Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Computational Robot Dynamics Using Spatial Operators

Abhi Jain & Guillermo Rodriguez
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

Abstract

This paper tells the story of spatial operators in robot dynamics, emphasizing their physical interpretation, while avoiding lengthy mathematical derivations. The spatial operators are rooted in the function space approach to estimation theory developed in the decades that followed the introduction of the Kalman filter. In the mid 1980's, the authors, who were familiar with the function space approach, recognized the analogy between Kalman filtering and robot dynamics, and began to use this approach on a wide range of multi-body systems of increasing complexity. This paper reviews the spatial operator approach to robot dynamics, and outlines current applications to the modeling, simulation and control of space robot dynamics and large molecular structures.

1 Foundations

The pivotal developments in the function space approach to estimation theory are summarized below.

Kalman ('60)	State Space Filter
Bryson ('62)	State Space Smoother
Kailath ('70)	Innovations Approach
Gohberg/Krein ('70)	Factor Positive Ops.
Kailath ('74)	Factor Covariances
Balakrishnan ('76)	Function Space

Kalman introduced the notion of a state space, and a recursive filter [1] that computes the best estimate of the state from possibly noisy past measurements. The optimal Bryson [2] smoother computes the best state estimate using both past and future data. Although several authors seemed to have arrived at similar results at approximately the same time, Kailath [3, 4] was most likely the first to recognize many new techniques. He introduced the "innovations" approach, which when specialized to state space systems was a more advanced way to derive optimal linear estimators such as the Kalman filter. He also recognized the value to estimation theory of powerful mathematical techniques (Gohberg and Krein) to factor positive operators into a product of two closely related integral operators with triangular kernels. The function space approach reached maturity in the work of Balakrishnan [5], who introduced the elegant methods of Hilbert space. At the end of this period, we knew how to easily solve very complicated linear filtering problems using linear integral operators, operator factorization methods, and triangular (Volterra) factors. In the mid 1980's, the authors recognized [6–8] that the equations of mechanical systems had an almost perfect analogy to those of state space linear systems. Discovery of this analogy allowed the use in mechanics of very advanced methods and computational architectures (Kalman, Bryson, Riccati, etc.) that had emerged from estimation theory.

2 Key Spatial Operators

Table 1 summarizes the most fundamental spatial operators [9]. The operator $\phi(k, k-1)$ converts a spatial force at joint $k-1$ of a given link k, into a corresponding spatial force at the next k^{th} joint. Its transpose $\phi^*(k, k-1)$ transforms spatial velocities and accelerations in the opposite direction. Both transformations are rigid, as the joint $k-1$ involved in the transformation is kept locked. Note that these are the only operators in the table that involve a SINGLE rigid body. The rest of the operators in the table are GLOBAL, in the sense that they involve the entire multibody system.

2.1 Rigid Force Shift Operator

The operator $\mathcal{E}_\phi$ is a shift operator whose elements are all zero, except along its lower subdiagonal as shown below, without loss of generality, for the special case of a system with 3 bodies

$$\mathcal{E}_\phi = \begin{pmatrix} 0 & 0 & 0 \\ \phi(2,1) & 0 & 0 \\ 0 & \phi(3,2) & 0 \end{pmatrix} \tag{2.1}$$

This operator shifts forces in a direction from the tip of the system toward its base. The transpose $\mathcal{E}_\phi^*$ shifts velocities and accelerations in the opposite direction from the base to the tip.

Operator	Interpretation
$\phi(k,k-1)$	Move force to next link
$\phi^*(k,k-1)$	Move vel. to prior link
$\mathcal{E}_\phi$	Rigid shift of forces
H	From state to joint space
H^*	From joint to state space
M	Rigid link inertia
P	Articulated inertia
$D = HPH^*$	Articulated joint inertia
G	Kalman gain
$K = \mathcal{E}_\phi G$	Shifted Kalman gain
$\overline{\tau} = (I - GH)$	Joint articulation
$\mathcal{E}_\psi$	Art. shift to next link
$\phi = (I - \mathcal{E}_\phi)^{-1}$	Rigid recursion
$\psi = (I - \mathcal{E}_\psi)^{-1}$	Articulated recursion
B	Force Pickoff Operator
B^*	Vel. Pickoff Operator

Table 1: Key spatial operators

2.2 Rigid Recursion Operator

The rigid recursion operator ϕ is defined as

$$\phi = (I - \mathcal{E}_\phi)^{-1} = \begin{pmatrix} I & 0 & 0 \\ \phi(2,1) & I & 0 \\ \phi(3,1) & \phi(3,2) & I \end{pmatrix} \tag{2.2}$$

and can be used to compute spatial recursions in the generally inward direction, toward the base of the system, starting from its tip. The recursions are rigid in the sense that the joints are kept locked.

2.3 From State to Joint Space

The operator H projects the 6-dimensional spatial forces at the joint into generalized force components along the joint axes.

$$H = \begin{pmatrix} H(1) & 0 & 0 \\ 0 & H(2) & 0 \\ 0 & 0 & H(3) \end{pmatrix} \tag{2.3}$$

Its transpose H^* converts or "expands" the scalar rotational rates along the joint axes into 6-dimensional relative spatial velocities across the joints.

2.4 Articulated Body Inertia

To our knowledge, the articulated body inertia P was labeled as such for the first time in the work of [10]. From the independent view of estimation theory, the articulated inertia is the solution to the Riccati equation

$$P = \overline{\tau}\mathcal{E}_\phi P \mathcal{E}_\phi + M \tag{2.4}$$

where M is a composite block-diagonal matrix whose diagonal blocks are the spatial inertias of the various rigid links forming the system. This spatial operator Riccati equation is equivalent to a spatial recursion that goes in an inward direction from the tip to the base of the system, and which sequentially generates the diagonal elements of the Riccati operator. The operator $P = diag[P(1), P(2), P(3)]$ is a diagonal operator defined in terms of the articulated body inertias $P(k)$ at the 3 joints forming the 3-body sample system.

2.5 Kalman Gain Operator

The Kalman gain operator $G = PH^*D^{-1}$ is computed from the articulated body inertia and appears as a key element in the recursive Kalman filtering algorithms. Its primary function is to compute the joint articulation operator $\overline{\tau}$ used in the Riccati equation to remove the articulable component of the inertia, thereby rendering the resulting body outboard of this joint as an articulated body.

2.6 Articulated Shift Operator

The operator $\mathcal{E}_\psi$ is similar to $\mathcal{E}_\phi$ except that it produces "articulated" shifts instead of "rigid" shifts. The related operator ψ is a lower-triangular matrix representing an inward spatial Kalman filtering recursion, i. e.,

$$\psi = (I - \mathcal{E}_\psi)^{-1} = \begin{pmatrix} I & 0 & 0 \\ \psi(2,1) & I & 0 \\ \psi(3,1) & \psi(3,2) & I \end{pmatrix} \tag{2.5}$$

This global recursion operator ψ propagates forces in a generally inward direction. In crossing the k^{th} joint, the articulation operator $\overline{\tau}(k)$ is applied, and this is the reason for using the term "articulated" to describe this recursion. Its transpose ψ^* is an upper-triangular matrix used to propagate velocities in an outward direction across articulated bodies. The articulated recursion ψ takes into account articulation at the joints, whereas the rigid recursion ϕ does not.

2.7 Force and Velocity Pick-Off

The pick-off operator B converts spatial forces defined at any given point $\mathcal{C}$ into a global force defined over the entire system. For example, if the force $f(C)$ is defined at the tip $\mathcal{C}$ of the system, the global force $f = [f(1), f(2), f(3)]^*$ that "stacks" the forces $f(k)$ at the 3 joints can be computed as $f = \phi B f(C)$ where ϕ is the rigid recursion operator defined in Eq. (2.2) and $B = [\phi^*(1, \mathcal{C}, 0, 0]^*$. The transpose operation $B^* = [\phi^*(1, C), 0, 0]$ maps the stacked velocity vector $V =$

$[V(1), V(2), V(3)]^*$ into the spatial velocity $V(\mathcal{C})$ at the point $\mathcal{C}$.

3 Trees: All Goes Through

While the definition of the fundamental spatial operators has been described in terms generally applicable to serial chain systems with a base and a tip, the notation remains the same for tree configurations in which each individual body can be either flexible or rigid [11]. This is one of the central advantages of using spatial operators: they allow the analyst to view high-level mathematical patterns defined by the various spatial operators and their relationships to each other. These high-level relationships are unchanged in going from serial chain systems to tree configuration formed either by rigid or flexible links.

3.1 Gather and Scatter Operations

To do tree-configurations, the construction of the ϕ operator must be modified slightly. For example, consider joint 3 in a system of four bodies, and assume that both the previous joints 1 and 2 are connected to body 3. In this situation, there is a "gathering" operation that computes the spatial force $f(3)$ at the given joint 3 as the appropriate combination of forces $f(1)$ and $f(2)$ at the previous joints 1 and 2. This is achieved by

$$f(3) = \phi(3,1)f(1) + \phi(3,2)f(2) \qquad (3.6)$$

In the opposite direction, as needed to compute velocities, the velocities $V(1)$ and $V(2)$ at joint 1 and 2 are computed by

$$V(1) = \phi^*(3,1)V(3); \qquad V(2) = \phi^*(3,2)V(3) \quad (3.7)$$

which can be viewed as a "scatter" operation from joint 3 into the two joints 1 and 2. The gather/scatter operations can easily be embedded into the operator ϕ for rigid recursions and into the operator ψ for articulated recursions.

3.2 Flexible Links

The high-level spatial operator notation (ϕ and ϕ^* for example) is unaltered in going from rigid links to flexible links. What changes is the way in which these operators are synthesized. To assemble an operator for a flexible link requires internal coordinates, either modal or physical, for the link flexibility. The combined effect of the internal link coordinates is then computed at one of the joints attached to the body by means of a "gather" or a "scatter" operation.

4 Analysis using Spatial Operators

Spatial operators achieve a very high level of abstraction. This allows one to see clearly, as there is no more mathematical clutter. There are no subscripts, superscripts, summations and their indices, and numerical labels for each of the individual bodies in the system. The analyst simply sees a spatial operator represented by a single symbol, ϕ or ψ for example, which has embedded in it more detailed information that is invisible to the user. Mathematical analysis can be done using the spatial operators only. Because of the sparsity of symbols, these operator manipulations and expressions are easier to understand. Simultaneously, each operator equation has an immediately obvious spatially recursive interpretation that can be obtained either by simple visual inspection or by using a computer program to do this conversion. There are many problems that have been addressed in this manner using operator notation, as outlined in the following subsections.

4.1 Recursive Jacobian and Its Inverse

We start with perhaps the most fundamental of operations characteristic of the entire multibody system, that of the Jacobian operator

$$J = B^*\phi^*H^* \qquad (4.8)$$

that maps the joint velocities $\dot{\theta}$ that live in what we refer to as the "joint space" of the system to the tip velocity. The H^* operator converts joint space into state space. It converts the scalar rotational rate at each joint into a corresponding 6-dimensional spatial velocity at the same joint. The transpose ϕ^* of the rigid recursion operator ϕ produces sequentially the CUMULATIVE spatial velocity at each joint due to rotation at all prior joints in the recursion operator. Finally, the velocity pick-off operator B^* picks off the last spatial velocity computed by ϕ^* and moves it over to the tip point $\mathcal{C}$.

Using this type of method to compute the effect of the Jacobian recursively is of course equivalent to well-known methods in differential kinematics. However, the spatial operator notation in Eq. (4.8) involves significantly fewer symbols.

Similarly, consider the problem of finding the joint rates $\dot{\theta}_0$ that result in no internal motion when a redundant manipulator (with more than 6 degrees of freedom) is required to achieve a prescribed tip velocity $V(\mathcal{C})$. The joint rates $\dot{\theta}_0$ have minimal norm, and a key quantity is the generalized Jacobian inverse

$$J^{-!} = J^*(JJ^*)^{-1} = H\phi B(B^*QB)^{-1} \qquad (4.9)$$

where $Q = \phi^* H^* H \phi$ is a positive definite spatial operator generated by the base-to-tip recursion

$$Q = \mathcal{E}_\phi^* Q \mathcal{E}_\phi + H^* H \tag{4.10}$$

followed by the tip velocity "pick-off" operation $B^*[\cdot]B$. This results in the matrix B^*QB that needs to be inverted. Once this inversion is done, a tip-to-base rigid recursion is used characterized by the spatial operator $H\phi B$ in Eq. (4.9).

4.2 Mass Matrix Recursive Factorization

The mass matrix has [7] the following Newton-Euler spatial operator factorization

$$\mathcal{M} = H \phi M \phi^* H^* \tag{4.11}$$

where $M = diag[M(1), M(2), M(3)]$ is a block diagonal matrix formed by the individual link spatial masses $M(k)$. This factorization implies, and is implied by, the traditional recursive Newton-Euler algorithm for robot arm inverse dynamics. Furthermore, the diagonal elements of the mass matrix can be computed by the tip-to-base rigid recursion

$$r = \mathcal{E}_\phi r \mathcal{E}_\phi^* + M \tag{4.12}$$

followed by the "sandwich" operation $H[\cdot]H^*$. A similar and closely related rigid recursion can be used to compute the off-diagonal elements of the composite mass matrix.

4.3 Mass Matrix Innovations Factorization

The mass matrix has [7] the alternative factorization

$$\mathcal{M} = (I + H\phi K) D (I + H\phi K)^* \tag{4.13}$$

where the outer factors are mutual transposes of each other. The factor $(I + H\phi K)$ is a square, invertible matrix whose inverse is

$$(I + H\phi K)^{-1} = I - H\psi K \tag{4.14}$$

involving the articulated Kalman filtering recursion ψ and the shifted Kalman gain operator K. This implies that the inverse of the mass matrix is

$$\mathcal{M}^{-1} = (I - H\psi K)^* D^{-1} (I - H\psi K) \tag{4.15}$$

The identity implies, and is implied by, a tip-to-base Kalman filtering operation followed by a base-to-tip Bryson smoothing operation. The spatially recursive algorithm that results has been shown to be equivalent to the articulated inertia forward dynamics algorithm advanced by Featherstone [10]. The identity also results in an explicitly symbolic expression for the inverse of the mass matrix.

4.4 Diagonalized Lagrangian Dynamics

When applied to the kinetic energy, the innovations factorization results in fully diagonalized Lagrangian equations of motion, i. e.,

$$K.E. = \frac{1}{2}\dot{\theta}^* \mathcal{M} \dot{\theta} = \frac{1}{2}\nu^* D \nu \tag{4.16}$$

where $\nu = (I + H\phi K)^* \dot{\theta}$ is the "innovations" process corresponding to Lagrangian quasi-coordinates. It is easy to go back and forth between the quasi-coordinates ν and the physical coordinates $\dot{\theta}$ because they are related by spatial operators that are mutual inverses of each other as in Eq. (4.14). The diagonalized equations that result from the diagonalized kinetic energy are

$$\dot{\nu} + \mathcal{C}(\nu, \theta) = \epsilon \tag{4.17}$$

where the generalized applied forces $\epsilon = (I - H\psi K)T$ are obtained from the physical applied moments T by means of the articulated Kalman filtering operation $(I - H\psi K)$. The Coriolis term can be computed explicitly in terms of spatial operators and spatial recursions already defined.

4.5 Operator Sensitivities

Operator sensitivities are defined [12] as partial derivatives of the spatial operators with respect to infinitesimally small variations at any given joint angle. Such sensitivities are useful in many situations, but in particular in evaluation explicitly the Coriolis term $\mathcal{C}(\nu, \theta)$ in the diagonalized Lagrangian equations of the previous subsection. We have computed operator sensitivities [12] for all of the spatial operators defined to date, as summarized in Table 1. The operator sensitivities have the critically important feature that they can all be computed in terms of the spatial operators themselves. This means that no new operators need to be introduced, as there is only a need to combine in a prescribed ways the existing set of spatial operators.

4.6 Closed-Chain Systems

Closed-chain systems need the new spatial operator

$$\Omega = B^* \psi^* H^* D^{-1} H \psi B \tag{4.18}$$

which can be generated [13] by the base-to-tip articulated recursion

$$\Lambda = \mathcal{E}_\psi^* \Lambda \mathcal{E}_\psi + H^* D^{-1} H \tag{4.19}$$

followed by the "sandwich" operator $B^*[\cdot]B$. The matrix Ω that results is the inverse of the operational space inertia first labeled as such by Khatib [14], as discussed by Kreutz in [13]. This matrix is embedded in the following expression

$$\alpha = (I - H\psi K)^* D^{-\frac{1}{2}} [I - \widehat{b}\widehat{b}^*] D^{-\frac{1}{2}} (I - H\psi K) T \tag{4.20}$$

that solves for the joint accelerations α, given a set of applied moments T. The operator $\widehat{b}$ is defined as

$$\widehat{b} = H\psi B(\Omega)^{-\frac{1}{2}} \tag{4.21}$$

$\widehat{b}$ is a "unit" operator, in that $\widehat{b}^*\widehat{b} = I$. Eq. (4.20) differs from the related equation (4.15) for the inverse of the serial system mass matrix by the insertion of the operator $(I - \widehat{b}\widehat{b}^*)$. This is a projection operator in the sense that its square is the operator itself. A forward dynamics algorithm for closed chains can be summarized as

- Computed biased innovations process by the tip-to-base filtering operator $(I - H\psi K)$.
- Remove this bias by the projection operator $I - \widehat{b}\widehat{b}^*$ which can be mechanized by an base-to-tip articulated smoothing recursion followed by a tip-to-base articulated filtering operation.
- Compute resulting accelerations α by means of an articulated filtering operation.

This illustrates a property of the use of spatial operators:to solve increasingly more complex problems, there may be a need to introduce more advanced spatial operators. However, the number of new operators needed is small, and they formed from the previously defined basic operators. New problems can be solved by combining in well-defined ways the existing spatial operators.

4.7 Under-actuated Systems

Under-actuated systems [15] are those that have fewer actuators than degrees of freedom. They are important to applications in free-flying space robots, hyper-redundant manipulators, and articulated systems with structural flexibility. The structure of the dynamics problems for such systems is a hybrid of forward and inverse dynamics. Spatial operators can easily establish this result, and also develop the corresponding recursive algorithms [15].

The hybrid mixture of forward and inverse dynamics problems can be summarized by the following equation:

$$\begin{pmatrix} T_a \\ \dot{\beta}_p \end{pmatrix} = \begin{pmatrix} \mathcal{S}_{aa} & \mathcal{S}_{ap} \\ -\mathcal{S}_{ap}^* & \mathcal{S}_{pp} \end{pmatrix} \begin{pmatrix} \dot{\beta}_a \\ T_p \end{pmatrix} \tag{4.22}$$

where for simplicity the Coriolis terms typically encountered in dynamics equations have not been shown. The active joint moments, applied at the actuated joints, are denoted by T_a. The passive joint moments, occurring due to friction and other effects at the non-actuated joints, are denoted by T_p. The dynamics problem consists of finding the active joint moments T_a required to achieve desired active joint accelerations $\dot{\beta}_a$. In addition, the resulting accelerations $\dot{\beta}_p$ at the passive joints must also be computed. The matrices $\mathcal{S}_{pp}$, $\mathcal{S}_{ap}$, and $\mathcal{S}_{aa}$ in Eq. (4.22). are all expressible in terms of spatial operators [15]. For example,

$$\mathcal{S}_{pp} = (I - H_p\psi K_p)^* D_p^{-1} (I - H_p\psi K_p) \tag{4.23}$$

which is the "innovations" factorization of the inverse of the strictly "passive" manipulator mass matrix, where only the passive joints are taken into account. The other matrices $\mathcal{S}_{aa}$ and $\mathcal{S}_{ap}$ in Eq. (4.22) have similar expressions. The spatial operator analysis of the dynamics problem in Eq. (4.22) results in the following spatially recursive algorithm. A spatial recursion starts from the tip and proceeds toward the base. At each joint, a check determines if the joint is passive or active. If the joint is active, its acceleration is known and is used to update a residual force. On the other hand, if the joint is passive, its generalized force is known and is used to update the residual force. This recursion continues until the base is reached. Now begins a recursion in the opposite direction. As each new joint is encountered, its joint acceleration is computed if it is a passive joint, or else, its unknown generalized force is computed if it is an active joint. This continues until the tip is reached and all of the joints have been processed. This algorithm is a hybrid of known recursive inverse and forward dynamics algorithms [15].

4.8 Base Invariant, Free-Flying Systems

Space robots have a unique characteristic: they are free to rotate and translate with respect to inertial coordinates, without being constrained to be immobile at their base. For these systems, the arbitrariness in the choice of the basebody defines a symmetry which has been used to develop decoupled forward dynamics algorithms [16, 17]. The algorithms we have investigated [16, 17] process the various dynamical quantities one at a time, in two independent sequences. One of the two sequences starts at one end, and the other starts at the opposite end. The two independent sequences go in opposite directions. Each of the sequences represents a spatially recursive algorithm. For the case of the inverse dynamics problem, the recursions are rigid recursions implemented by the spatial operator ϕ. To solve forward dynamics problems, the articulated recursions in the operator ψ are used. The left-to-right algorithm uses only "past" information, in the sense that its output at any given body in the sequence depends only on dynamical quantities, such as link masses, associated with bodies to the left of the given body that have already been processed. Similarly, the right-to-left algorithm uses only "future" information associated with bodies to the right of the given body. Together, these two algorithms use precisely all of the available information. The algorithms are independent in the sense that their outputs are uncorrelated to each other. Due to this independence, the outputs of the two algorithms can be combined in an optimal sense, using the by now classical result [18, 19] of the optimal combination of two uncorrelated state estimates.

5 Applications

While the spatial operator methods can potentially be applied many types of multibody systems, the authors have focused on applications to spacecraft dynamics simulation, to robotics, and to large molecular structures.

5.1 Spacecraft Dynamics Simulation

Real-time computer simulators to predict accurately the dynamical motion of an actual spacecraft in flight lead to significant reductions in cost during system design, development and testing. They also improve system performance and reliability during flight operations. In the early stages of a flight project, design options and trades can easily be made by computer simulation. In the final stages of design and testing, when the spacecraft may already be built and almost ready to be flown, computer simulations are used for hardware-in-the-loop testing and evaluation. Such hardware-in-the-loop simulations are hybrid systems consisting of both flight hardware (sensors, actuators, etc.) together with software for spacecraft dynamics. During the operational phase, missions use simulations to design and verify spacecraft command sequences prior to in-flight execution.

DARTS (Dynamics Algorithms for Real Time Simulation) is a software package developed by the authors that provides a high-performance, rapid-prototyping tool for end-to-end system design, development and testing. DARTS has been adopted as a standard by a number of space missions including, the Mars Pathfinder mission that landed on Mars in 1997, the Galileo spacecraft now in orbit about Jupiter, the Cassini voyage to Saturn launched in late 1998, the mission Stardust to collect samples from a comet and return to Earth, and advanced studies for a Neptune orbiting spacecraft. In July 1997, DARTS received the NASA Software of the Year Award for its contributions to NASA missions [20].

5.2 Robotics

The authors have use spatial operators to address a number of robotics research problems: efficient inversion of the manipulator mass matrix [6], non-interacting manipulator control [12], control of under-actuated manipulators [15], free-flying robotic systems [17], and dual arm manipulation [21].

5.3 Large Molecular Structures

The DARTS simulation package has been adapted [22, 23] to develop the NEIMO package for large-scale molecular dynamics simulations using internal coordinates in a joint research effort with Caltech. The simulations study the structural and functional relationships of proteins and enzymes; protein folding mechanisms and pathways; new drug design; and design and study of catalysts and polymers. The fundamental technical problem being addressed is the global and local dynamical behavior a complex collection of many atoms joined together by interactive forces. The efficiency of the dynamics algorithms embedded in the spatial operators enable the accurately detailed study of much larger systems than could be studied otherwise.

6 Concluding Remarks

This overview aimed at presenting the spatial operator approach to robot dynamics with a minimal amount of mathematical detail. This is made possible by the very high-level of mathematical abstraction allowed by spatial operators. While the mathematical description is very abstract, each spatial operator has a physical interpretation and a corresponding spatially recursive algorithm. The more detailed recursive algorithms are explained in more detail in the references.

Acknowledgments

The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under contract with the National Aeronautics and Space Administration. Thanks are due to K. Kreutz-Delgado of the University of California at San Diego for his keen insight and contributions in the early stages of the development of the spatial operator approach, and to G. Man of JPL for his key contributions in developing the DSHELL/DARTS

software package and in getting it accepted by NASA missions.

References

[1] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," ASME Trans. J. Basic Engr., vol. D, pp. 35-45, March 1960.

[2] A. Bryson and M. Frazier, "Smoothing for Linear and Nonlinear Dynamical Systems," Proc. Optimum Sys. Synthesis Conf. U.S. Air Force Tech. Rept. ASD-TDR-63-119, Feb. 1963.

[3] T. Kailath, "The Innovations Approach to Detection and Estimation Theory," Proc. IEEE, vol. 58, pp. 680-695, 1970.

[4] T. Kailath, "A View of Three Decades of Linear Filtering Theory," IEEE Trans. Info. Thoery, vol. IT-20, pp. 147-181, 1974.

[5] A. V. Balakrishnan, "Applied Functional Analysis," Springer-Verlag, New York 1976.

[6] G. Rodriguez, "Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics," *IEEE J. Robotics Automat.*, vol. 3, no. 6, pp. 624-639, Dec. 1987.

[7] G. Rodriguez and K. Kreutz-Delgado, "Spatial operator factorization and inversion of the manipulator mass matrix," *IEEE Trans. Robotics Automat.*, vol. 8, no. 1, pp. 65-76, Feb. 1992.

[8] G. Rodriguez, "Random Field Estimation Approach to Robot Dynamics," *IEEE Trans. Syst., Man, and Cybern.*, vol. 20, no. 5, pp. 1081-1093, Sept./Oct. 1990.

[9] G. Rodriguez, K. Kreutz-Delgado, and A. Jain, "A Spatial Operator Algebra for Manipulator Modeling and Control," Int. J. Robotics Res., Aug. 1991.

[10] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. Robotics Res.*, vol. 2, pp. 13-30, Spring 1983.

[11] A. Jain and G. Rodriguez, "Recursive Flexible Multibody System Dynamics Using Spatial Operators," J. Guidance, Control, and Dynamics, vol. 15, pp. 1453-1466, Nov. 1992.

[12] A. Jain and G. Rodriguez, "Diagonalized Lagrangian Robot Dynamics," *IEEE Trans. Robotics Automat.*, vol. 11, no. 4, pp. 571-584, Aug. 1995.

[13] K. Kreutz-Delgado, et al., "Recursive Formulation of Operational Space Control," *Int. J. of Robotics Res.*, Vol. 11, No. 4, Aug. 1992.

[14] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robot. Automation* RA-3: pp. 43-53.

[15] A. Jain and G. Rodriguez, "An Analysis of the Kinematics and Dynamics of Underactuated Manipulators," IEEE Transactions on Robotics and Automation, Vol. 9, No. 4, August 1993.

[16] A. Jain and G. Rodriguez, "Base-Invariant Symmetric Dynamics of Free-Flying Manipulators," *IEEE Trans. Robotics Automat.*, vol. 11, no. 4, pp. 585-597, Aug. 1995.

[17] "Two-Way Recursive Decomposition of the Free-Flying Robot Mass Matrix," Journal of Applied Mathematics and Computer Science, Vol. 7, pp. 101-119, June 1997.

[18] D. Fraser and J. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Trans. Automat. Contr.*, pp. 387-390, Aug. 1969.

[19] A. Gelb, *Applied Optimal Estimation*, Cambridge, MA: M.I.T. Press, 1974.

[20] A. Jain, G. Rodriguez, G. Man, "1997 NASA Software of the Year Award, http://dshell.jpl.nasa.gov/

[21] G. Rodriguez and A. Jain, "Recursive Forward Dynamics for Multiple Robot Arms Moving a Common Task Object, "IEEE Transactions on Robotics and Automation," Vol. 5, No. 4, August 1989.

[22] Jain, A., N. Vaidehi, and G. Rodriguez, "A Fast Recursive Algorithm for Molecular Dynamics Simulation,"Journal of Computational Physics, Academic Press, Vol. 106, pp. 258-268, May 1993.

[23] Mathiowetz, A. M., A. Jain, N. Karasawa, and W. A. Goddard, III, "Protein Simulations Using Techniques Suitable for Very Large Systems," Journal of Proteins: Structure, Function, and Genetics, June 1994.