G. Rodriguez
A. Jain
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

K. Kreutz-Delgado
AMES Department
University of California, San Diego
La Jolla, California

A Spatial Operator Algebra for Manipulator Modeling and Control

Abstract

A recently developed spatial operator algebra for manipulator modeling, control, and trajectory design is discussed. The elements of this algebra are linear operators whose domain and range spaces consist of forces, moments, velocities, and accelerations. The effect of these operators is equivalent to a spatial recursion along the span of a manipulator. Inversion of operators can be efficiently obtained via techniques of recursive filtering and smoothing. The operator algebra provides a high-level framework for describing the dynamic and kinematic behavior of a manipulator and for control and trajectory design algorithms. The interpretation of expressions within the algebraic framework leads to enhanced conceptual and physical understanding of manipulator dynamics and kinematics. Furthermore, implementable recursive algorithms can be immediately derived from the abstract operator expressions by inspection. Thus the transition from an abstract problem formulation and solution to the detailed mechanization of specific algorithms is greatly simplified.

1. Introduction: A Spatial Operator Algebra

A new approach to the modeling and analysis of systems of rigid bodies interacting among themselves and their environment has recently been developed in Rodriguez (1987a) and Rodriguez and Kreutz (1990b). This work develops a framework for clearly understanding issues relating to the kinematics, dynamics, and control of manipulators in dynamic interaction with each other, while keeping the complexity involved in analyzing such systems to manageable proportions.

The International Journal of Robotics Research,
Vol. 10, No. 4, August 1991,

The analysis given in Rodriguez (1987a) and Rodriguez and Kreutz (1990b) has shown that certain linear operators are always present in the dynamic and kinematic equations of robot arms. These operators are called *spatial operators*, because they show how forces, velocities, and accelerations propagate through space from one rigid body to the next. Not only do the operators have obvious physical interpretations, but they are also implicitly equivalent to tip-to-base or base-to-tip recursions, which if needed, can be immediately turned into implementable algorithms by projecting them onto appropriate coordinate frames.

Compositions of spatial operators, when allowed to operate on functions of the joint velocities and accelerations, result in the dynamic equations of motion that arise from a Lagrangian analysis. The fact that the operators have equivalent recursive algorithms is a generalization of the well-known equivalence [described in Silver (1982)] between the Lagrangian and recursive Newton-Euler approaches to manipulator dynamics. The operator-based formulation of robot dynamics leads to an integration of these two approaches, so that analytic expressions can be shown to almost always have implicit, and obvious, recursive equivalents that are straightforward to mechanize.

The essential ingredients of the operator algebra are the operations of addition and multiplication (Roman 1975; Rudin 1973). There is also an "adjoint," or "*", operator that can operate on elements of the spatial algebra. If a spatial operator A is "causal" in the sense that it implies an inward recursion, then its adjoint A^* is "anticausal." An anticausal operation implies an outward recursion. Operator inversion is also defined in the spatial operator algebra. For an arbitrary finite-dimensional linear operator, inversion is achieved by the traditional techniques of linear algebra. However, many important spatial operators encountered in multibody dynamics belong to a class that can be factored as the product of a causal operator, a diagonal operator, and an anticausal operator. For these operators, inversion can often be achieved using the inward/outward sweep solutions of spatially recursive Kalman filtering and smoothing described in Rodriguez (1987a), Rodriguez and Kreutz (1990b), and Anderson and Moore (1979).

That the equations of multibody dynamics can be completely described by an algebra of spatial operators is certainly of mathematic interest. However, the significance of this result goes beyond the mathematics and is useful in a very practical sense. The spatial operator algebra provides a convenient means to manipulate the equations describing multibody behavior at a very high level of abstraction. This liberates the user from the excruciating detail involved in more traditional approaches to multibody dynamics where often one "can't see the forest for the trees." Furthermore, at any stage of an abstract manipulation of equations, spatially recursive algorithms to implement the operator expressions can be readily obtained by inspection. Therefore the transition from abstract operator mathematics to practical implementation is straightforward to perform and often requires only a simple mental exercise. When applied to the dynamic analysis of a manipulator with $n =$ links, the algebra typically leads to $O(n)$ recursive algorithms. However, numeric efficiency is not the main motivation for its development. What the algebra primarily offers is a mathematic framework that, because of its simplicity, is believed to have great potential for addressing advanced control and motion planning problems (Rodriguez 1989c).

To illustrate the use of the spatial operators, several applications of the algebra to robotics will be presented: (1) an operator representation of the manipulator Jacobian matrix; (2) the robot dynamic equations formulated in terms of the spatial algebra, showing the equivalence between the recursive Newton-Euler and Lagrangian formulations of robot dynamics in a far more transparent way than before; (3) the operator factorization and inversion of the manipulator mass matrix, which immediately results in $O(n)$ recursive forward dynamics algorithms for a serial manipulator; (4) the joint accelerations of a manipulator caused by a tip contact force; (5) the recursive computation of the equivalent mass matrix as seen at the tip of a manipulator, referred to by Khatib (1985) as the *operational space inertia matrix*; (6) recursive forward dynamics of a closed-chain system. Finally, we discuss additional applications and research involving the spatial operator algebra.

2. The Jacobian Operator

Consider an n-link serial chain manipulator. After defining a link spatial velocity to be $V(k) = \text{col}[\omega(k), \nu(k)] \in R^6$, the recursion that describes the relationship between joint angle rates, $\dot{\theta} = \text{col}[\dot{\theta}(1), \ldots, \dot{\theta}(n)]$, and link velocities, $V = \text{col}[V(1), \ldots, V(n)]$ is (Rodriguez and Kreutz 1990b; Craig 1986):

$$\begin{cases} \quad V(n+1) = 0 \\ \textbf{for k = n} \cdots \textbf{1} \\ \quad V(k) = \phi^*(k+1, k)V(k+1) + H^*(k)\dot{\theta}(k) \\ \textbf{end loop} \end{cases}$$

$$V(0) = \phi^*(1,0)V(1)$$

$H(k) = [h^*(k)\ 0\ 0\ 0]$ where $h(k) \in R^3$ is the unit vector in the direction of the kth joint axis. $\phi(k+1, k)$ is defined as

$$\phi(k+1, k) = \begin{pmatrix} I & \tilde{l}(k+1, k) \\ 0 & I \end{pmatrix},$$

where $l(k+1, k)$ is the vector from the $(k+1)$th joint to the kth joint. Thus $\phi^*(k+1, k)$ is the Jacobian that transforms velocities across a rigid link. This recursion represents a base-to-tip recursion that shows how link velocities propagate outward to the tip, point "0" on link 1, from the base "link $n + 1$." This assumes for simplicity that the base has zero velocity. Note that the link numbering convention used here and in Rodriguez (1987a) and Rodriguez and Kreutz (1990b) increases from the tip to the base, unlike the numbering convention described in most robotics textbooks such as Craig (1986). This convention makes it easier to describe the recursive algorithms presented in this article.

Summation of the preceding recursion leads to

$$V(k) = \sum_{i=k}^{n} \phi^*(i, k)H^*(i)\dot{\theta}(i),$$

where the facts that $\phi(i, i) = I$ and $\phi(i, j) \cdot \phi(j, k) = \phi(i, k)$ have been used. Also note that $\phi^{-1}(i, j) = \phi(j, i)$. This naturally suggests that we define the "operators" $H^* = \text{diag}[H^*(1), \ldots, H^*(n)]$, $B^* = [\phi^*(1, 0), 0, \ldots, 0]$ and

$$\phi \triangleq \begin{pmatrix} I & 0 & 0 & \ldots & 0 \\ \phi(2, 1) & I & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \phi(n, 1) & \phi(n, 2) & \ldots & \ldots & I \end{pmatrix}$$

This results in $V(0) = B^*\phi^*H^*\dot{\theta}$ or

$$V(0) = J\dot{\theta}, \qquad \text{where } J = B^*\phi^*H^* \tag{1}$$

The Jacobian operator J in eq. (1) is seen to be the product of three operators: B^*, ϕ^*, and H^*. The operator H^*, being block diagonal, is called *memoryless,* or nonrecursive. The operator B^* projects out the link 1 velocity $V(1)$ of the composite velocity V and propagates it to the tip location at point 0. The operator ϕ is lower block triangular, which we denote as "causal," and ϕ^* is upper block triangular and hence "anticausal." ϕ^* represents a propagation of link velocities from the base to the tip, which is viewed as the anticausal direction, as opposed to the tip-to-base recursion represented by ϕ, which is denoted as causal.

The action of the Jacobian operator on the joint angle rates $\dot{\theta}$ then is as follows: (1) $H^*\dot{\theta}$ results in relative spatial velocities between the links along the joint axes; (2) ϕ^* then anticausally propagates these relative velocities from the base to the tip to form the link spatial velocities $V = \text{col}[V(1), \ldots, V(n)]$; and (3) B^* then projects out $V(1)$ from V and propagates it to the tip forming $V(0)$.

The well-known (Craig 1986) dual relationship to $V(0) = J\dot{\theta}$ is $T = J^*f(0) = H\phi Bf(0)$, where $f(0) = \text{col}[N(0), F(0)] \in R^6$ is a spatial force that represents the tip interaction with the environment. The action of J^* on $f(0)$ is as follows: (1) B takes $f(0)$ to $\text{col}[f(1), 0, \ldots, 0]$; (2) ϕ propagates $f(1)$ causally from link 1 to the base forming the interaction spatial forces between neighboring links represented by $f = \text{col}[f(1), \ldots, f(n)]$; and (3) H projects each component of f, $f(k)$ onto joint axis $H^*(k) = h(k)$ to obtain the joint moments $T = \text{col}[T(1), \ldots, T(n)]$.

The key points to note here are that J and J^* have operator factorizations that have immediate physical interpretations and obvious recursive algorithmic equivalents. Working with the factorized version of J, one can manipulate expressions involving J in new ways while maintaining the physical insight provided by the factors and the ability to produce equivalent recursive algorithms at key steps of a calculation. For example, using the techniques of the spatial operator algebra, one can find algorithms for efficient recursive construction of J, JJ^*, J^*J, and (when an arm is nonredundant and nonsingular) $(J^*J)^{-1}$ (see Rodriguez and Scheid 1987).

3. An Operator-Formulated Robot Dynamics

Consider the following equations of motion for an n-link serial manipulator in a gravity-free environment with the tip imparting a spatial force $f(0)$ to the external environment:

$$\mathcal{M}\ddot{\theta} + \mathcal{C} + J^*f(0) = T. \tag{2}$$

$\mathcal{C}$ denotes "bias" torques caused by the velocity-dependent Coriolis and centrifugal effects. Eq. (2) is precisely the form that arises from a Lagrangian analysis of manipulator dynamics. Eq. (2) has an operator interpretation that arises from the following spatial operator factorizations of $\mathcal{M}$, $\mathcal{C}$, and J^*:

$$\mathcal{M} = H\phi M\phi^*H^* \tag{3}$$

$$\mathcal{C} = H\phi(M\phi^*a + b) \tag{4}$$

$$J^* = H\phi B \tag{5}$$

These factorizations are derived in Rodriguez and Kreutz (1990b). The mass matrix factorization in eq. (3) is called the *Newton-Euler factorization,* for reasons to be discussed later. The quantity

$$M = \text{diag}[M(1), \ldots, M(n)]$$

is made up of the spatial inertia $M(k)$ associated with each link of the manipulator. M, being block diagonal, is interpreted as a memoryless operator. For a given link k, $M(k)$ has the form

$$M(k) = \begin{pmatrix} \mathcal{J}(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)I \end{pmatrix}$$

where $\mathcal{J}(k)$ is the inertia tensor of link k about joint k; $m(k)$ is the link k mass; and $p(k)$ is the 3-vector from joint k to the link k mass center. The "tilde" operator is defined by $\tilde{x}y = x \times y$ for any 3-vectors x and y. In eq. (4), $a = \text{col}[a(1), \ldots, a(n)]$ and $b = \text{col}[b(1), \ldots, b(n)]$ are known quadratic functions of the link spatial velocities. The operators H, ϕ, and B were described in the previous section.

When eq. (2) is given an operator interpretation via eqs. (3)–(5), it is immediately apparent that eq. (2) is functionally identical to the Newton-Euler recursions given in Rodriguez and Kreutz (1990b), Craig (1986), and Luh et al. (1980):

$$\begin{cases} \alpha(n+1) = 0 \\ \textbf{for k = n} \cdots \textbf{1} \\ \quad \alpha(k) = \phi^*(k+1, k)\alpha(k+1) + H^*(k)\ddot{\theta}(k) + a(k) \\ \textbf{end loop} \end{cases}$$

$$\begin{cases} f(0) = f_{\text{ext}} \\ \textbf{for k = 1} \cdots \textbf{n} \\ \quad f(k) = \phi(k, k-1)f(k-1) + M(k)\alpha(k) + b(k) \\ \quad T(k) = H(k)f(k) \\ \textbf{end loop} \end{cases}$$

where $\alpha = \text{col}[\alpha(1), \ldots, \alpha(n)]$, and $\alpha(k) = \dot{V}(k)$ denotes the spatial acceleration of link k.

To make this equivalence clearer, consider the "bias-free" manipulator dynamics given by

$$\mathcal{M}\ddot{\theta} = T' \tag{6}$$

This corresponds to taking $a = 0$, $b = 0$, and $f(0) = 0$ in the Newton-Euler recursions. Eq. (6) is also valid for the case when the Coriolis, centrifugal, and tip-contact force terms have been subtracted out of eq. (2), resulting in $T' = T - \mathcal{C} - J^*f(0)$. From the Newton-Euler factorization in eq. (3) we see that eq. (6) is equivalent to

$$H\phi M\phi^* H^*\ddot{\theta} = T'. \tag{7}$$

The action of H^* on the joint angle accelerations $\ddot{\theta}$ is memoryless (nonrecursive) and results in a vector of relative spatial accelerations between the manipulator links. The action of ϕ^* on $H^*\ddot{\theta}$ is equivalent to an anticausal base-to-tip recursion that propagates link relative accelerations resulting in all the link spatial accelerations α. The combined action of ϕ^* and H^* on $\ddot{\theta}$, denoted by $\phi^*H^*\ddot{\theta}$, is equivalent to the recursion

$$\begin{cases} \quad \alpha(n+1) = 0 \\ \textbf{for k = n} \cdots \textbf{1} \\ \quad \alpha(k) = \phi^*(k+1, k)\alpha(k+1) + H^*(k)\ddot{\theta}(k) \\ \textbf{end loop} \end{cases}$$

The action of M on $\alpha = \phi^*H^*\ddot{\theta}$ is memoryless and leads to the D'Alembert forces col$[M(k)\alpha(k)]$, which represent the net spatial forces acting on each of the links. The action of ϕ on $M\alpha$ is equivalent to a causal tip-to-base recursion of all the single-link D'Alembert forces to form the link interaction spatial forces $f = \phi M\alpha$ acting on the manipulator links. Finally, the action of $H = \text{diag}[H(1), \ldots, H(n)]$ on f is to project the link spatial forces $f(k)$ onto the joint axes $H^*(k)$ to obtain the joint moments $T = Hf = \text{col}[T(k)]$, $T(k) = H(k)f(k)$. The combined actions of H, ϕ, and M on α, denoted by $H\phi M\alpha$, is equivalent to the recursion

$$\begin{cases} \quad f(0) = 0 \\ \textbf{for k = 1} \cdots \textbf{n} \\ \quad f(k) = \phi(k, k-1)f(k-1) + M(k)\alpha(k) \\ \quad T(k) = H(k)f(k) \\ \textbf{end loop} \end{cases}$$

This establishes the equivalence between the Lagrangian and recursive Newton-Euler formulations of manipulator dynamics (Silver 1982) and justifies the use of the terminology *Newton-Euler factorization* for eq. (3).

The factorizations given by eqs. (3)–(5) allow us to manipulate the dynamic equations of motion in ways not previously apparent. The fact that each factor has an interpretation as a causal, memoryless, or anticausal recursion of spatial quantities means that at any point of the mathematic analysis, one can interpret expressions in a deeply physical way or immediately produce an equivalent recursive algorithm. In the following sections it will be shown that an important alternative factorization to the Newton-Euler factorization [eq. (3)] exists that results in new causal, memoryless, anticausal operators with corresponding equivalent recursions. Also, we will discuss the existence of very useful operator identities that allow one to manipulate kinematic and dynamic equations in ways that would be otherwise impossible, all the while keeping the correspondence of abstract mathematical expressions to equivalent implementable algorithms.

4. Operator Inversion of the Manipulator Mass Matrix

From eq. (3), the well-known fact that $\mathcal{M}$ is symmetric and positive definite can be easily seen. It is also well known that a symmetric positive-definite operator is a covariance for some Gaussian random process. A deeper result is that the factorization given by eq. (3) shows that $\mathcal{M}$ has the structure of a covariance of the output of a discrete-step causal finite-dimensional linear system whose input is a Gaussian white-noise process. This a very important fact, for it is well known (Rodriguez 1990a) that such an operator can be factored and inverted efficiently by the use of standard techniques from filtering and estimation theory. Applications of these techniques to the manipulator mass matrix can be found in Rodriguez and Kreutz (1990b) and are partially summarized in this section.

First we present an important alternative factorization to eq. (3). To this end, we define

$$D \triangleq HPH^*, \quad G \triangleq PH^*D^{-1},$$

$$\mathcal{E}_\phi \triangleq \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \phi(2,1) & 0 & \cdots & 0 & 0 \\ 0 & \phi(3,2) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \phi(n, n-1) & 0 \end{pmatrix},$$

and

$$K \triangleq \mathcal{E}_\phi G$$

$$= \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ K(2,1) & 0 & \cdots & 0 & 0 \\ 0 & K(3,2) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & K(n, n-1) & 0 \end{pmatrix}$$

Note that $K(i, i-1) = \phi(i, i-1)G(i-1)$. $P \triangleq$

diag[$P(1), \ldots, P(n)$], where the diagonal elements $P(k)$ are obtained by the following causal discrete-step Riccati equation

$$\begin{cases} P(1) = M(1) \\ \textbf{for k = 2} \cdots \textbf{n} \\ P(k) = \psi(k, k-1)P(k-1)\psi^*(k, k-1) + M(k) \\ \textbf{end loop} \end{cases} \tag{8}$$

where

$$\psi(k, k-1) = \phi(k, k-1)[I - G(k-1)H(k-1)] \tag{9}$$

$P(k)$ is always symmetric positive definite, and hence D, which is diagonal with the positive diagonal elements $D(k) = H(k)P(k)H^*(k)$, is always invertible.

In a fashion analogous to the definitions of ϕ and $\mathscr{E}_\phi$, we define ψ and $\mathscr{E}_\psi$:

$$\psi \triangleq \begin{pmatrix} I & 0 & 0 & \cdots & 0 \\ \psi(2,1) & I & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \psi(n,1) & \psi(n,2) & \cdots & \cdots & I \end{pmatrix},$$

$$\mathscr{E}_\psi \triangleq \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ \psi(2,1) & 0 & \cdots & 0 & 0 \\ 0 & \psi(3,2) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \psi(n, n-1) & 0 \end{pmatrix}$$

where $\psi(k, k-1)$ is given by eq. (9), $\psi(k, k) = I$, and

$$\psi(i, j) = \psi(i, i-1)\psi(i-1, i-2) \cdots \psi(j+1, j)$$

for $i \geq j$.

With these definitions, we can restate the definition in eq. (9) as

$$\mathscr{E}_\psi = \mathscr{E}_\phi(I - GH) = \mathscr{E}_\phi - KH. \tag{10}$$

The action of ψ on a composite spatial quantity y to form $z = \psi y$ is equivalent to the following causal tip-to-base recursion

$$\begin{cases} z(0) = 0 \\ \textbf{for k = 1} \cdots \textbf{n} \\ z(k) = \psi(k, k-1)z(k-1) + y(k) \\ \textbf{end loop} \end{cases}$$

LEMMA 1: An alternative factorization of $\mathcal{M} = H\phi M\phi^* H^*$ is the innovations factorization

$$\mathcal{M} = (I + H\phi K)D(I + H\phi K)^* \tag{11}$$

where $I + H\phi K$ is causal (lower triangular), and D is memoryless, diagonal, and invertible.

Proof: See appendix.

□

The innovations factorization eq. (11) is equivalent to viewing the mass operator $\mathcal{M}$ as the covariance of a filtered innovations process, y. In stochastic estimation theory, the innovations representation is given by the causal operator $I + H\phi K$ operating on an innovations process $\varepsilon = \text{diag}[\varepsilon(1), \ldots, \varepsilon(n)]$, which can be taken to be an independent Gaussian sequence. The action of $(I + H\phi K)$ on ε,

$$y = (I + H\phi K)\varepsilon$$

is equivalent to the following causal tip-to-base recursion

$$\begin{cases} \hat{z}(0) = 0; \quad \varepsilon(0) = 0 \\ \textbf{for k = 1} \cdots \textbf{n} \\ \hat{z}(k) = \phi(k, k-1)\hat{z}(k-1) \\ \qquad + K(k, k-1)\varepsilon(k-1) \\ y(k) = H(k)\hat{z}(k) + \varepsilon(k) \\ \textbf{end loop} \end{cases}$$

The importance of the innovations operator $I + H\phi K$ is that it is trivially and causally invertible and that its inverse is precisely a discrete-step Kalman filter viewed as a whitening filter.

LEMMA 2: The causal (lower triangular) operators $I + H\phi K$ and $I - H\psi K$ are mutual causal inverses of each other

$$(I + H\phi K)^{-1} = I - H\psi K. \tag{12}$$

Proof: See appendix.

□

The relationship $\varepsilon = (I + H\phi K)^{-1}y = (I - H\psi K)y$ is equivalent to the following causal tip-to-base recursion:

$$\begin{cases} \hat{z}(0) = 0; \quad y(0) = 0 \\ \textbf{for k = 1} \cdots \textbf{n} \\ \hat{z}(k) = \psi(k, k-1)\hat{z}(k-1) \\ \qquad + K(k, k-1)y(k-1) \\ \varepsilon(k) = -H(k)\hat{z}(k) + y(k) \\ \textbf{end loop} \end{cases}$$

This recursion is precisely a discrete-step Kalman filter. Lemmas 1 and 2 result in:

LEMMA 3: The operator $\mathcal{M}^{-1}$ has the following anti-causal-memoryless-causal operator factorization

$$\mathcal{M}^{-1} = (I - H\psi K)^* D^{-1}(I - H\psi K) \tag{13}$$

Application of lemma 3 to the bias-free robot equations of motion given by eq. (7) immediately yields the following $O(n)$ forward dynamics algorithm:

Algorithm FD

$$T' = T - H\phi[M\phi^* a + b + Bf(0)] \quad (14)$$

$$\ddot{\theta} = (I - H\psi K)^* D^{-1}(I - H\psi K)T' \quad (15)$$

Eq. (14) represents an $O(n)$ Newton-Euler recursion to remove the bias torques. Eq. (15) leads to the following $O(n)$ recursive forward dynamics algorithm:

$$\begin{cases} \hat{z}(0); \quad T'(0) = 0 \\ \textbf{for k = 1} \cdots \textbf{n} \\ \quad \hat{z}(k) = \psi(k, k-1)\hat{z}(k-1) \\ \quad\quad + K(k, k-1)T'(k-1) \\ \quad \varepsilon(k) = T'(k) - H(k)\hat{z}(k) \\ \quad \nu(k) = D^{-1}(k)\varepsilon(k) \\ \textbf{end loop} \end{cases}$$

$$\begin{cases} \lambda(n+1) = 0 \\ \textbf{for k = n} \cdots \textbf{1} \\ \quad \lambda(k) = \psi^*(k+1, k)\lambda(k+1) + H^*(k)\nu(k) \\ \quad \ddot{\theta}(k) = \nu(k) - K^*(k+1)\lambda(k+1, k) \\ \textbf{end loop} \end{cases}$$

It can be shown that the forward dynamics algorithm given by eqs. (14) and (15) is equivalent to that of Featherstone (1983) but is derived by vastly different means. Similarly, it can be shown that $P(k)$ defined earlier is an articulated body inertia as defined by Featherstone (1983) but discovered independently, and in a much different context, in Rodriguez (1987a).

In addition to these operator factorizations, there exist many operator identities relating the various operator factors. This greatly enhances the ability to obtain a number of important results. For instance, it is shown in Rodriguez and Kreutz (1990b) how these identities can be used to obtain a variety of $O(n)$ forward dynamics algorithms, all of them significantly different. Indeed, among these algorithms are ones that do not require the separate computation of T' as in eq. (14) and directly take care of the terms involving a, b, and $f(0)$ in the recursive implementation of eq. (15). It is seen that the algorithm given by eqs. (14) and (15) is but one in a whole class of such algorithms available from an application of the spatial operator algebra. Furthermore, extensions to closed-chain systems made up of several arms rigidly grasping a common rigid object can be found in Rodriguez and Kreutz (1990a) and in Rodriguez (1989b). The case of loose grasp of nonrigid articulated objects is found in Jain et al. (1990b). General closed-graph rigid multibody systems are analyzed in Rodriguez, Jain, and Kreutz (1989).

5. Applications of Spatial Operator Identities

Previously, we have referred to the availability of identities relating elements of the spatial operator algebra. In Rodriguez and Kreutz (1990b), many such relationships are derived. In this section, we will focus on the application of one such identity as representative of how these identities can be used to perform high-level manipulations that result in new algorithms useful in dynamic analysis and control. The identity of interest is:

LEMMA 4:

$$(I - H\psi K)H\phi = H\psi \quad (16)$$

Proof: See appendix.

□

5.1. Application 1: Tip Force Correction Accelerations

From eq. (2) it is evident that

$$\ddot{\theta} = \ddot{\theta}_f + \Delta\ddot{\theta}$$

where

$$\ddot{\theta}_f = \mathcal{M}^{-1}(T - \mathcal{C}), \quad \text{and} \quad \Delta\ddot{\theta} = -\mathcal{M}^{-1}J^* f(0)$$

can be determined from the forward dynamics algorithm eqs. (14) and (15). Our first application of lemma 4 is to find a simple relationship between tip contact forces and the resulting joint accelerations, $\Delta\ddot{\theta}$, caused solely by such tip forces. From eq. (1) and eq. (13),

$$\Delta\ddot{\theta} = -(I - H\psi K)^* D^{-1}(I - H\psi K)H\phi Bf(0). \quad (17)$$

Application of lemma 4 then results in

$$\Delta\ddot{\theta} = -(I - H\psi K)^* D^{-1} H\psi Bf(0). \quad (18)$$

Eq. (18) is significantly simpler than eq. (17). It shows how the effect of the tip force propagates from the tip to the base of a manipulator, producing link accelerations that then propagate from the base to the tip. The algorithmic equivalent to eq. (18) is given by

$$\begin{cases} \hat{z}(1) = \psi(1, 0)f(0) \\ \textbf{for k = 1} \cdots \textbf{n} \\ \quad \hat{z}(k) = \psi(k, k-1)\hat{z}(k-1) \\ \quad \nu(k) = -D^{-1}(k)H(k)\hat{z}(k) \\ \textbf{end loop} \end{cases}$$

$$\begin{cases} \lambda(n+1) = 0 \\ \textbf{for k = n} \cdots \textbf{1} \\ \quad \lambda(k) = \psi^*(k+1, k)\lambda(k+1) + H^*(k)\nu(k) \\ \quad \Delta\ddot{\theta}(k) = \nu(k) - K^*(k+1, k)\lambda(k+1) \\ \textbf{end loop} \end{cases}$$

5.2. *Application 2: Effective Manipulator Inertia Reflected to the Tip*

The next application of lemma 4 will be to produce an $O(n)$ recursive algorithm [see Rodriguez and Kreutz (1990a) for computing the operational space inertia matrix Λ of Khatib (1985)]. Knowledge of Λ, together with the operational space Coriolis, centrifugal, and gravity terms, enables the use of operational space control—a form of feedback linearizing control described in Khatib (1985). The ability to obtain the operational space dynamics recursively avoids the need to have explicit analytic expressions, which can be quite complex. Although we will only discuss the recursive construction of the operational space inertia matrix Λ, the entire operational space dynamics can be computed via $O(n)$ recursions using the techniques of the spatial operator algebra, allowing for recursive implementation of operational space control.

If the dynamics of an n-link manipulator are reflected to the tip locations, the resulting manipulator inertia has the form

$$\Lambda = (J\mathcal{M}^{-1}J^*)^{-1}.$$

For a manipulator whose work space is R^6, the inversion of the 6×6 operator $J\mathcal{M}^{-1}J^*$ entails a constant cost that is independent of the number of manipulator links. The real work is to obtain an efficient algorithm for the construction of $\Omega(0) \overset{\Delta}{=} J\mathcal{M}^{-1}J^*$. Eq. (1) and eq. (13) reveal that

$$\begin{aligned} \Omega(0) &= J\mathcal{M}^{-1}J^* \\ &= B^*\phi^*H^*(I - H\psi K)^*D^{-1}(I - H\psi K)H\phi B \end{aligned} \tag{19}$$

Application of lemma 4 to eq. (19) immediately results in

$$\Omega(0) \overset{\Delta}{=} J\mathcal{M}^{-1}J^* = B^*\psi^*H^*D^{-1}H\psi B \tag{20}$$

It is quite straightforward (Rodriguez and Kreutz 1990a; Rodriguez, Jain, and Kreutz 1989) to show that the following $O(n)$ anticausal base-to-tip recursive algorithm is equivalent to eq. (20):

$$\begin{cases} \Omega(n + 1) = 0 \\ \textbf{for k = n} \cdots \textbf{1} \\ \quad \Omega(k) = \psi^*(k + 1, k)\Omega(k + 1)\psi(k + 1, k) \\ \qquad\qquad + H^*(k)D^{-1}(k)H(k) \\ \textbf{end loop} \end{cases}$$

$$\Omega(0) = \phi^*(1, 0)\Omega(1)\phi(1, 0)$$

5.3. *Application 3: Closed-Chain Forward Dynamics*

Figure 1*A* represents a closed chain of rigid bodies connected by revolute joints that are all actuated. Figure 1*A* can be viewed as a graph whose nodes are links and whose edges are joints. A spanning tree can be found for this graph, which is equivalent to cutting the chain at some point—say, point c—of Figure 1*A*. The root of this tree is indicated by the arrow.

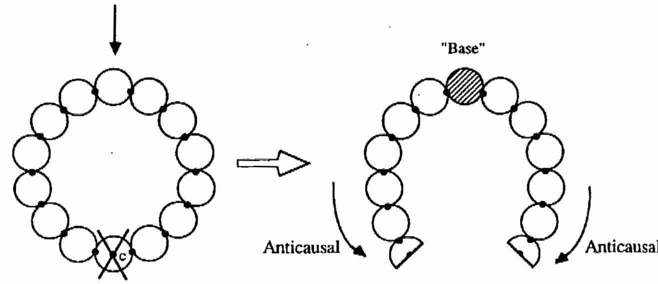

Fig. 1. Closed chain system.

Imagine that the chain is physically cut at c, and designate the root link to be the "Base." This results in Figure 1*B*. For simplicity, assume that the base is immobile. This assumption results in no real loss of generality (Rodriguez and Kreutz 1990a). Cutting the chain has resulted in arms 1 and 2 with n_1 and n_2 links, respectively. We can now assign the causal/anticausal directions to each arm. (Note that this assignment propagated back to Figure 1*A* corresponds to the existence of a directed graph.)

The fact that the tips of arms 1 and 2 are always constrained to remain in contact corresponds to the boundary conditions

$$f_2(0) = -f_1(0) \equiv f(0) \tag{21}$$

$$\alpha_1(0) = \alpha_2(0) \tag{22}$$

With eq. (21), the dynamic behavior of arms 1 and 2 is given by

$$\begin{aligned} \mathcal{M}_1\ddot{\theta}_1 + \mathcal{C}_1 &= T_1 + J_1^*f(0), \\ \mathcal{M}_2\ddot{\theta} + \mathcal{C}_2 &= T_2 - J_2^*f(0) \end{aligned} \tag{23}$$

subject to eq. (22). Looking first at arm 1,

$$\begin{aligned} \ddot{\theta}_1 &= \mathcal{M}_1^{-1}(T_1 - \mathcal{C}_1) + \mathcal{M}_1^{-1}J_1^*f(0) \\ &= \ddot{\theta}_{1f} + \Delta\ddot{\theta}_1 \end{aligned}$$

where

$$\begin{aligned} \ddot{\theta}_{1f} &= \mathcal{M}_1^{-1}(T_1 - \mathcal{C}_1) \\ &= [I - H\psi K]^*D^{-1}[I - H\psi K](T - \mathcal{C}) \end{aligned} \tag{24}$$

and

$$\begin{aligned} \Delta\ddot{\theta}_1 &= \mathcal{M}_1^{-1}J_1^*f(0) \\ &= [I - H\psi K]^*D^{-1}H\psi Bf(0). \end{aligned}$$

Note that $\ddot{\theta}_{1f}$ is the "free" joint acceleration (i.e., the joint acceleration that would exist if the tip were unconstrained), while $\Delta\ddot{\theta}_1$ is the correction joint acceleration for arm 1 resulting from the presence of the tip constraint force $f(0)$. Although $\ddot{\theta}_{1f}$ can be obtained using the recursive $O(n_1)$ single-arm forward dynamics algorithms, so can $\Delta\ddot{\theta}_1$ once $f(0)$ is determined. The same story holds for arm 2 also.

Because $V_1(0) = J_1\dot{\theta}_1$,

$$\alpha_1(0) = \dot{V}_1(0) = J_1\ddot{\theta}_1 + \dot{J}_1\dot{\theta}_1. \tag{25}$$

It then follows from eqs. (24) and (25) that

$$\begin{aligned}\alpha_1(0) &= \alpha_{1f}(0) + J_1\,\Delta\ddot{\theta}_1,\\ &\quad \text{where } \alpha_{1f}(0) = J_1\ddot{\theta}_{1f} + \dot{J}_1\Delta\dot{\theta}_1\\ &= \alpha_{1f}(0) + \Lambda_1^{-1}f(0),\\ &\quad \text{where } \Lambda_1^{-1} \triangleq J_1\mathcal{M}_1^{-1}J_1^*.\end{aligned}$$

Similarly,

$$\alpha_2(0) = \alpha_{2f}(0) - \Lambda_2^{-1}f(0) \quad \text{where } \Lambda_2^{-1} \triangleq J_2\mathcal{M}_2^{-1}J_2^*.$$

Then, from the boundary condition constraint in eq. (22)

$$f(0) = \Lambda_c[\alpha_{2f}(0) - \alpha_{1f}(0)] \qquad \text{where } \Lambda_c^{-1} \equiv \Lambda_1^{-1} + \Lambda_2^{-1} \tag{26}$$

As discussed previously, Λ_1^{-1} and Λ_2^{-1} can be found via $O(n_1)$ and $O(n_2)$ recursive algorithms, respectively. Noting that the inversion of $\Lambda_c^{-1} \in R^{6\times6}$ involves a flat cost independent of n_1 and n_2, we see that we have produced an $O(n_1 + n_2)$ recursive algorithm for finding the forward dynamics of the system of Figure 1*A*. Λ_c is the effective inertia of the closed-chain system reflected to point c.

For additional applications of the spatial operator algebra similar to those of this section, see Rodriguez and Scheid (1987) and Rodriguez and Kreutz (1990a). In Rodriguez and Scheid (1987) an operator expression for $(J^*J)^{-1}$ is obtained for nonredundant arms that is used in a recursive solution to the manipulator inverse kinematics problem. Rodriguez and Kreutz (1990a) contains an extensive listing of additional operator identities. Also shown is a method to easily find the effective inertia matrix for a system consisting of several arms grasping a commonly held rigid body.

6. Research Applications of the Spatial Operator Algebra

The ability to adequately model rigid bodies in arbitrary configurations and states of contact is important for the development of effective computer-aided design (CAD)–based motion planners. In situations involving remote multiarm robotic servicing of a multibody system (such as a space station), manipulator arms, tools, objects, and the environment will be constantly forming new and changing configurations of interaction. The topology of such configurations will, in general, be quite complex. The special, representative case of several arms rigidly grasping a commonly held rigid body is studied in Rodriguez (1986), Kreutz and Lokshin (1988) and Kreutz and Wen (1988), both from the control and the modeling perspectives. In these references, several alternative representations for the dynamic equations describing this case are derived. An important quantity for understanding the behavior of a closed-chain system is seen to be the effective inertia matrix, which is just the natural generalization of the Khatib operational space inertia matrix for a single serial link arm.

As our closed-chain example has shown, a key step in obtaining the effective inertia matrix is understanding how a new effective inertia is formed when a single arm grasps an object, which may be a simple single rigid body or a complex multibody mechanism. The solution is best obtained not by recomputing the effective inertia for the new arm-object system from scratch, but by including the effect of the object as an incremental change to the solution of the dynamics problem. To add the effect of the object, one first computes the contact forces at the points of contact between the arm and the object. This is achieved by an approach that is analogous to combining two distinct state estimates, each of which has a built-in error with a known "covariance" (i.e., articulated body inertia) (Rodriguez 1989b). This perspective enables the generation of efficient recursive algorithms for computing the effective inertia of a system of several arms grasping a common object that is of complexity $O(n) + O(l)$, when no arm is at a kinematic singularity. More generally, $O(n) + O(l^3)$ algorithms can be developed where n is the total number of links in the system and l is the number of arms grasping the object (Rodriguez and Kreutz 1990a).

For the model of several rigid-link serial arms grasping a common object to be well posed (in the sense that unique system accelerations and unique contact forces result for given applied joint moments), it can be shown that the inverse of the effective inertia must be full rank. This enables the determination of unique contact forces, which in turn are sufficient for computing accelerations. It is important that this full-rank condition be satisfied everywhere in the work space if a dynamic simulation is to be well posed for all possible motions. In Rodriguez, Milman, and Kreutz (1988), it is shown that the property of well-posedness throughout the work space is generic with respect to the base locations of the arms. Thus almost surely, "with probability one," any set of base locations for the arms will result in a closed-chain system that is well posed for simulation purposes. Assuming well-posedness, the techniques of the spatial algebra allow the joint accelerations and contact forces of a multiarm/object-grasp system to be computed from applied joint moments by means of an $O(n) + O(l)$ recursive algorithm.

Most of the multibody results mentioned previously assume that a rigid attachment has been made between objects as they come in contact. Of course, this is a highly limiting assumption that must be relaxed in realistic problem domains. The algorithms described here for the dynamics of manipulators in rigid grasp of a rigid object have been extended in Jain et al. (1990a) to the case where the grasp is loose and the task object is nonrigid and has internal degrees of freedom. The grasp constraints are allowed to be either holonomic or nonholonomic. This includes (1) possibly one-sided contacts, such as line contacts with friction; (2) point contacts with friction; (3) "soft-finger" contacts; and (4) sliding contacts, such as occur in hybrid force/position control.

Notice that the factorization eq. (11) can be interpreted as a change of basis, which results in a "decoupled" (i.e., diagonal) inertia matrix D. This key insight can be used to obtain highly decoupled equations of motion in terms of the "innovations"

$$\varepsilon = D^{-1/2}(I - H\psi K)T$$

and the "residuals"

$$\nu = D^{1/2}(I + H\phi K)^{*}\dot{\theta}.$$

The resulting equations of motion are of the form

$$\dot{\nu} + \mu(\theta, \nu) = \varepsilon.$$

The diagonalized innovations form of the dynamic equations result in a significant simplification of dynamic analysis. Application of Lyapunov stability theory for control design is particularly appropriate when manipulator dynamics are described in this diagonal innovations canonical form and results in new forms of decoupled control. The analysis is simplified as a result of the diagonalization of the kinetic energy term, which is contained in many useful Lyapunov candidate functions.

In Kreutz and Lokshin (1988) and Rodriguez, Milman, and Kreutz (1988), feedback linearizing type control laws for controlling a system of multiple arms grasping a common object are derived. These controllers enable the simultaneous control of configuration, as well as internal forces, either to regulate the contact forces imparted to the held object or for load-balancing among the arms. Via the spatial operator algebra, it is straightforward to obtain $O(n)$ recursively implementable forms of these control laws.

Recently new forms of manipulator control laws have been derived via the use of Lyapunov stability theory (Wen and Bayard 1988; Wen et al. 1988). Work is underway to extend these results to the closed-chain case (see Wen and Kreutz (1988)). A straightforward application of the recursive Newton-Euler algorithm will not work because of the need to distinguish in a complex manner the placing of desired and actual joint velocities into the bilinear Coriolis/centrifugal terms. For this reason, exact analytic expressions of these controllers have been required to date. Recently, however, we have applied the techniques of the spatial operator algebra to obtain $O(n)$ recursive implementations of these new forms of control laws.

The use of the spatial operator algebra for dynamic modeling and algorithms for arbitrary tree topology multibody systems can be found in Rodriguez (1987b); for arbitrary graph topology rigid multibody systems, in Rodriguez, Jain, and Kreutz (1989); and for flexible manipulators, in Rodriguez (1990b). Other application areas include motion and force planning for manipulators (Rodriguez 1989c); algorithms for manipulators with gear-driven joints (Jain and Rodriguez 1990a); computation of robot linearized robot dynamics models (Jain and Rodriguez 1990b); operational space control (Kreutz et al. 1990); and as a unifying framework for multibody dynamics (Jain 1990).

One of the most important features of the spatial operator algebra is that it is easy (Rodriguez and Kreutz 1990b) to develop hierarchical software for implementation of recursive algorithms. The complexity of the algorithms are not visible to the user, because only spatial operator expressions are required to do the computer programming. This simplifies software prototyping without increasing computational complexity. It also makes simulation programs arm-independent, bcause the operator statements and the computer program architecture do not vary in going from one arm to another arm.

7. Conclusions

A new spatial operator algebra for describing the kinematic and dynamic behavior of multibody systems has been presented. The algebra makes it easy to see the relationship between abstract expressions and recursive algorithms that propagate spatial quantities from link-to-link. One consequence of the operator algebra is that the equivalence between the Lagrangian and Newton-Euler formulations of dynamics is made transparent. Abstract dynamic equations of motion, such as those arising from a Lagrangian analysis, can be reinterpreted as equivalent operator-formulated equations.

Important elements of the spatial operator algebra were presented, in particular those that arise from natural factorizations of critical kinematic and dynamic quantities. These factorizations allow one to manipulate equations of motion in previously unknown ways. This is particularly true given the existence of important identities and inversions that relate the spatial operators. A key result is the operator factorization and inversion of the manipulator mass matrix given by lemma 1 and lemma 3. Various applications of the spatial algebra to kinematics, dynamics, and control were presented, including the development of a recursive forward dynamics algorithm that essentially comes for free once the key step of obtaining the innovations factorization eq. (9) is carried out.

The factorizations made possible by the spatial operator algebra are model based, in the sense that the physical model of the manipulator itself is used to conduct every computational step. Hence every computational step has a physical interpretation. Numeric errors are easy to detect, because the results of any given computation can easily be checked against physical intuition. These model-based factorizations are quite distinct from the more traditional factorizations, such as Cholesky decomposition, that are rooted in numeric analysis and for which there is not typically a one-to-one physical interpretation for every computational step.

The potential payoff of the spatial algebra in managing the complexity associated with multibody systems is large. For example, compare the abstract simplicity of the development of the forward dynamics algorithm in this article with those developed by other means that often require extensive notation and development. In section 6, we touched on some of the other areas where the spatial operator algebra is being applied. This algebra can greatly aid in the generation of computer programs that model the behavior of the dynamic world by the use of a suitable hierarchy of abstraction.

Appendix

We first establish the following identity.

LEMMA 5:

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

Proof: It is easy to verify that $\phi^{-1} = (I - \mathcal{E}_\phi)$ and $\psi^{-1} = (I - \mathcal{E}_\psi)$. Then,

$$\psi^{-1} = I - \mathcal{E}_\psi = (I - \mathcal{E}_\phi) + KH = \phi^{-1} + KH \square$$

Proof of lemma 1: From eq. (8) it follows that

$$M = P - \mathcal{E}_\psi P \mathcal{E}_\psi^*.$$

However, it is easy to verify that $\bar{\tau} P \bar{\tau}^* = \bar{\tau} P$, where $\bar{\tau} 1 \triangleq I - GH$. And so, using eq. (10) and the fact that $\tilde{\phi} \triangleq \phi - I = \phi \mathcal{E}_\phi$,

$$M = P - \mathcal{E}_\psi P \mathcal{E}_\phi^* P - \mathcal{E}_\phi P \mathcal{E}_\phi^* + KDK^*$$

$$\Rightarrow \phi M \phi^* = P + \tilde{\phi} P + P \tilde{\phi}^* + \phi KDK^* \phi^*$$

$$\Rightarrow \mathcal{M}$$

$$= H \phi M \phi^* H^*$$

$$= H[P + \tilde{\phi} P + P \tilde{\phi}^* + \phi KDK^* \phi^*]H^*$$

$$= D + H\phi KD + DK^*\phi^* H^* + H\phi KDK^* \phi^* H^*$$

$$= [I + H\phi K]D[I + H\phi K]^*$$

$\square$

Proof of lemma 2: Using a standard matrix identity followed by lemma 5, we have that

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

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

$\square$

Proof of lemma 4:

$$(I - H\psi K)H\phi = H(I - \psi KH)\phi$$

From lemma 5 it follows that $(I - \psi KH) = \psi\phi^{-1}$, and using this, the result follows.

$\square$

Acknowledgment

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

References

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

Craig, J. J. 1986. *Introduction to Robotics*. Reading, Mass.: Addison-Wesley.

Featherstone, R. 1983. The calculation of robot dynamics using articulated-body inertias. *Int. J. Robot. Res.* 2(1):13–30.

Jain, A. 1990. Unified formulation of dynamics for serial rigid multibody systems. *J. Guidance Control Dynam.* In press.

Jain, A., Kreutz, K., and Rodriguez, G. 1990a (Vancouver, June). Recursive dynamics of multiarm robotic systems in loose grasp of articulated task objects. *3rd International Symposium on Robotics and Manufacturing*. ASME Rev.

Jain, A., Kreutz, K., and Rodriguez, G. 1990b (Honolulu, December). Multi-arm grasp and manipulation of objects with internal degrees of freedom. *IEEE Conference on Decision and Control*. IEEE Press.

Jain, A., and Rodriguez, G. 1990a (Honolulu, December). Recursive dynamics of geared robotic manipulators. *IEEE Conference on Decision and Control*. IEEE Press.

Jain, A., and Rodriguez, G. 1990b (Los Angeles, November). Recursive linearization of manipulator dynamics models. *IEEE Conference on Systems, Man and Cybernetics*. IEEE Press.

Khatib, O. 1985. The operational space formulation in the analysis, design, and control of manipulators. *3rd Int. Symp. Robotics Research.*

Kreutz, K., and Lokshin, A. 1988 (Atlanta, June). Load balancing and closed-chain multiple arm control. *1988 American Control Conference*. IEEE Press.

Kreutz, K., and Wen, J. 1988 (Atlanta, June). Attitude control of an object commonly held by multiple robot arms: A Lyapunov approach. *American Control Conference*. IEEE Press.

Kreutz, K., Jain, A., and Rodriguez, G., 1990 (Vancouver, June). Recursive formulation of operational space control. *3rd International Symposium on Robotics and Manufacturing*. ASME Press.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. 1980. On-line computational scheme for mechanical manipulators. *ASME J. Dynam. Sys. Measurement Control* 102(2):69–76.

Rodriguez, G. 1986 (Albuquerque, November). Filtering and smoothing approach to dual robot arm dynamics. *Proc. 1st Int. Symp. on Robotics and Manufacturing*. ASME Press.

Rodriguez, G. 1987a. Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics. *IEEE J. Robot. Automat.* 3(6):624–639.

Rodriguez, G. 1987b (Blacksburg, Va., June). Kalman filtering, smoothing, and topological tree dynamics. *VPI/SU AIAA Symposium on Control of Large Structures.* AIAA Press.

Rodriguez, G. 1989b. Recursive forward dynamics for multiple robot arms moving a common task object. *IEEE Trans. Robot. Automat.* 5(4):510–521.

Rodriguez, G. 1989c (Philadelphia, November). Statistical mechanics models for motion and force planning. *SPIE Conf. on Intelligent Control and Adaptive Systems*. SPIE Proceedings Series.

Rodriguez, G. 1990a. Random field estimation approach to robot dynamics. *IEEE Trans. Sys. Man Cybernet.* 20(5):1081–1093.

Rodriguez, G. 1990b (Cincinnati, May). Spatial operator approach to flexible multibody manipulator inverse and forward dynamics. *IEEE International Conference on Robotics and Automation*. IEEE Press.

Rodriguez, G., Jain, A., and Kreutz, K. 1989 (Oxnard, Calif., August). Spatial operator algebra framework for multibody system dynamics. *Journal of Astro. Sci.*, in press. *Proceedings of the 3rd Annual Conf. on Aerospace Computational Control.*

Rodriguez, G., and Kreutz, K. 1990a. Closed-chain forward dynamics by manipulator mass matrix factorization and inversion. *IEEE Trans. Robot. Automat.* In press.

Rodriguez, G., and Kreutz, K. 1990b. Recursive mass matrix factorization and inversion: An operator approach to manipulator forward dynamics. *IEEE Trans. Robot. Automat.* In press.

Rodriguez, G., Milman, M., and Kreutz, K. 1988 (Detroit, August). Dynamics and coordination of multiple robot arms. *3rd Int. Conf. on CAD/CAM and Robotics.* ASME Press.

Rodriguez, G., and Scheid, R. E. 1987 (Monterey, Calif., August). Recursive inverse kinematics for robot arms via Kalman filtering and Bryson-Frazier smoothing. *AIAA Guidance, Navigation, and Control Conference.* AIAA Press.

Roman, P. 1975. *Modern Mathematics for Physicists and Other Outsiders*. New York: Pergamon Press.

Rudin, W. 1973. *Functional Analysis*. New York: McGraw Hill.

Silver, W. M. 1982. On the equivalence of Lagrangian and Newton-Euler dynamics for manipulators. *Int. J. Robot. Res.* 1(2):118–128.

Wen, J., and Bayard, D. 1988. A new class of control laws for robotic manipulators. *Int. J. Control* 47.

Wen, J., and Kreutz, K. 1988 (Albuquerque). A control perspective for multiple arm systems. *2nd Int. Symp. Robotics and Manufacturing*. ASME Press.

Wen, J., Kreutz, K., and Bayard, D. 1988 (Atlanta). A new class of energy based control laws for revolute arms. *American Control Conference*. IEEE Press.