)–(38), it follows that

$$\begin{aligned} f(k+1) \quad &= \quad [\phi(k+1,k)P(k)\overline{\tau}^*(k)\phi^*(k+1,k) + M(k+1)]\alpha(k+1) \\ \Longrightarrow P(k+1) \quad &= \quad \phi(k+1,k)P(k)\overline{\tau}^*(k)\phi^*(k+1,k) + M(k+1) \end{aligned} \tag{40}$$

and this is a recursive expression for $P(k+1)$. This equation is of the same form as the discrete Riccati equations that appear in the optimal filtering of discrete-time systems. It is easy to verify by substitution that

$$P^+(k) \quad \stackrel{\triangle}{=} \quad P(k)\overline{\tau}^*(k) = \overline{\tau}(k)P(k) = \overline{\tau}(k)P(k)\overline{\tau}^*(k) \tag{41}$$

and so (40) may be rewritten in the form

$$P(k+1) = \psi(k+1,k)P(k)\psi^*(k+1,k) + M(k+1) \quad \text{where} \quad \psi(k+1,k) \ \stackrel{\triangle}{=} \ \phi(k+1,k)\overline{\tau}(k) \tag{42}$$

Note the similarity of the recursion in (42) to the recursions for the composite body inertias $R(k)$ in (25). From (42) it is evident that $P(k+1)$ is symmetric and positive definite. Here, $\psi(k+1,k)$ is the force/velocity transformation operator for *articulated bodies* analogous to the transformation operator $\phi(k+1,k)$ for *composite bodies*. Note that, while the link transformation operator $\phi(k+1,k)$ is a function of $l(k+1,k)$ alone, $\psi(k+1,k)$ depends additionally on the joint types and spatial inertias of all the outboard links.

Note that the substitution of the expression for $\dot{\beta}(k)$ in (32) into (34) leads to

$$\alpha(k) = \psi^*(k+1,k)\alpha(k+1) + H^*(k)\nu(k)$$

Thus $\nu(k)$ is the relative joint acceleration for an *articulated body model* for the serial chain, analogous to $\dot{\beta}(k)$ for the *composite body model*.

[‡] Q is a projection operator if and only if $Q^2 = Q$.

Recursive Computation of $z(k)$'s

If the outboard joints were truly unactuated, i.e., there were no joint forces at the outboard joints, then from Eqs. (37) and (38) it follows that

$$\alpha(k) = \psi^*(k+1,k)\alpha(k+1) \implies f(k) = P(k)\psi^*(k+1,k)\alpha(k+1)$$

However, in the presence of non-zero joint forces $\bar{T}(k)\cdots\bar{T}(1)$, these equalities do not hold, and as a result, an extra force $z(k+1)$ is felt at the $(k+1)^{th}$ joint. Using Eqs. (31), (33), (38), (42) and (43), and a series of algebraic manipulations, the force $z(k+1)$ at the $(k+1)^{th}$ joint is given by

$$\begin{aligned} z(k+1) &= \phi(k+1,k)[f(k) - P(k)\psi^*(k+1,k)\alpha(k+1)] \\ &= \psi(k+1,k)z(k) + K(k+1,k)\bar{T}(k) \end{aligned} \tag{43}$$

Note that this recursive equation for $z(k)$ is similar in form to that for $y(k)$ in (29) in the composite rigid body inertias based inverse dynamics algorithm. Using (32), the following alternative form of the recursion for $z(k+1)$ can be obtained:

$$z(k+1) = \phi(k+1,k)z^+(k), \quad \text{where} \quad z^+(k) = z(k) + G(k)\epsilon(k) \tag{44}$$

Overall Articulated Body Forward Dynamics Algorithm

Gathering together the expressions derived in Eqs. (32)–(34), (39)–(41) and (44), we obtain the following $O(\mathcal{N})$ *Articulated Body (AB)* recursive forward dynamics algorithm:

$$\left\{\begin{array}{l} P^+(0) = 0 \\ \textbf{for } \boldsymbol{k} = \boldsymbol{1\cdots n} \\ \quad \begin{aligned} P(k) &= \phi(k,k-1)P^+(k-1)\phi^*(k,k-1) + M(k) \\ D(k) &= H(k)P(k)H^*(k) \\ G(k) &= P(k)H^*(k)D^{-1}(k) \\ K(k+1,k) &= \phi(k+1,k)G(k) \\ \overline{\tau}(k) &= I - G(k)H(k) \\ P^+(k) &= \overline{\tau}(k)P(k) \\ \psi(k+1,k) &= \phi(k+1,k)\overline{\tau}(k) \end{aligned} \\ \textbf{end loop} \end{array}\right. \tag{45}$$

$$\left\{\begin{array}{l} z^+(0) = 0 \\ \textbf{for } \boldsymbol{k} = \boldsymbol{1\cdots n} \\ \quad \begin{aligned} z(k) &= \phi(k,k-1)z^+(k-1) \\ \epsilon(k) &= \bar{T}(k) - H(k)z(k) \\ \nu(k) &= D^{-1}(k)\epsilon(k) \\ z^+(k) &= z(k) + G(k)\epsilon(k) \end{aligned} \\ \textbf{end loop} \end{array}\right. , \quad \left\{\begin{array}{l} \alpha(n+1) = 0 \\ \textbf{for } \boldsymbol{k} = \boldsymbol{n\cdots 1} \\ \quad \begin{aligned} \alpha^+(k) &= \phi^*(k+1,k)\alpha(k+1) \\ \dot{\beta}(k) &= \nu(k) - G^*(k)\alpha^+(k) \\ \alpha(k) &= \alpha^+(k) + H^*(k)\dot{\beta}(k) \end{aligned} \\ \textbf{end loop} \end{array}\right. \tag{46}$$

Note that the first tip to base recursion in (45) can be merged with the first one in (46) to obtain a single tip to base recursion sequence. Thus, the AB algorithm consists of a recursion from tip to base, followed by one from base to tip. A discussion of the relationship of a variety of $O(\mathcal{N})$ forward dynamics algorithms to the AB algorithm follows.

Vereshchagin's Algorithm

The first $O(\mathcal{N})$ algorithm to appear in the research literature was by Vereshchagin (see Ref. 7). The algorithm solved the forward dynamics problem for serial chains with 1–dof rotational and prismatic joints. Vereshchagin uses the fact that the dynamics of multibody systems evolve in time in accordance with Gauss's principle of least action. Treating a modified Gibbs-Appell function as a cost function on the joint accelerations, Vereshchagin then uses dynamic programming to recursively solve for the "optimal" joint accelerations. His algorithm is the AB algorithm restricted to the 1–dof joints case.

Armstrong's Algorithm

Reference 8 describes an $O(\mathcal{N})$ algorithm for the forward dynamics of serial chains with 3–dof spherical joints. It assumes the existence of a linear decomposition of the following form of the angular accelerations and the linear forces:

$$\dot{\omega}(k) \;=\; \Omega_\omega(k)\dot{v}(k) + \Upsilon_\omega(k), \quad \text{and} \quad F(k) = \Omega_F(k)\dot{v}(k) + \Upsilon_F(k) \tag{47}$$

$\Upsilon_\omega(k)$ and $\Upsilon_F(k)$ depend only on the moments $N(i),\ i \in k \cdots 1$, and the coefficient terms $\Omega_\omega(k)$ and $\Omega_F(k)$ are functions of the kinematical and inertial properties of the links in the serial chain. The following recursive scheme is used to compute the coefficients:

$$\left\{\begin{array}{rcl}
 & & \Omega_F(0) = 0, \Upsilon_F(0) = 0 \\
\textbf{for } \boldsymbol{k} & = & \mathbf{1} \cdots \boldsymbol{n} \\
\Gamma(k) & = & \Big[\mathcal{J}(k) - \tilde{l}(k,k-1)\Omega_F(k-1)\tilde{l}(k,k-1)\Big]^{-1} \\
\mathcal{L}(k) & = & -\Big[\tilde{l}(k,k-1)\Omega_F(k-1) + m(k)\tilde{p}(k)\Big] \\
\Omega_\omega(k) & = & \Gamma(k)\mathcal{L}(k) \\
\Upsilon_\omega(k) & = & \Gamma(k)\Big[-\tilde{l}(k,k-1)\Upsilon_F(k-1) + N(k) - N(k-1)\Big] \\
\Omega_F(k) & = & \Omega_F(k-1) + m(k)I - \mathcal{L}^*(k)\Gamma(k)\mathcal{L}(k) \\
\Upsilon_F(k) & = & \mathcal{L}^*(k)\Upsilon_\omega(k) \\
 & = & \Upsilon_F(k-1) - \mathcal{L}^*(k)\Gamma(k)\Big[-\tilde{l}(k,k-1)\Upsilon_F(k-1) + N(k) - N(k-1)\Big] \\
\textbf{end loop} & &
\end{array}\right. \tag{48}$$

Once these coefficients are computed, the forward dynamics problem is solved by a back-substitution based base to tip recursion to obtain the joint accelerations. The linear decomposition in (47) is similar to the idea of the decomposition of the spatial forces in (31) for the AB algorithm. In fact, when the AB algorithm is specialized to the 3–dof spherical joint case, the terms defined in (48) can be identified directly with the terms appearing in the AB algorithm. Rewriting (31) in partitioned form as:

$$\begin{pmatrix} N(k) \\ F(k) \end{pmatrix} = \begin{pmatrix} P_{11}(k) & P_{12}(k) \\ P_{12}^*(k) & P_{22}(k) \end{pmatrix} \begin{pmatrix} \dot{\omega}(k) \\ \dot{v}(k) \end{pmatrix} + \begin{pmatrix} z_N(k) \\ z_F(k) \end{pmatrix}$$

and after a series of algebraic manipulations, it can be shown that

$$\begin{aligned} P_{11}(k) &= \Gamma^{-1}(k) & P_{12}(k) &= -\mathcal{L}(k) \\ P_{22}(k) &= \Omega_F(k-1) + m(k)I & z_F(k) &= \Upsilon_F(k-1) \\ z_N(k) &= \tilde{l}(k,k-1)\Upsilon_F(k-1) + N(k-1) \end{aligned}$$

or alternatively,

$$\begin{aligned} \Omega_\omega(k) &= -P_{11}^{-1}(k)P_{12}(k) & \Omega_F(k) &= P_{22}(k) - P_{12}^*(k)P_{11}^{-1}(k)P_{12}(k) \\ \Upsilon_\omega(k) &= P_{11}^{-1}[N(k) - z_N(k)] & \Upsilon_F(k) &= z_F(k) + P_{12}^*(k)P_{11}^{-1}(k)[N(k) - z_N(k)] \end{aligned} \tag{50}$$

An appendix to Ref. 8 contains an $O(\mathcal{N})$ extension of this method to the 1–dof rotational joint case. However, this algorithm turns out to be involved and awkward, because the force decomposition from the 3–dof joints case is carried over to the 1–dof joints case too. During the earlier discussion for the AB algorithm it was seen that the appropriate force decomposition depends on the specific joints in the chain.

Featherstone's Algorithm

In Ref. 9, Featherstone introduces the spatial notation and uses the idea of a decomposition of the spatial force to develop a recursive $O(\mathcal{N})$ algorithm for 1–dof rotational and prismatic joint serial chains. Featherstone provides a great deal of physical insight by invoking the notion of articulated body inertias. Conceptually, the development of the AB algorithm presented here is a straightforward extension of the Featherstone algorithm from the 1–dof joints to the general joints case.

Rodriguez's Algorithm

The spatial operator algebra for multibody dynamics was initially developed in Ref. 10 and was based on the recognition of the parallels between the two-point boundary value problems that appear in multibody dynamics and the ones that are studied in optimal filtering theory. The area of Kalman filtering, with its rich body of analysis and computational techniques was thus available for application to problems in multibody dynamics. These techniques were used to obtain results such as novel factorizations and the explicit operator inversion of the mass matrix, which in turn led to the natural derivation of the recursive forward dynamics AB algorithm for the general joints case. The spatial operator algebra is discussed in more detail in Section 6.

Brandl, Johanni, and Otter's Algorithm

In Ref. 11, the authors use the spatial notation to derive the Newton-Euler equations of motion for the serial chain and use component level manipulations to derive the recursive forward dynamics AB algorithm for the general joints case.

Bae and Haug's Algorithm

In Ref. 12, the equations of motion for a serial chain are derived using a variational approach based on the virtual work form of D'Alembert's principle. The recursive forward dynamics algorithm that was subsequently derived is the AB algorithm for the cases of 1–dof joints and 2–dof cylindrical joints. One minor difference between the two algorithms is that, in Ref. 12, the generalized spatial forces $Q(.)$ at the centers of mass of the links rather than the joint forces $\bar{T}(.)$ are taken to be the known forces. For the sake of consistency with the rest of this paper, we assume that $Q(k)$ is referenced to the joint location $\mathcal{O}_k$ rather than the center of mass of the k^{th} link. The generalized force $Q(k)$ at $\mathcal{O}_k$ is the sum of the external forces acting on the k^{th} link and is given by

$$Q(k) = f(k) - \phi(k, k-1)f(k-1) \implies Q = (I - \mathcal{E}_\phi)f = \phi^{-1}f \tag{51}$$

where Q is a stacked vector representation of the generalized forces at the joint locations. Equation (51) implies that the spatial forces $f(k)$ and the generalized forces $Q(k)$ can be obtained from each other. We describe briefly the variational method used to derive the equations of motion. Taking a cue from the expression for the spatial velocity V in (21), it is easy to see that the virtual spatial displacement δ_x, for the points $\mathcal{O}_k$ along the whole serial chain is of the form $\delta_x = \phi^* H^* \delta_\beta$ for some δ_β. The virtual work form of D'Alembert's principle then states that the equations of motion of the system are given by

$$\begin{aligned} \delta_x^*(M\alpha - Q) &= 0 \quad \forall\, \delta_x \implies \delta_\beta^* H\phi(M\phi^* H^* \dot{\beta} - Q) = 0 \quad \forall\, \delta_\beta \\ \implies H\phi M \phi^* H^* \dot{\beta} &= H\phi Q \implies \mathcal{M}\dot{\beta} = Hf = \bar{T} \end{aligned} \tag{52}$$

We have used (21) and (51) in the last step. The above is a rederivation, based on the principle of virtual work, of the equations of motion in (22). The terms $Q(k)$ and $L(k)$ in the algorithm in Ref. 12 correspond to the terms $f(k)$ and $z(k)$ in the AB algorithm, and the following equation shows the simple relationship between them:

$$P(k)\alpha(k) = f(k) - z(k) = Q(k) + L(k) \tag{53}$$

Rosenthal's Algorithm

In Ref. 13, an $O(\mathcal{N})$ algorithm for the forward dynamics of serial chains with 1–dof joints was derived based on component level manipulation of the equations of motion obtained via Kane's method. This method is identical to the AB algorithm specialized to the 1 dof case.

Table 5 summarizes the notational relationship between the AB algorithm and the algorithms in Refs. 7, 9, 11, 12, 13. The notation used here is the same as in Rodriguez's algorithm in Ref. 2 except that in the latter the specific choice of $\beta = \dot{\theta}$ is used. The links are numbered in increasing order from tip to base in Ref. 2 while the reverse link numbering is used in the other references. This table is meant as an aid to help compare the algorithms, and a complete faithfulness to the individual notation is neither attempted nor possible due to the differences in their approaches. Thus, for instance, the derivations in Ref. 13 use partial velocities and momenta, and the notation in the table indicates these quantities rather than the velocities and forces.

Table 1: Summary of notation used in the various $O(\mathcal{N})$ algorithms

	Here	Ref. 7	Ref. 9	Ref. 11	Ref. 12	Ref. 13
Link index	$n-k+1$	k	k	k	k	
Joint rate var.	$\beta(k)$	$\dot{q}_k$	$\dot{q}_k$	$\dot{q}(k)$	$\dot{q}_{(k-1)k}$	u_k
Joint map	$H^*(k)$	B_k	$\hat{s}_k$	ϕ_k	$B_{(k-1)k2}$	$\sim \lambda_k$
Joint forces	$T(k)$	Q_k	Q_k	λ_k	$--$	τ_k
Link transf. op.	$\phi^*(k+1,k)$	A_k	${}_k\hat{X}_{k+1}$	C_{k+1}	$B_{(k-1)k1}$	$--$
Spatial accel.	$\alpha(k)$	$\ddot{x}_k$	$\hat{a}_k$	$\begin{pmatrix} \alpha_k \\ a_k^H \end{pmatrix}$	$\dot{Y}_k$	$\begin{pmatrix} {}^N\alpha^k \\ {}^N a^k \end{pmatrix}$
Spatial force	$f(k)$	$--$	$\hat{f}_k$	$\begin{pmatrix} \tau_k^h \\ f_k^h \end{pmatrix}$	$\sim Q(k)$, see (51)	$--$
Art. body inertia	$P(k)$	P_k	$\hat{I}_k^A$	I_k^*	$M_k + K_k$	M
Joint inertia	$D(k)$	$B_k^* P_k B_k$	$\hat{s}_k^* \hat{I}_k^A \hat{s}_k$	M_k	$B_{(k-1)k2}^* M_k B_{(k-1)k2}$	m_{kk}
Correction force	$z(k)$	R_k	$\hat{P}_k$	β_k^*	$\sim L(k)$, see (53)	$X(k)$

6 Factorization and Inversion of the Mass Matrix

As stated above, Ref. 10 describes the parallels between the structure of the dynamical equations for multibody systems and the equations that arise in the optimal estimation theory for discrete-time systems. Covariance factorization techniques from filtering theory (see Ref. 20) are applied in Ref. 1 to serial rigid multibody systems to obtain novel factorizations of the mass matrix and to obtain closed-form expressions for its inverse. These results reveal important new structural properties of the mass matrix, and they also directly lead to the $O(\mathcal{N})$ forward dynamics AB algorithm.

The *articulated body inertia matrix* P is defined as the block diagonal matrix with the k^{th} diagonal element being $P(k)$. The component level definitions in Eqs. (32), (33), (39), and (42) correspond to the following spatial operators in the stacked notation:

$$D \stackrel{\triangle}{=} HPH^*, \quad G \stackrel{\triangle}{=} PH^*D^{-1}, \quad K \stackrel{\triangle}{=} \mathcal{E}_\phi G, \quad \overline{\tau} \stackrel{\triangle}{=} I - GH, \quad \tau \stackrel{\triangle}{=} GH, \quad \mathcal{E}_\psi \stackrel{\triangle}{=} \mathcal{E}_\phi \overline{\tau} \tag{54}$$

Note that D, G, $\overline{\tau}$ and τ are all block diagonal with the k^{th} block diagonal entry being $D(k)$, $G(k)$, etc. On the other hand, the only nonzero block elements of K and $\mathcal{E}_\psi$ are are along their first lower subdiagonal, and are $K(k+1,k)$'s and $\psi(k+1,k)$'s respectively. Like $\mathcal{E}_\phi$, $\mathcal{E}_\psi$ is nilpotent ($\mathcal{E}_\psi^n = 0$), and so,

$$\psi \stackrel{\triangle}{=} (I-\mathcal{E}_\psi)^{-1} = I + \mathcal{E}_\psi + \mathcal{E}_\psi^2 + \ \cdots \ + \mathcal{E}_\psi^{n-1} = \begin{pmatrix} I & 0 & \dots & 0 \\ \psi(2,1) & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \psi(n,1) & \psi(n,2) & \dots & I \end{pmatrix} \tag{55}$$

where

$$\psi(i,j) \stackrel{\triangle}{=} \psi(i,i-1) \ \cdots \ \psi(j+1,j) \quad \text{for} \quad i > j$$

The operators $\mathcal{E}_\psi$ and ψ are the articulated body analogs corresponding to ϕ and $\mathcal{E}_\phi$ for composite bodies. Note that the elements of the operators ψ and $\mathcal{E}_\psi$ have the same state-transition properties as the elements of the operators ϕ and $\mathcal{E}_\phi$, and as a consequence, high-level operator expressions involving them can be directly mapped into recursive algorithms, and the explicit computation of the elements of the operator ψ is not required. We now establish the following identity.

Lemma 6.1 $\quad \psi^{-1} = \phi^{-1} + KH$

Proof: *From* (55), *we have that* $\psi^{-1} = I - \mathcal{E}_\psi = (I - \mathcal{E}_\phi) + \mathcal{E}_\phi \tau = \phi^{-1} + KH$ ∎

In contrast to the earlier nonsquare factorization of the mass matrix, $\mathcal{M} = H\phi M \phi^* H^*$, the following lemma yields an alternative factorization of $\mathcal{M}$ in terms of *square factors*.

Lemma 6.2 $\quad \mathcal{M} = [I + H\phi K] D [I + H\phi K]^*$

Proof: From Eqs. (41) and (42) we have that

$$\begin{aligned} M &= P - \mathcal{E}_\psi P \mathcal{E}_\phi^* = P - \mathcal{E}_\phi P \mathcal{E}_\phi^* + KDK^* \implies \phi M \phi^* = P + \tilde{\phi} P + P\tilde{\phi}^* + \phi K D K^* \phi^* \\ \implies \mathcal{M} &= H\phi M \phi^* H^* = H[P + \tilde{\phi} P + P\tilde{\phi}^* + \phi K D K^* \phi^*]H^* \\ &= D + H\phi K D + D K^* \phi^* H^* + H\phi K D K^* \phi^* H^* = [I + H\phi K] D [I + H\phi K]^* \qquad \blacksquare \end{aligned}$$

Note that $[I + H\phi K] \in \mathcal{R}^{\mathcal{N}\times\mathcal{N}}$ is square, lowertriangular and nonsingular, while D is a block diagonal matrix. This factorization may be regarded as a block LDU decomposition of $\mathcal{M}$ and forms the basis for the $O(\mathcal{N}^2)$ forward dynamics algorithm described in Section 4. The following lemma gives the closed form expression for the inverse of the factor $[I + H\phi K]$.

Lemma 6.3 $\qquad [I + H\phi K]^{-1} = [I - H\psi K]$

Proof: Using a standard matrix identity followed by Lemma 6.1, we have that

$$[I + H\phi K]^{-1} = I - H\phi[I + KH\phi]^{-1}K = I - H\phi[\psi^{-1}\phi]^{-1}K = I - H\psi K \qquad \blacksquare$$

Combining Lemma 6.2 and Lemma 6.3 leads to the following closed form operator expression for the inverse of the mass matrix.

Lemma 6.4 $\qquad \mathcal{M}^{-1} = [I - H\psi K]^* D^{-1} [I - H\psi K] \qquad \blacksquare$

Once again, note that $[I - H\psi K]$ is square, lowertriangular and nonsingular and so Lemma 6.4 may be regarded as providing a block LDU decomposition of $\mathcal{M}^{-1}$. The solution to the forward dynamics problem may now be written as

$$\dot{\beta} = \mathcal{M}^{-1}\bar{T} = [I - H\psi K]^* D^{-1} [I - H\psi K]\bar{T} \tag{56}$$

From the mapping that exists between high-level spatial operator expressions and recursive algorithms, it can be seen that the recursions in the $O(\mathcal{N})$ AB forward dynamics algorithm described in (46) correspond to the recursive implementation of (56). In fact, the expressions for the intermediate quantities in (46) are given by:

$$\begin{aligned} z &= \psi K \bar{T}, \qquad \epsilon = T - Hz = [I - H\psi K]\bar{T} \\ \nu &= D^{-1}\epsilon = D^{-1}[I - H\psi K]\bar{T}, \qquad \alpha = \psi^* H^* \nu = \psi^* H^* D^{-1} [I - H\psi K]\bar{T} \\ \dot{\beta} &= \nu - K^*\alpha = [I - H\psi K]^* D^{-1}[I - H\psi K]\bar{T} = \mathcal{M}^{-1}\bar{T} \end{aligned}$$

The AB algorithm thus follows naturally from the operator factorization and inversion of the mass matrix described above. Since concepts from filtering theory form the basis of the spatial operator algebra , we briefly describe the analogs of the terms that appear in the AB forward dynamics algorithm to those in stochastic filtering theory: $\mathcal{M}$ is the covariance of a discrete time stochastic process whose signal model has the state transition matrix ϕ, the observation matrix H, and input white noise whose covariance is M, Also, z is the state estimate, K the Kalman predictor gain, P the state estimation error covariance, ϵ the innovations process, α the co-state, and ψ the state transition matrix of the Kalman filter.

Full $O(\mathcal{N})$ Forward Dynamics Algorithm

For the sake of completeness, we derive the full forward dynamics algorithm which directly takes care of the Coriolis type terms a and b in the algorithm and does not require the precomputation of $\mathcal{C}$. Thus we now do away with the assumption that the multibody system is at rest, and allow the Coriolis and gyroscopic terms a, b and $\mathcal{C}$ to be non-zero. We have from (21) and Lemma 6.4 that

$$\dot{\beta} = \mathcal{M}^{-1}(T - \mathcal{C}) = [I - H\psi K]^* D^{-1}[I - H\psi K][T - H\phi(b + M\phi^* a)] \tag{57}$$

Lemma 6.5 $\qquad \dot{\beta} = [I - H\psi K]^* D^{-1}[T - H\psi(KT + b + Pa)] - \psi^* K^* a$

Proof: From Lemma 6.1 we have that

$$[I - H\psi K]H\phi = H[I - \psi K H]\phi = H[\psi\phi^{-1}]\phi = H\psi \tag{58}$$

Also, from Eqs. (41) and (42) it follows that

$$M = P - \mathcal{E}_\psi P \mathcal{E}_\phi^* \implies \psi M \phi^* = \psi P + P\tilde{\phi}^* \tag{59}$$

And, from Lemma 6.1,

$$[I - H\psi K]^* D^{-1} H P\tilde{\phi}^* = [I - H\psi K]^* K^* \phi^* = K^*[I - K H\psi]^* \phi^* = K^* \psi^* \tag{60}$$

Using Eqs. (58)–(60) in (57), the result follows. ∎

Based on Lemma 6.5, the full recursive algorithmic extensions of the recursions in (46) is obtained by replacing the equations for $z(k)$ and $\alpha(k)$ there by the following equations:

$$\begin{aligned} z(k) &= \phi(k, k-1)z^+(k-1) + P(k)a(k) + b(k) \\ \text{and} \quad \alpha(k) &= \alpha^+(k) + H^*(k)\dot{\beta}(k) + a(k) \end{aligned} \tag{61}$$

Other instances of the use of the spatial operator algebra can be found in Ref. 3 for robot dynamics, in Ref. 21 for general topology multibody dynamics, in Ref. 22 for flexible multibody systems, and in Ref. 23 for the decoupled control of multibody systems.

7 Conclusion

The main goal of this paper has been to analyze a variety of serial chain forward dynamics algorithms. The economy of representation and analysis tools provided by the spatial operator algebra have allowed a wide range of algorithms to be studied. There are three types of $O(\mathcal{N}^3)$ algorithms. The one based on composite rigid body inertias takes the most advantage of the structure of the mass matrix and is the most efficient. There are two classes of $O(\mathcal{N}^2)$ algorithms. The first consists of two variants of the conjugate-gradient based algorithms. The ones in the second class are based on solving triangularized systems of equations, and they are shown to be the same algorithm. Apart from minor differences, all the $O(\mathcal{N})$ algorithms are also shown to be the same algorithm.

The following set of equations succinctly summarizes the relationships among the classes of $O(\mathcal{N}^3)$, $O(\mathcal{N}^2)$ and $O(\mathcal{N})$ algorithms.

$$
\begin{array}{llcll}
(A) & \mathcal{M} & = & H\phi M\phi^* H^* & \rightarrow \; O(\mathcal{N}^3) \text{ Forward Dynamics} \\
(B) & & = & [I + H\phi K]D[I + H\phi K]^* & \rightarrow \; O(\mathcal{N}^2) \text{ Forward Dynamics} \\
(C) & [I + H\phi K]^{-1} & = & [I - H\psi K] & \\
(D) & \mathcal{M}^{-1} & = & [I - H\psi K]^* D^{-1} [I - H\psi K] & \rightarrow \; O(\mathcal{N}) \text{ Forward Dynamics}
\end{array}
$$

The equations of motion lead directly to the mass matrix expression in (A), which forms the basis for the $O(\mathcal{N}^3)$ algorithms such as the CRB algorithm. Further analysis results in the alternative factorization in (B), and the triangularized systems-based $O(\mathcal{N}^2)$ algorithm. The derivation of the explicit expression for the inverse of the factor in (C) results in the closed form expression for the mass matrix inverse in (D), and in turn to the $O(\mathcal{N})$ AB algorithm. The operator factorization and inversion results thus provide a unifying perspective within which diverse forward dynamics algorithms arise as a natural consequence of a progressive exploitation of the structure of the mass matrix.

8 Acknowledgement

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. The author gratefully acknowledges many helpful discussions with Dr. G. Rodriguez and Dr. K. Kreutz regarding the spatial operator algebra.

References

[1] Rodriguez, G., “Spatially Random Models, Estimation Theory, and Robot Arm Dynamics,” in *Proceedings of the Workshop on Space Telerobotics* (Rodriguez, G., ed.), Jet Propulsion Laboratory, Pasadena, CA, July 1987. (JPL Publication 87-13, 1987).

[2] Rodriguez, G. and Kreutz-Delgado, K., “Recursive Mass Matrix Factorization and Inversion: An Operator Approach to Manipulator Forward Dynamics,” 1992. (see also JPL Publication 88-11, 1988).

[3] Rodriguez, G., Kreutz-Delgado, K., and Jain, A., “A Spatial Operator Algebra for Manipulator Modeling and Control,” *The International Journal of Robotics Research*, Vol. 10, No. 4, Aug. 1991, pp. 371–381.

[4]Walker, M.W. and Orin, D.E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, No. 3, Sept. 1982, pp. 205–211.

[5]Fijany, A. and Scheid, R.E., "Efficient Conjugate Gradient Algorithms for Computation of the Manipulator Forward Dynamics," in *NASA Conference on Space Telerobotics* (Rodriguez, G. and Seraji, H., eds.), Pasadena, CA, 1989.

[6]Rosenthal, D.E., "Triangularization of Equations of Motion for Robotic Systems," *Journal of Guidance, Control and Dynamics*, Vol. 11, No. 3, June 1988, pp. 278–281.

[7]Vereshchagin, A.F., "Computer Simulation of the Dynamics of Complicated Mechanisms of Robot-Manipulators," *Engineering Cybernetics*, Vol. 6, 1974, pp. 65–70.

[8]Armstrong, W.W., "Recursive solution to the Equations of Motion of an N-link Manipulator," in *Fifth World Congress on Theory of Machines and Mechanisms*, Vol. 2, Montreal, Canada, pp. 1343–1346, 1979.

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

[10]Rodriguez, G., "Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 6, Dec. 1987, pp. 624–639.

[11]Brandl, H., Johanni, R., and Otter, M., "A Very Efficient Algorithm for the Simulation of Robots and similar Multibody Systems without Inversion of the Mass Matrix," in *IFAC/IFIP/IMACS Symposium (1st)*, Vienna, Austria, pp. 95–100, 1986.

[12]Bae, D.S. and Haug, E.J., "A Recursive Formulation for Constrained Mechanical System Dynamics: Part I. Open loop systems," *Mech. Struct. & Mach.*, Vol. 15, No. 3, 1987, pp. 359–382.

[13]Rosenthal, D., "Order N Formulation for Equations of Motion of Multibody Systems," in *SDIO/NASA Workshop on Multibody Simulations*, Arcadia, CA, Sept. 1987.

[14]Schwertassek, R. and Senger, K.H., "Representation of Joints in Multibody Systems," *Z. angew. Math. Mech.*, Vol. 68, No. 2, 1988, pp. 111–119.

[15]Jain, A., "Unified Formulation of Dynamics for Serial Rigid Multibody Systems," JPL Publication 89-43, Jet Propulsion Laboratory, Pasadena, CA, Nov. 1989.

[16]Luh, J.Y.S., Walker, M.W., and Paul, R.P.C., "On-line Computational Scheme for Mechanical Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 102, No. 2, June 1980, pp. 69–76.

[17]Kane, T.R. and Levinson, D., *Dynamics: Theory and Applications.* McGraw-Hill, New York, 1985.

[18]Golub, G.H. and Van Loan, C.F., *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, 1983.

[19]Kailath, T., "The Innovations Approach to Detection and Estimation Theory," *Proceedings of the IEEE*, Vol. 58, No. 5, Mar. 1970, pp. 680–695.

[20]Gevers, M.R. and Kailath, T., "An Innovations Approach to Least-Squares Estimation—Part VI: Discrete-Time Innovations Representations and Recursive Estimation," *IEEE Transactions on Automatic Control*, Vol. AC-18, No. 6, Dec. 1973, pp. 588–600.

[21]Rodriguez, G., Jain, A., and Kreutz-Delgado, K., "Spatial Operator Algebra for Multibody System Dynamics," *Journal of the Astronautical Sciences*, Vol. 40, No. 1, Jan.–March 1992, pp. 27–50.

[22]Rodriguez, G., "Spatial Operator Approach to Flexible Multibody Manipulator Inverse and Forward Dynamics," in *IEEE International Conference on Robotics and Automation*, Cincinnati, OH, May 1990.

[23]Rodriguez, G., "An Innovations Approach to Decoupling of Manipulator Dynamics and Control," in *Proceedings of the 3rd Annual Conference on Aerospace Computational Control*, Oxnard, CA, Aug. 1989. (JPL Publication 89–45, 1989).

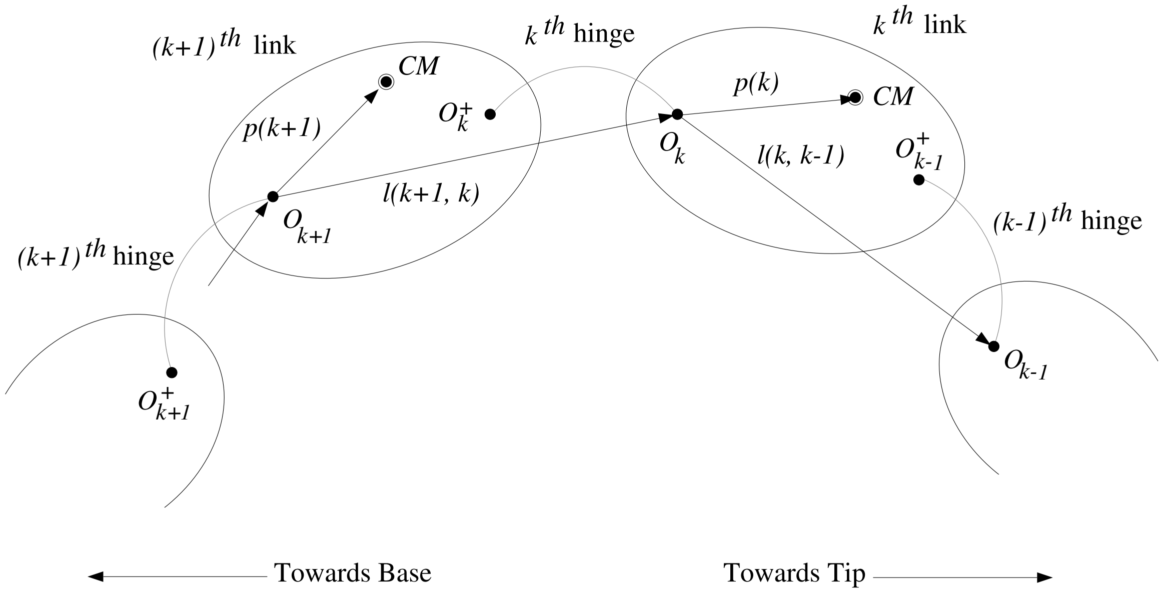

Figure 1: Illustration of Links and Joints in a Serial Rigid Multibody System