

A SPATIAL OPERATOR ALGEBRA FOR MANIPULATOR MODELING AND CONTROL

G. Rodriguez, K. Kreutz, and A. Jain
 Jet Propulsion Laboratory/California Institute of Technology
 4800 Oak Grove Drive, MS 198-330, Pasadena, CA 91109

Abstract

A recently developed spatial operator algebra, useful for modeling, control and trajectory design of manipulators 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 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. This paper discusses the analytical formulation of the operator algebra, as well as its implementation in the Ada programming language.

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 [1-10]. 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 analysis of robot arms given in [1-10] has shown that certain linear operators are always present in the equations describing the dynamical and kinematical behavior of such arms. As a class, these operators are called "spatial operators" since they show how forces, velocities, and accelerations propagate through space from one rigid body to the next. Not only do such operators have obvious physical interpretations, but they are 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 exactly the dynamical equations of motion which arise from a Lagrangian analysis. The fact that the operators have equivalent recursive algorithms is a generalization of the well-known equivalence between the Lagrangian and recursive Newton-Euler approaches to manipulator dynamics [17]. The operator-based formulation of robot dynamics leads to a complete integration of these two approaches, so that analytical expressions can be shown to almost always have implicit, and obvious, recursive equivalents which are straightforward to mechanize.

The essential ingredients of the operator algebra are the operations of addition and multiplication [11, 12]. There is also an "adjoint," or "*", operator which can operate on elements of the spatial algebra. If a spatial operator A is "causal," in a sense to be described below, then its adjoint A^* is "anticausal." 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 [1,3,6,8].

That the equations of multibody dynamics can be completely described by an algebra of spatial operators is certainly of mathematical interest. However, the significance of this result goes beyond the mathematics and is useful in a very practical sense. The spatial 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 trivial to perform and requires only a simple mental exercise. Although, when applied to the dynamical analysis of an N link manipulator the algebra typically leads to $O(N)$ recursive algorithms, numerical efficiency is not the main motivation for its development. What the algebra primarily offers is a powerful mathematical framework that because of its simplicity is believed to be superior for addressing advanced control issues.

Another characteristic of the spatial algebra is that it is closed, in the sense that it completely describes the dynamics of multiple rigid bodies. There are no situations (at least none have been found to date) in which it is necessary to go outside the algebra in order to solve rigid multibody dynamics problems. This implies that the user can reliably use the algebra knowing that the language of discourse can cover all problems that can be modeled as a “world” of multiple rigid bodies dynamically interacting. The algebra therefore puts a mathematical wrapping around this world. It provides a self-contained framework to formulate, analyze, and understand higher-level modeling and control issues.

To show the power and use of the spatial operator algebra, several applications of the algebra to robotics will be presented: 1) An operator representation of the manipulator Jacobian matrix; 2) The robot dynamical equations formulated in terms of the spatial algebra, showing the equivalence between the recursive Newton-Euler formulations to robot dynamics in a far more transparent way than [17]; 3) The operator factorization and inversion of the manipulator mass matrix which immediately results in $O(N)$ recursive forward dynamics algorithms; 4) The joint accelerations of a manipulator due to a tip contact force; 5) The recursive computation of the equivalent mass matrix as seen at the tip of a manipulator (the operational space mass matrix of Khatib [13]); 6) Recursive forward dynamics of a closed chain system. Finally, we will discuss in general terms additional applications and current research involving the application of the spatial operator algebra.

2. The Jacobian Operator

After defining a link spatial velocity to be $V(k) = \text{col}[\omega(k), v(k)]$, the recursion which describes the relationship between joint angle rates, $\dot{\theta} = \text{col}[\dot{\theta}(1), \dots, \dot{\theta}(N)]$, and link velocities, $V = \text{col}[V(1), \dots, V(N)]$, is [8]

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

LOOP $k = N, \dots, 1$;

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

END LOOP;

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

$H^T(k) = h(k)$ where $h(k) \in R^6$ is the unit vector in the direction of the k^{th} joint axis, and $\phi^T(k+1, k)$ is the Jacobian which transforms velocities across a rigid link. This recursion represents a base-to-tip recursion which shows how link velocities propagate outward to the tip, point “0” on link 1, from the base “link $N+1$,” assuming that the base has zero velocity. Note that the link numbering convention used here, and in [1-10], increases from the tip to the base unlike the numbering convention described in most robotics textbooks [14].

Summation of the above recursion leads to

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

where the facts that $\phi(i, i) = \mathcal{I}$ 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” $H^* = \text{diag}[H^T(1), \dots, H^T(N)]$, $B^* = [\phi^T(1, 0), 0, \dots, 0]$ and

$$\phi \triangleq \begin{pmatrix} \mathcal{I} & 0 & 0 & \dots & 0 \\ \phi(2, 1) & \mathcal{I} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \phi(N, 1) & \phi(N, 2) & \dots & \dots & \mathcal{I} \end{pmatrix}$$

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

$$J = B^* \phi^* H^* \quad (2.1)$$

The Jacobian operator in (1.1) is seen to be the product of three operators B^* , ϕ^* and H^* . The operator H^* , being block diagonal, is called “memoryless” while 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,” making ϕ^* 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: $H^* \dot{\theta}$ results in relative spatial velocities between the links along the joint axes: The action of ϕ^* then anticausally propagates these relative velocities from the base to the tip to form the link spatial velocities $V = \text{col}[V(1), \dots, V(N)]$; B^* then projects out $V(1)$ from V and propagates it to the tip forming $V(0)$.

The well-known dual relationship to $V(0) = J \dot{\theta}$ is $T = J^* f(0) = H \phi B f(0)$, where $f(0) = \text{col}[N(0), F(0)]$ is a spatial force which represents the tip interaction with the environment [14]. The action of J^* on $f(0)$ is as follows: B takes $f(0)$ to $\text{col}[f(1), 0, \dots, 0]$. The effect of ϕ is to propagate $f(1)$ causally from link 1 to the base forming the spatial forces acting at each link represented by $f = \text{col}[f(1), \dots, f(N)]$. Finally, H projects each component of f , $f(k)$, onto joint axis $H^T(k) = h(k)$ to obtain the joint moments $T = \text{col}[T(1), \dots, T(N)]$.

The key points to note here are that J and J^* have operator factorizations which have immediate physical interpretations and obvious recursive algorithmic equivalents. Working with the factorized version of J , one can manipulate expressions involving J in novel 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 [5] and [15].

3. An Operator Formulated Robot Dynamics

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

$$\mathcal{M} \ddot{\theta} + \mathcal{C} + J^* f(0) = T \quad (3.1)$$

\mathcal{C} denotes “bias” torques due to the velocity dependent coriolis and centrifugal effects. Equation (3.1) is precisely the form that arises from a Lagrangian analysis of manipulator dynamics. Equation (3.1) has an operator interpretation which arises from the following spatial operator factorizations of \mathcal{M} , \mathcal{C} , and J^*

$$\mathcal{M} = H \phi M \phi^* H^* \quad (3.2a)$$

$$\mathcal{C} = H \phi (M \phi^* a + b) \quad (3.2b)$$

$$J^* = H \phi B \quad (3.2c)$$

These factorizations are derived in [1,3,6,8]. The quantity

$$M = \text{diag}[M(1), \dots, M(N)]$$

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

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

where $I(k)$ 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 (3.2b), $a = \text{col}[a(1), \dots, a(N)]$ and $b = \text{col}[b(1), \dots, b(N)]$ are quadratic functions of the link spatial velocities. The operators H , ϕ , and B were described in the previous section.

When (3.1) is given an operator interpretation via (3.2), it is immediately apparent that (3.1) is functionally identical to the Newton-Euler recursions given by [8,14,16]

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

LOOP $k = N, \dots, 1$;

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

END LOOP;

$$f(0) = f_{ext}$$

LOOP $k = 1, \dots, N$;

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

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

END LOOP;

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

It is now evident that the equivalence between the Lagrangian and recursive Newton-Euler formulations of manipulator dynamics [17] has been made trivial. Furthermore, the factorizations given by (3.2) allow us to manipulate the dynamical 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 mathematical analysis one can interpret expressions in a deeply physical way or immediately produce an equivalent recursive algorithm.

The true power of the spatial algebra applied to manipulator dynamics will become clearer in the following sections. It will be shown that an important alternative factorization to (3.2a) exists which results in new causal, memoryless, anticausal operators with corresponding equivalent recursions. Also, we will discuss the existence of powerful operator identities which allow one to manipulate kinematical and dynamical equations in ways which 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 (3.2a), the well known fact that \mathcal{M} is symmetric positive definite can be easily shown. 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 (3.2a) 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 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 [1,3,6,8] and are partially summarized in this section.

First, we present an important alternative factorization to (3.2a). To this end, we define

$$S \triangleq \begin{pmatrix} 0 & & & & 0 \\ \mathcal{I} & 0 & & & 0 \\ 0 & \mathcal{I} & 0 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \mathcal{I} & 0 \end{pmatrix} \quad \text{and} \quad \Phi \triangleq \phi S = \begin{pmatrix} 0 & & & & 0 \\ \phi(2,2) & 0 & & & 0 \\ \phi(3,2) & \phi(3,3) & 0 & & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi(N,2) & \phi(N,3) & \phi(N,4) & \dots & \phi(N,N) & 0 \end{pmatrix}$$

Also define

$$\Delta_\phi \triangleq \text{diag}[\phi(2,1), \dots, \phi(N+1,N)]$$

and

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

where $P = \text{diag}[P(1), \dots, P(N)]$. The diagonal elements $P(k)$ are obtained by the following causal discrete-step Riccati equation

$$P(1) = M(1)$$

LOOP $k = 2, \dots, N$;

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

END LOOP;

where

$$\psi(k, k-1) \triangleq \phi(k, k-1) - L(k-1)H(k-1) \quad (4.1)$$

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

In an analogous fashion to the definitions of the operators ϕ and Δ_ϕ we define the operators ψ and Δ_ψ with the component operators $\psi(k, k-1)$ being given by Eq. (4.1), $\psi(k, k) \triangleq \mathcal{I}$, and

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

for $i \geq j$. Note that the $\psi(i, j)$'s form a semigroup.

With these definitions, we have that

$$\Delta_\psi = \Delta_\phi(I - GH)$$

Lemma 4.1: An alternative factorization of $\mathcal{M} = H\phi M\phi^*H^*$ is

$$\mathcal{M} = (\mathcal{I} + H\Phi L)D(\mathcal{I} + H\Phi L)^* \quad (4.2)$$

where $\mathcal{I} + H\Phi L$ is causal (lower triangular), and D is memoryless, diagonal and invertible.

Proof: See Appendix.

The factorization (4.2) 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 $\mathcal{I} + H\Phi L$ operating on an innovations process $\epsilon = \text{diag}[\epsilon(1), \dots, \epsilon(N)]$ which can be taken to be an independent Gaussian sequence. The action of $\mathcal{I} + H\Phi L$ on ϵ ,

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

is equivalent to a causal tip-to-base recursion. The importance of the innovations operator $\mathcal{I} + H\Phi L$ is that it is trivially and causally invertible and that its inverse is precisely a discrete-step Kalman filter viewed as a whitening filter. To demonstrate this, define

$$\Psi \triangleq \psi S$$

Lemma 4.2: The causal (lower triangular) operators $\mathcal{I} + H\Phi L$ and $\mathcal{I} - H\Psi L$ are mutual causal inverses of each other

$$(\mathcal{I} + H\Phi L)^{-1} = \mathcal{I} - H\Psi L \quad (4.3)$$

Proof: See Appendix.

The relationship $\epsilon = (\mathcal{I} + H\Phi L)^{-1}y = (\mathcal{I} - H\Psi L)y$ is equivalent to a causal tip-to-base recursion. This recursion is precisely a discrete-step Kalman filter. Lemmas 4.1 and 4.2 result in

Theorem 4.1: The operator \mathcal{M}^{-1} has the following anticausal-memoryless-causal operator factorization

$$\mathcal{M}^{-1} = (\mathcal{I} - H\Psi L)^* D^{-1} (\mathcal{I} - H\Psi L) \quad (4.4)$$

Application of Theorem 4.1 to the bias-free robot equations of motion given by Equation (3.3) immediately yields the following $O(N)$ forward dynamics algorithm

Algorithm FD.

$$T' = T - H\phi[M\phi^*a + b + Bf(0)] \quad (4.5a)$$

$$\ddot{\theta} = (\mathcal{I} - H\Psi L)^* D^{-1} (\mathcal{I} - H\Psi L) T' \quad (4.5b)$$

Equation (4.5a) represents an $O(N)$ Newton-Euler recursion to remove the bias torques. Equation (4.5b) leads to the following $O(N)$ recursive algorithm

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

LOOP $k = 1, \dots, N$;

$$\hat{z}(k) = \psi(k, k-1)\hat{z}(k-1) + L(k-1)T'(k-1)$$

$$\epsilon(k) = T'(k) - H(k)\hat{z}(k)$$

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

END LOOP;

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

LOOP $k = N, \dots, 1$;

$$\lambda(k) = \psi^T(k+1, k)\lambda(k+1) + H^T(k)\nu(k)$$

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

END LOOP;

It can be shown that the forward dynamics algorithm given by Equations (4.5) is equivalent to that of [18], but derived by vastly different means. Similarly, it can be shown that $P(k)$ defined above is an articulated body inertia as defined by [18], but discovered independently, and in a much different context, in [1].

In addition to the operator factorizations at our disposal, there exist a multitude of operator identities relating the various operator factors. This greatly enhances the ability to obtain any number of important results. For instance, it is shown in [8] how these identities can be used to obtain a variety of $O(N)$ forward dynamics algorithms, all of them significantly different. It is seen that the algorithm given by Equations (4.5) above is but one in a whole class of such algorithms available from an application of the spatial operator algebra. Furthermore, in [8] it is shown how these algorithms can be easily extended to the closed-chain system made up of several arms grasping a common rigid object.

5. Applications of Spatial Operator Identities

Above, we have referred to the availability of identities relating elements of the spatial operator algebra. In [8], 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 which result in novel algorithms useful in dynamical analysis and control.

The identity of interest is

Identity 5.1:

$$(\mathcal{I} - H\Psi L)H\phi = H\psi \quad (5.1)$$

Proof: See Appendix.

The action of ψ on a composite spatial quantity $Z = \psi Y$ is equivalent to a causal tip-to-base recursion.

Application 1: Tip Force Correction Accelerations

From Eq. (3.1) it is evident that

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

where

$$\ddot{\theta}_f = \mathcal{M}^{-1}(T - C) = \mathcal{M}^{-1}T'$$

can be determined from the forward dynamics algorithm (4.5). Our first application of Identity (5.1) is to find a simple relationship between tip contact forces and the resulting joint accelerations, $\Delta\ddot{\theta}$, due solely to such tip forces. From Eqs. (3.1) and (3.2), we have $\mathcal{M}\Delta\ddot{\theta} = -J^*f(0)$ or

$$H\phi M\phi^* H^* \Delta\ddot{\theta} = -H\phi Bf(0)$$

With Equation (4.4) this becomes

$$\Delta\ddot{\theta} = -(\mathcal{I} - H\Psi L)^* D^{-1}(\mathcal{I} - H\Psi L)H\phi Bf(0) \quad (5.2)$$

Application of Identity (5.1) then results in

$$\Delta\ddot{\theta} = -(\mathcal{I} - H\Psi L)^* D^{-1}H\psi Bf(0) \quad (5.3)$$

Eq. (5.3) is significantly simpler than (5.2). It shows how the effect of the tip force propagates from the tip to the base of a manipulator, producing link accelerations which then propagate from the base to the tip.

Application 2: Composite Body Inertia Reflected to the Manipulator Tip

The next application of Identity (5.1) will be to produce an $O(N)$ recursive algorithm for computing the Operational Space mass matrix Λ of Khatib [13]. 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 [13]. The ability to obtain the Operational Space dynamics recursively avoids the need to have explicit analytical expressions which can be quite complex. Although we will only discuss the recursive construction of the Operational Space mass 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 composite body inertia has the form

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

For a manipulator whose workspace is R^6 , the inversion of the 6×6 operator $J\mathcal{M}^{-1}J^*$ entails a constant cost which is independent of the number of manipulator links. The real work is to obtain an efficient $O(N)$ algorithm for the construction of $\Omega(0) = J\mathcal{M}^{-1}J^*$. Equations (1.1) and (4.4) reveal that

$$\Omega(0) = J\mathcal{M}^{-1}J^* = B^*\phi^*H^*(\mathcal{I} - H\Psi L)^* D^{-1}(\mathcal{I} - H\Psi L)H\phi B \quad (5.4)$$

Application of Identity (5.1) to Eq. (5.4) immediately results in

$$\Omega(0) = J\mathcal{M}^{-1}J^* = B^*\psi^*H^*D^{-1}H\psi B \quad (5.5)$$

It is quite straightforward [8] to show that the following $O(N)$ anticausal base-to-tip recursive algorithm is equivalent to (5.5)

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

LOOP $k = N, \dots, 1$;

$$\Omega(k) = \psi^T(k+1, k)\Omega(k+1)\psi(k+1, k) + H^T(k)D^{-1}(k)H(k)$$

END LOOP;

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

Application 3: Closed Chain Forward Dynamics

Figure 1a represents a closed chain of rigid bodies connected by revolute joints which are all actuated. Figure 1a 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 1a. The root of this tree is indicated by the arrow.

Imagine that the chain is physically cut at c and designate the root link to be the "Base." This results in Figure 1b. For simplicity, assume that the base is immobile. This assumption results in no real loss of generality – see, e.g., ref. [8]. 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 Fig. 1a corresponds to the existence of a directed graph associated with Fig. 1a.)

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

$$f_2(0) = -f_1(0) \equiv f(0) \quad (5.6a)$$

$$\alpha_1(0) = \alpha_2(0) \quad (5.6b)$$

With (5.6a), the dynamical behavior of arms 1 and 2 is given by

$$\mathcal{M}_1\ddot{\theta}_1 + \mathcal{C}_1 = T_1 + J_1^* f(0) \quad , \quad \mathcal{M}_2\ddot{\theta}_2 + \mathcal{C}_2 = T_2 - J_2^* f(0) \quad (5.7)$$

subject to (5.6b). Note from $V(0) = J\dot{\theta}$ and $\alpha(0) = \dot{V}(0)$ that

$$\alpha_1(0) = J_1\dot{\theta}_1 + \dot{J}_1\theta_1 \quad , \quad \alpha_2(0) = J_2\dot{\theta}_2 + \dot{J}_2\theta_2 \quad (5.8)$$

From (5.7)

$$\dot{\theta}_1 = \ddot{\theta}_{1f} + \Delta\ddot{\theta}_1 \quad , \quad \dot{\theta}_2 = \ddot{\theta}_{2f} + \Delta\ddot{\theta}_2 \quad (5.9)$$

where

$$\ddot{\theta}_1 = \mathcal{M}_1^{-1}(T_1 - \mathcal{C}_1) \quad , \quad \ddot{\theta}_2 = \mathcal{M}_2^{-1}(T_2 - \mathcal{C}_2) \quad (5.10)$$

and

$$\Delta\ddot{\theta}_1 = \mathcal{M}_1^{-1}J_1^* f(0) \quad , \quad \Delta\ddot{\theta}_2 = \mathcal{M}_2^{-1}J_2^* f(0) \quad (5.11)$$

If $f(0)$ can be determined, $\ddot{\theta}_1$ and $\ddot{\theta}_2$ can be found from (5.9)-(5.10) via the recursive algorithms presented earlier.

To find $f(0)$, define

$$\alpha_{1f}(0) = J_1\ddot{\theta}_{1f} + \dot{J}_1\dot{\theta}_1 \quad , \quad \alpha_{2f}(0) = J_2\ddot{\theta}_{2f} + \dot{J}_2\dot{\theta}_2 \quad , \quad \Delta\alpha_f(0) = \alpha_{1f}(0) - \alpha_{2f}(0) \quad (5.12)$$

These quantities can be computed via $O(N_1)$ and $O(N_2)$ recursive algorithms [8]. Eqns. (5.8) and (5.12) yield

$$\begin{aligned} \alpha_1(0) &= \alpha_{1f}(0) + J_1\Delta\ddot{\theta}_1 \\ \alpha_2(0) &= \alpha_{2f}(0) + J_2\Delta\ddot{\theta}_2 \end{aligned}$$

which with the boundary condition (5.6b) gives

$$\Delta\alpha_f(0) = J_1\Delta\ddot{\theta}_1 - J_2\Delta\ddot{\theta}_2 \quad . \quad (5.13)$$

From (5.11), (5.13) becomes

$$\Delta\alpha_f(0) = (\Lambda_1^{-1} + \Lambda_2^{-1})f(0) \quad (5.14)$$

where

$$\Lambda_1^{-1} = J_1\mathcal{M}_1^{-1}J_1^* \quad , \quad \Lambda_2^{-1} = J_2\mathcal{M}_2^{-1}J_2^* \quad (5.15)$$

Thus

$$f(0) = \Lambda_c\Delta\alpha_f(0) \quad (5.16a)$$

$$\Lambda_c^{-1} \equiv \Lambda_1^{-1} + \Lambda_2^{-1} \quad (5.17a)$$

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 $\Lambda_c^{-1} \in R^{6 \times 6}$ 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 1a. Λ_c is the composite body inertia of the closed chain system reflected to point c .

The spatial algebra perspective enables the generation of efficient recursive algorithms for computing the composite body inertia of a system of several arms grasping a common object which is of complexity $O(N) + O(\ell)$, when no arm is at a kinematical singularity, or, more generally, $O(N) + O(\ell^3)$, where N is the total number of links in the system and ℓ is the number of arms grasping the object.

For additional applications of the spatial operator algebra similar to those of this section, see for example [5,8]. In [5] an operator expression for $(J^*J)^{-1}$ is obtained for nonredundant arms which is used in a recursive scheme for solving the manipulator inverse kinematics problem. In [8], many additional examples may be found along with an extensive listing of operator identities. For instance, in [8] it is shown how one can easily find the composite body inertia matrix for a system consisting of several arms grasping a commonly held rigid body.

6. Conclusions

A powerful new spatial operator algebra for describing the kinematical and dynamical behavior of multi-body systems has been presented. Abstract dynamical equations of motion, such as arise from a Lagrangian analysis, can be reinterpreted as equivalent operator formulated equations. From this perspective, the distinction between abstract expressions, the interlink physical relationships of spatial quantities (velocities, accelerations, and forces), and recursive algorithms which propagate spatial quantities from link-to-link entirely vanishes. One consequence of the operator algebra is that the equivalence between the Lagrangian and Newton-Euler formulations of dynamics is trivial and transparent.

Important elements of the spatial operator algebra were presented, in particular those which arise from natural factorizations of critical kinematical and dynamical quantities. These factorizations allow one to manipulate equations of motion in hitherto unknown ways, greatly increasing our powers of analysis. This is particularly true given the existence of important identities and inversions which relate the spatial operators. A key result is the operator factorization and inversion of the manipulator composite body inertia given by Fact 4.1 and Theorem 4.1.

Various applications of the spatial algebra to kinematics, dynamics, and control were presented, including the development of a recursive forward dynamics algorithm, which essentially comes for free once the key step of obtaining the innovations factorization (4.1) is seen.

The potential payoff of the spatial algebra in terms of providing a framework which can manage the complexity associated with multibody systems is immense. For example, compare the abstract simplicity of the development of the forward dynamics algorithm in this paper with those developed by other means which often require a morass of notation and development. In Sec. 5, we touched only lightly on some of the current areas where the spatial operator algebra is being applied. We believe that this algebra can provide a complete framework for describing multibody systems. This will greatly aid in the ultimate generation of "smart" programs which can reason about the behavior of the dynamical world by the use of a suitable hierarchy of abstraction.

7. 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.

8. References

- [1] G. Rodriguez, "Kalman Filtering, Smoothing, and Recursive Robot Arm Dynamics," JPL Publication 86-48, Dec. 1986.
- [2] G. Rodriguez, "Filtering and Smoothing Approach to Dual Robot Arm Dynamics," Proc. 1st Int. Symp. on Robotics and Manufacturing, Albuquerque, NM, Nov. 1986.
- [3] G. Rodriguez, "Spatially Random Models, Estimation Theory, and Robot Arm Dynamics," IEEE Symp. on Intelligent Control, Philadelphia, PA, Jan. 1987.
- [4] G. Rodriguez, "Kalman Filtering, Smoothing, and Topological Tree Dynamics," VPI/SU Symp. on Control of Large Structure, Blacksburg, VA. June 1987.
- [5] G. Rodriguez and R. E. Scheid, Jr., "Recursive Inverse Kinematics for Robot Arms via Kalman Filtering and Bryson-Frazier Smoothing," AIAA Guidance, Navigation, and Control Conference, Monterey, CA, Aug. 1987.
- [6] G. Rodriguez, "Kalman Filtering, Smoothing, and Recursive Robot Arm Forward Dynamics," IEEE J. Robotics and Automation, Vol. RA-3, Dec. 1987.
- [7] G. Rodriguez, "Recursive Forward Dynamics for Multiple Robot Arms Moving a Common Task Object," JPL Publication 88-6, Feb. 1988.
- [8] G. Rodriguez and K. Kreutz, "Recursive Mass Matrix Factorization and Inversion: An Operator Approach to Open- and Closed-Chain Multibody Dynamics," JPL Publication 88-11, March 1988.
- [9] K. Kreutz and A. Lokshin, "Load Balancing and Closed-Chain Multiple Arm Control," 1988 American Control Conference, Atlanta, GA, June 1988.
- [10] G. Rodriguez, M. Milman and K. Kreutz, "Dynamics and Coordination of Multiple Robot Arms," 3rd Int. Conf. on CAD/CAM and Robotics, Detroit, MI, Aug. 1988.
- [11] P. Roman, Modern Mathematics for Physicists and Other Outsiders, Pergamon Press, New York, 1975.
- [12] W. Rudin, Functional Analysis, McGraw-Hill, New York, 1973.
- [13] O. Khatib, "The Operational Space Formulation in the Analysis, Design, and Control of Manipulators," 3rd Int. Symp. Robotics Research, Paris, 1985.
- [14] J. Craig, Introduction to Robotics, Addison-Wesley, Reading, Mass., 1986.
- [15] K. Kreutz, "Ada Packages Supporting the Dual Arm Robot Programming Library," JPL Engr. Memo. 347-88-245 (internal document), June 1988.
- [16] J. Y. S. Luh, M. Walker, and R. Paul, "On-line Computational Scheme for Mechanical Manipulators," J. Dyn. Sys. Meas. and Control, Vol. 120, 1980.
- [17] W. Silver, "On the Equivalence of the Lagrangian and Newton-Euler Dynamics for Manipulators," Int. J. Robotics Research, Vol. 1, 1982.
- [18] R. Featherstone, "The Calculation of Robot Dynamics Using Articulated- Body Inertias," Int. J. Robotics Research, Vol. 2, 1983.

Appendix

Prior to proving Lemma 4.1, Lemma 4.2 and Identity 5.1, we define

$$\tilde{\phi} \triangleq \phi - \mathcal{I} \quad \text{and} \quad \tilde{\psi} \triangleq \psi - \mathcal{I}$$

and establish the following identities.

Lemma A1:

$$(a) \quad \psi^{-1} - \phi^{-1} = SLH \quad (A.1)$$

$$(b) \quad \begin{aligned} \phi\psi^{-1} &= I + \Phi LH \\ \psi^{-1}\phi &= I + SLH\phi \end{aligned} \quad (A.2)$$

$$(c) \quad \tilde{\psi}PH^* = 0 \quad (A.3)$$

$$(d) \quad \psi M \psi^* = P + \tilde{\psi}P + P\tilde{\psi}^* \quad (A.4)$$

$$(e) \quad H\psi M \psi^* H^* = D \quad (A.5)$$

Proof:

(a) It is easy to verify that

$$\tilde{\phi} = \Phi\Delta_\phi \quad \text{and} \quad \tilde{\psi} = \Psi\Delta_\psi$$

It then follows that

$$\phi^{-1} = I - S\Delta_\phi \quad \text{and} \quad \psi^{-1} = I - S\Delta_\psi \quad (\text{A.6})$$

Thus, $\psi^{-1} - \phi^{-1} = S(\Delta_\phi - \Delta_\psi)$. However, we have seen from the definition of Δ_ψ that

$$\Delta_\psi = \Delta_\phi(I - GH) \quad (\text{A.7})$$

and the result follows immediately.

(b) These two identities are obtained by respectively pre- and post-multiplying (A.1) by ϕ .
(c) It follows from the definition of G that $(I - GH)PH^* = 0$. Thus, using (A.7)

$$\tilde{\psi}PH^* = \Psi\Delta_\psi PH^* = \Psi\Delta_\phi(I - GH)PH^* = 0$$

(d) From the definition of P it follows that P satisfies the following equation:

$$M = P - (S\Delta_\psi)P(S\Delta_\psi)^*$$

which using (A.6) and some simple manipulations yields the result.

(e) We have that,

$$\begin{aligned} H\psi M\psi^*H^* &= H(P + \tilde{\psi}P + P\tilde{\psi}^*)H^*, \quad \text{using (A.4)} \\ &= HPH^*, \quad \text{using (A.3)} \\ &= D \end{aligned}$$

Proof of Lemma 4.1:

$$\begin{aligned} \mathcal{M} &\triangleq H\phi M\phi^*H^* \\ &= H(\phi\psi^{-1})\psi M\psi^*(\phi\psi^{-1})^*H^* \\ &= (I + H\Phi L)H\psi M\psi^*H^*(I + H\Phi L)^*, \quad \text{using (A.2)} \\ &= (I + H\Phi L)D(I + H\Phi L)^*, \quad \text{using (A.5)} \end{aligned}$$

Proof of Lemma 4.2:

$$\begin{aligned} (I + H\Phi L)^{-1} &= (I + H\phi SL)^{-1} \\ &= I - H\phi(I + SLH\phi)^{-1}SL, \quad \text{using a standard matrix identity} \\ &= I - H\phi(\psi^{-1}\phi)^{-1}SL, \quad \text{using (A.2)} \\ &= I - H\psi SL = I - H\Psi L \end{aligned}$$

Proof of Identity 5.1:

Premultiplying (A.1) by ψ we see that $\psi\phi^{-1} = I - \Psi LH$. Hence,

$$\begin{aligned} (I - H\Psi L)H\phi &= H(I - \Psi LH)\phi \\ &= H(\psi\phi^{-1})\phi = H\psi \end{aligned}$$

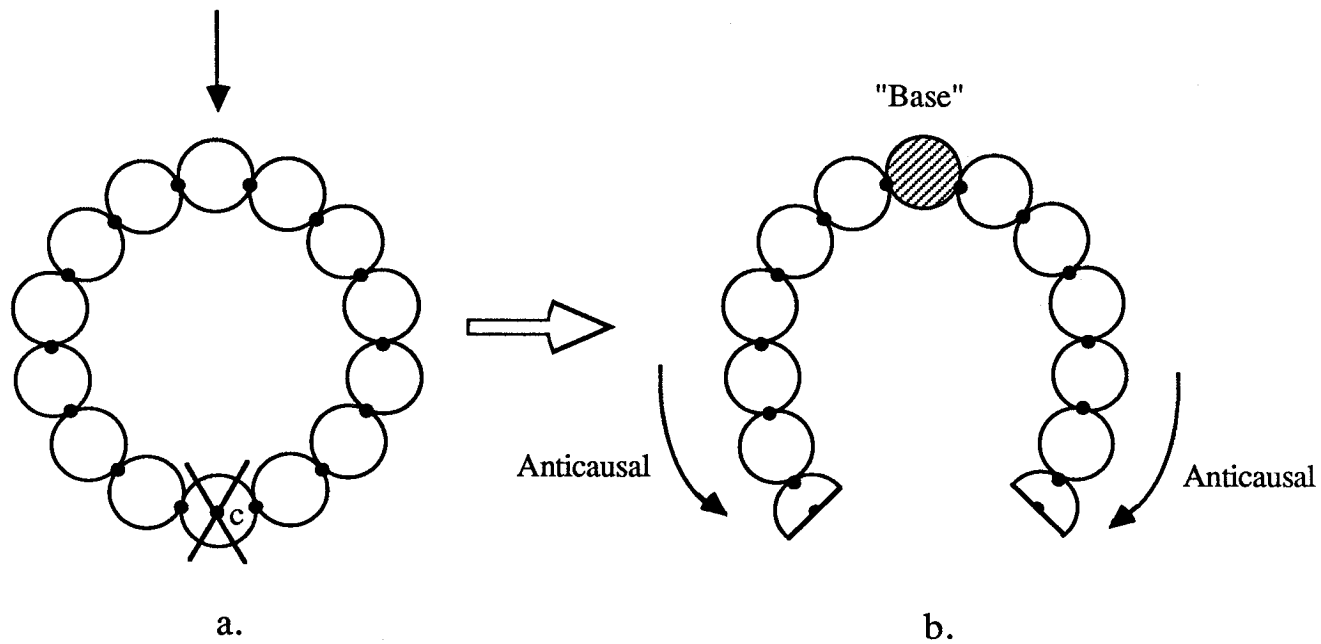


Figure 1: Closed Chain System