Kalman Filtering, Smoothing, and Recursive Robot Arm Forward and Inverse Dynamics

GUILLERMO RODRIGUEZ

Abstract—**The inverse and forward dynamics problems for multilink serial manipulators are solved by using recursive techniques from linear filtering and smoothing theory. The pivotal step is to cast the system dynamics and kinematics as a two-point boundary-value problem. Solution of this problem leads to filtering and smoothing techniques similar to the equations of Kalman filtering and Bryson–Frazier fixed time-interval smoothing. The solutions prescribe an inward filtering recursion to compute a sequence of constraint moments and forces followed by an outward recursion to determine a corresponding sequence of angular and linear accelerations. An inward recursion refers to a sequential technique that starts at the tip of the terminal link and proceeds inwardly through all of the links until it reaches the base. Similarly, an outward recursion starts at the base and propagates out toward the tip. The recursive solutions are $O(N)$, in the sense that the number of required computations only grows linearly with the number of links. A technique is provided to compute the relative angular accelerations at all of the joints from the applied external joint moments (and vice versa). It also provides an approach to evaluate recursively the composite multilink system inertia matrix and its inverse. The main contribution is to establish the equivalence between the filtering and smoothing techniques arising in state estimation theory and the methods of recursive robot dynamics. The filtering and smoothing architecture is very easy to understand and implement. This provides for a better understanding of robot dynamics. While the focus is not on exploring computational efficiency, some initial results in that direction are obtained. This is done by comparing performance with other recursive methods for a planar chain example. The analytical foundation is laid for the potential use of filtering and smoothing techniques in robot dynamics and control.**

I. Introduction

The central theme of this paper is to investigate the use of filtering and smoothing techniques in studying robot dynamics. In particular, the paper shows that the recursive difference equations of Kalman filtering [1] and Bryson–Frazier fixed time-interval smoothing [2], arising in the state estimation theory [3] for linear state space systems, can be used to solve the problems of serial manipulator inverse and forward dynamics. A more detailed development of the results of this paper is contained in [4], [5]. The configuration analyzed is that of a joint connected N-link serial manipulator attached to an immobile base. The joints are assumed to be rotational, although extension to configurations with joints allowing translation is simple. The inverse dynamics problem is to find the joint moments to achieve a set of prescribed accelerations. The forward dynamics problem is to determine the joint accelerations resulting from a set of applied joint moments. Typically, inverse dynamics solutions are useful for control design, whereas forward dynamics solutions are useful for system simulation.

The primary motivation for using filtering and smoothing techniques is to provide a better means to formulate, analyze, and understand spatial recursions for robot dynamics. It is known that the joint angle accelerations at any given time depend linearly on the joint moments applied at the same time. One of the initial steps of the paper is to develop a spatially recursive state space model to characterize this linear relationship. Development of this state space model makes it possible to apply many of the ideas and concepts (transition matrix, prediction, filtering, smoothing, etc.) from state estimation theory. These concepts have proven themselves ideally suited to investigate discrete-time systems. They are also very useful to organize the recursive computations required to solve the inverse and forward dynamics problems. The filter and smoother are very easy to understand. Extensive analytical and computational experience exists with this architecture in other application areas. Standardized software also is available that can be used to set up readily the required computations.

The notions of spatial force, acceleration, and inertia [6] are used to develop the state space model and thereby simplify the statement of the recursive equations. A spatial force acting on a link is defined here as a six-dimensional vector whose first three components represent a pure moment and whose last three components represent a force. Both the moment and the force forming the spatial force act on the link with which the spatial force is associated. Similarly, a spatial acceleration is defined to be a six-dimensional vector formed by an angular acceleration and a linear acceleration. For any given link, the spatial inertia is a 6×6 matrix which very compactly embodies the mass and inertia properties of the link about its inner joint (that joint closest to the base). It should be pointed out that there are minor differences between the definitions for spatial force, acceleration, and inertia used here and those of [6].

One of the important steps in the paper is to recognize that the equations of translational and rotational motion (derived from Newton's second law) for each link can be cast as a linear difference equation that allows the spatial force at the inner joint to be computed from the spatial force at the outer joint and the spatial acceleration of the link. The difference equation is very similar to those describing the evolution of the state of a discrete-time state space system [3]. The spatial force plays the role of the state. The spatial interval, defined as the vector from the inner to the outer joint, plays the role of the time

Manuscript received July 9, 1986; revised May 27, 1987. This work was supported by the National Aeronautics and Space Administration.

The author is with the Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

IEEE Log Number 8718073.

0882-4967/87/1200-0624$01.00

interval between discrete time samples. This establishes a means to "propagate" the spatial force inwardly within a link from the outer joint to the inner joint. In addition, since the magnitude of the spatial force is continuous at the joints (due to Newton's third law), a means also exists to propagate the spatial force across a joint at the interface between two adjacent links. Recursive use of these two steps allows a complete link-to-link sequential propagation of the spatial force from the tip to the base. An "output" equation is associated with the above state equation in order to generate the scalar joint moments. This output equation is defined in terms of a 6×1 matrix that projects the six-dimensional spatial force into the scalar moment along the joint axis. It should be stressed that the equation for the spatial forces is a difference equation in space and not in time. There is no time discretization involved, and a fully continuous time evolution is retained.

Similarly, a complementary difference equation is obtained that produces a set of spatial accelerations as its solution and uses the joint angle accelerations as inputs. The spatial accelerations play the role of the costates (or adjoint variables) that are typical in optimal control and estimation problems [3]. This costate equation reflects the kinematic relationship that exists between the spatial (angular and linear) acceleration of a link at its outer joint given the acceleration of its inner joint. The difference equation can therefore be used to propagate outwardly the spatial accelerations within a link. A similarly outward propagation across the joint at the interface between two adjacent links is obtained from the observation that the relative angular acceleration along the joint axis introduces a "jump" discontinuity in the joint–axis angular acceleration component of the costate (spatial acceleration) vector.

When combined, the above state and costate difference equations define a two-point boundary-value problem that very closely resembles those typically encountered as necessary (and at times sufficient) conditions for optimality in optimal control and estimation theory. The boundary conditions in this problem are that the state vanishes at the tip of the manipulator and the costate vanishes at the base. These conditions arise because of the assumptions that the tip is unconstrained and the base is immobile (undergoes no accelerations). By considering slightly different boundary conditions, other types of configurations can be analyzed. This is done without otherwise changing the statement of the two-point boundary-value problem. For example, free–free boundary conditions can be used to study systems in which the base link is not attached to a fixed reference. Closed-chain systems, in which the bodies form a loop, can also be studied by proper selection of the boundary conditions. In all of these configurations, the boundary-value problem is used as a starting point to develop the recursive inverse and forward dynamics solutions.

Consider first the inverse dynamics problem. Its solution is obtained by means of a two-stage process involving 1) an outward recursion from the base link to the tip to obtain a set of costates (spatial accelerations), using the set of prescribed joint angle accelerations and the boundary condition at the manipulator base, and 2) an inward recursion from the tip to the base using the results of the first stage above and producing a set of states (spatial forces) at all of the links and the base. The required joint moments are produced by the output equation that is appended to the state equation. This two-stage process can be performed either numerically or symbolically. If performed numerically, the process is quite similar to those commonly used to solve inverse dynamics problems for serial robotic manipulators [7]. If performed symbolically, the process results in the by now traditional dynamical equations for an N-link manipulator, expressed in terms of an $N \times N$ composite system inertia matrix.

To arrive at the above equations requires that the state and costate equations of the two-point boundary-value problem be solved in terms of the spatial transition matrix and its transpose. Substitution of the solution for the costate into that of the state leads to the desired equations of motion. An interesting byproduct is a method for recursive computation of the inertia matrix itself by means of an inward iteration from the tip to the base. This recursive relationship for the inertia matrix is equivalent to those that describe the propagation of the covariance of the state of a linear discrete-time state space system that is driven by white noise. With this result, the similarities between the statistical estimation theory for discrete-time systems and recursive robot arm dynamics begin to reveal themselves. More similarities become apparent upon investigation of the forward dynamics problem as outlined below.

The forward dynamics solution also uses the two-point boundary-value problem as a starting point. The key idea is to seek a solution of the form $x(k) = z(k) + P(k)\lambda(k)$, where $x(k)$ and $\lambda(k)$ denote the state and costate for link k. The symbol $z(k)$ denotes a six-dimensional vector which turns out to play the role of the predicted state estimate whose propagation is described by the Kalman filter. The applied joint moments play the role of the measurements. Similarly, $P(k)$ is a 6×6 matrix, with the units of spatial inertia, which satisfies a difference equation analogous to the Riccati equation of the discrete Kalman filter. The above relationship between states and costates is central to the "sweep method" referred to in [3]. Use of this in the two-point boundary-value problem leads to a two-stage computation consisting of 1) inward filtering to obtain a sequence of state (spatial force) estimates and a corresponding "innovations" process defined at each joint as the difference between the actual and the predicted joint moment; and 2) outward smoothing in which the innovations process resulting from the first stage is used to generate a sequence of costates (spatial accelerations) and the desired joint angle accelerations.

The filtering recursions have a predictor/corrector architecture corresponding to that of the Kalman filter, specialized to the case of no measurement noise. Prediction occurs by means of a difference equation that for each link allows computation of a state estimate for the spatial force at the inner joint using the previously obtained state estimate at the outer joint. Correction occurs in transferring the state estimate across a joint between two adjacent links. In the correction step, the updated state estimate is obtained as a linear weighted combination of the predicted state estimate and the innovations process. The weight multiplying the innovations is a 6×1

matrix playing the role of the Kalman gain. The gain can be computed from the spatial inertia matrix $P(k)$. The Riccati equation satisfied by this matrix involves also the two steps of prediction and correction. Within any given link, prediction occurs from the outer to the inner joint of the link. This prediction step is achieved with an equation that requires multiplication of the spatial inertia matrix $P(k)$ by the spatial transition matrix and its transpose. The spatial inertia matrix associated with the link appears as a driving term in this propagation equation. Correction occurs at each joint by means of the update equation of the discrete Riccati equation.

A second stage involving smoothing uses the results of the above filtering stage. Smoothing is performed by means of an outward recursion that computes a sequence of costates (spatial accelerations) using the innovations as an input. It also produces the desired joint accelerations. The computations involved in this stage are identical to those of the fixed time-interval smoother [2], [3] of linear state estimation theory. The smoother is mechanized with the Bryson–Frazier equations, which also have a predictor–corrector architecture. Prediction occurs in outward propagation of the costates within a link from the inner joint to the outer joint. Correction occurs in propagating the costate across a joint connecting two adjacent links.

If the foregoing filtering and smoothing process is conducted symbolically (instead of numerically), a closed-form expression for the inverse of the composite multilink system inertia matrix results. To this end, the filtering equation is first solved in terms of the transition matrix for the Kalman filter. Then, the smoother equations for the costates are solved symbolically in terms of the transpose of the same transition matrix. Use of the solution for the states in the solution for the costates leads to the desired equation of motion. This equation requires no further matrix inversion in order to compute the joint angle accelerations from the applied joint moments. An interesting byproduct is a recursive technique for direct nonnumerical evaluation of the inverse of the composite system inertia matrix. The recursive equations are identical to those that compute the covariance of the smoothed (as opposed to the filtered) state estimation error in a fixed time-interval smoother. They involve an inward recursion to compute the predicted state estimation error covariance followed by an outward recursion to obtain the covariance of the costates and of the smoothed state estimation error.

The remaining sections of the paper describe the configuration, the notions of spatial force, acceleration and inertia, recursive system dynamics and kinematics, the two-point boundary-value problem, inverse and forward dynamics solutions, closed-form inversion of the inertia matrix, physical interpretation, a planar chain example, relationship to other work, and concluding remarks.

II. Configuration and Problem Statement

Consider a mechanical system of N links numbered $1, \cdots, N$ connected together by N joints numbered $1, \cdots, N$ to form a branch-free kinematic chain. The system is illustrated in Fig. 1.

The links and joints are numbered in an increasing order that goes from the tip of the system toward the base. Joint k in the sequence connects links k and $k + 1$. Joint 0 can be selected at any arbitrary point in link 1. Note that link and joint numbers increase toward the base of the system. This differs from the more common numbering approach in which the numbers increase toward the tip. The ordering shown in Fig. 1 allows a simpler description of the recursive algorithms contained in the paper.

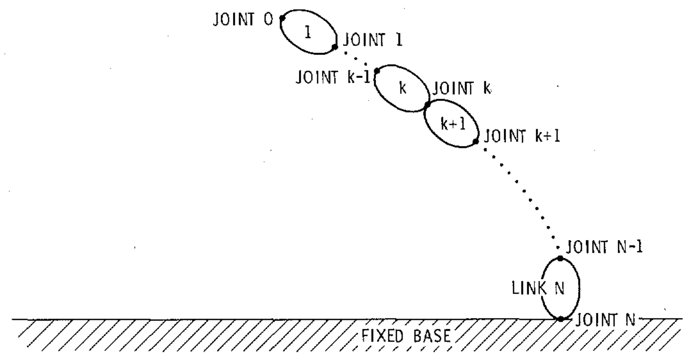


Fig. 1. N-link joint-connected mechanical system.

Let link k be characterized by an inertia tensor $I(k)$ about joint k, a mass $m(k)$, a link vector $L(k)$ from joint k to joint $k - 1$, and a vector $p(k)$ from joint k to the link k mass center.

The joints labeled $1, \cdots, N$ are single-degree-of-freedom joints, which allow rotation along the joint axis only. For these joints, $h(k)$ is a unit vector along the axis of rotation; $\tau(k)$ is the active moment applied about the axis of joint k; and $u(k)$ is the corresponding joint angle which is positive in the right-hand sense about $h(k)$. The relative angular acceleration at joint k is denoted by $a(k)$. The objective is to outline a recursive method for computation of the joint accelerations $a(k)$, given the values of $I(k)$, $m(k)$, $L(k)$, $p(k)$, $h(k)$, and $\tau(k)$. A secondary objective is to solve the closely related inverse problem of computing $\tau(k)$ from the desired accelerations $a(k)$.

III. Spatial Force, Acceleration, and Inertia

To describe simply the recursive dynamics solutions, it is convenient to define the notions of spatial force, acceleration, and inertia [6]. Generally, the term spatial force refers to a 6×1 vector whose first three components are pure moments and whose last three components are forces. Similarly, the term spatial acceleration describes a 6×1 vector of three angular accelerations and three linear accelerations. The link k spatial inertia is a 6×6 matrix that summarizes the mass and inertia properties of link k about joint k. A more detailed definition of these concepts is provided below.

- $T^+(k)$ and $F^+(k)$ are 3×1 vectors representing, respectively, the constraint moment and force acting at joint k on link $k + 1$ and that is due to the adjacent link k. The spatial force is the 6×1 composite vector defined by $x^+(k) = [-T^+(k), -F^+(k)]$ in which the $+$ superscript indicates that the corresponding variable is evaluated at a point on the link $k + 1$ immediately adjacent and on the "positive" side, toward the base, of joint k.

- $T^-(k)$ and $F^-(k)$ are 3×1 vectors representing, respectively, the constraint moment and force acting on link k at joint k. The spatial force is the 6×1 composite vector defined by $x^-(k) = [T^-(k), F^-(k)]$. The $-$ superscript indicates that the corresponding variable is evaluated at a point on link k that is immediately adjacent and on the "negative" side, in an outward direction, of joint k. Note that Newton's third law implies $x^+(k) = x^-(k)$.
- $T(k)$ and $F(k)$ are, respectively, the external moment and the force (due to gravity, for example) acting on link k at its mass center.
- $\omega^+(k)$ and $v^+(k)$ are 3×1 vectors representing, respectively, the angular and linear velocity of a point on link $k+1$ immediately inward of joint k. The corresponding spatial velocity is defined as $V^+(k) = [\omega^+(k), v^+(k)]$. Both angular and linear velocities associated with a link are specified in a coordinate frame attached to the link.
- $\omega^-(k)$ and $v^-(k)$ are 3×1 vectors representing, respectively, the angular and linear velocity of link k on the negative side of joint k. The corresponding spatial velocity is $V^-(k) = [\omega^-(k), v^-(k)]$.
- $\lambda^-(k) = [\dot{\omega}^-(k), \dot{v}^-(k)]$ is a 6×1 vector of angular and linear accelerations of link k at the negative side of joint k. Similarly, $\lambda^+(k) = [\dot{\omega}^+(k), \dot{v}^+(k)]$ is a vector of accelerations at the positive side of joint k. These accelerations are expressed in link k coordinates. The time derivative of the velocity is performed in a coordinate frame in link k.

The spatial inertia matrix $M(k)$ for link k is defined as

$$M(k) = \begin{pmatrix} I(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)U \end{pmatrix} \qquad (1)$$

in which $I(k)$ is the inertia matrix of link k about joint k; $p(k)$ is the vector from joint k to the link k mass center; $\tilde{p}(k)$ is the 3×3 matrix equivalent to the cross-product operation $p(k) \times (\cdot)$; and U is the 3×3 identity. Note that the spatial inertia matrix summarizes the inertia and mass properties of link k about joint k. As an aside, observe that the kinetic energy associated with link k is $(1/2)V^-(k) \circ M(k) \circ V^-(k)$ in which $V^-(k)$ is the 6×1 vector of link k spatial velocities at joint k.

For later reference, it is also convenient to define the following 6×6 matrix:

$$\phi(k, m) = \begin{pmatrix} U & \tilde{L}(k, m) \\ 0 & U \end{pmatrix} \qquad (2)$$

in which $L(k, m)$ is the vector from joint k to joint m; and $\tilde{L}(k, m)$ is the 3×3 matrix equivalent to $L(k, m) \times (\cdot)$. This matrix has the following properties usually associated with a *transition* matrix for a discrete linear state space system [3]:

$$\phi(k, m) = \phi(k, i)\phi(i, m) \qquad \phi(k, k) = U$$
$$\phi^{-1}(k, m) = \phi(m, k) \qquad (3)$$

which state that the matrix satisfies the semigroup property; that it becomes the identity when its two arguments (its subscripts) coincide; and that the matrix inverse is obtained by reversing its two arguments.

IV. Dynamics: Inward Recursion for Spatial Forces

The main objective of this section is to establish that the spatial forces $x(k)$ satisfy

$$x^-(k) = \phi(k, k-1)x^+(k-1) + M(k)\lambda^-(k) + b(k) \qquad (4)$$

$$x^+(k) = x^-(k) \qquad (5)$$

$$x^+(0) = 0 \qquad (6)$$

where $\lambda^-(k)$ is the 6×1 vector of spatial accelerations and $b(k)$ is the bias spatial force

$$b = \begin{pmatrix} \omega \times I \cdot \omega + mp \times [\omega \times v^-] - T - p \times F \\ m\omega \times p + m\omega \times v^- - F \end{pmatrix}. \qquad (7)$$

The argument k has been omitted from all of the variables in (7) to simplify notation.

Proof: Observe [4] that the equation of rotational motion for link k about joint k is

$$I \cdot \dot{\omega} + \omega \times I \cdot \omega + mp \times [\dot{v}^- + \omega \times v^-] = T + T^- + p \times F + T^+(k-1) + L(k) \times F^+(k-1) \qquad (8)$$

in which I is the link k inertia about joint k and m is the link k mass. The argument k has been omitted from all variables except those terms involving forces and moments acting at joint $k-1$. Similarly, the translation of link k is governed by

$$m[\omega \times p + \dot{v}^- + \omega \times v^-] = F + F^- + F^+(k-1). \qquad (9)$$

These two equations combine into (4). To establish (5), observe that

$$-T^+(k) = T^-(k) \qquad -F^+(k) = F^-(k). \qquad (10)$$

Finally, (6) reflects the assumed lack of constraints (due to loads, for example) at the initial joint of the system.

V. Kinematics: Outward Recursion for Spatial Accelerations

The sequence of spatial velocities satisfies

$$V^+(k-1) = \phi^T(k, k-1)V^-(k) \qquad (11)$$

$$V^-(k) = V^+(k) + H^T(k)\dot{u}(k) \qquad (12)$$

in which $V^-(k)$ and $V^+(k)$ are, respectively, the spatial velocities on the negative and positive side of joint k, and $\dot{u}(k)$ is the relative angular velocity at joint k. The accelerations satisfy the closely related recursion

$$\lambda^+(k-1) = \phi^T(k, k-1)\lambda^-(k) \qquad (13)$$

$$\lambda^-(k) = \lambda^+(k) + H^T(k)a(k) + n(k) \qquad (14)$$

where $n(k)$ is the "bias" acceleration

$$n(k)=\begin{pmatrix}\omega^+(k)\times h(k)\dot{u}(k)\\ v^+(k)\times h(k)\dot{u}(k)\end{pmatrix}. \tag{15}$$

Proof: Since link k is rigid, then $\omega^+(k-1)=\omega^-(k)$ and $v^+(k-1)=v^-(k)+\omega^-(k)\times L(k)$. This establishes (11). To establish (12), observe that the linear velocities on both sides of joint k are equal to each other, and that the relative angular velocity in crossing joint k equals the rotation about the joint axis. The recursive relationships (13)–(15) can be established by appropriate time differentiation of (11) and (12).

VI. Two-Point Boundary-Value Problem

The sequences of spatial forces $x(k)$ and spatial accelerations $\lambda(k)$ satisfy the two-point boundary-value problem:

$$x^-(k)=\phi(k,\,k-1)x^+(k-1)+M(k)\lambda^-(k)+b(k) \tag{16}$$

$$x^+(k)=x^-(k) \tag{17}$$

$$\lambda^+(k-1)=\phi^T(k,\,k-1)\lambda^-(k) \tag{18}$$

$$\lambda^-(k)=\lambda^+(k)+H^T(k)a(k)+n(k) \tag{19}$$

$$\tau(k)=H(k)x^+(k) \tag{20}$$

$$x^+(0)=\lambda^+(N)=0. \tag{21}$$

This is a two-point boundary-value problem in the sense that the boundary conditions (21) are satisfied at two distinct points in space: the initial joint at the tip of the system and the terminal joint at the base. The boundary conditions reflect the assumptions that the spatial force vanishes at the tip and that the base is (by definition) immobile. This two-point boundary-value problem is analogous to those encountered in quadratic optimal control and estimation theory for linear systems [3]. Such problems have been investigated extensively to develop filtering and smoothing solutions for dynamical systems. The equivalence between the boundary-value problems of estimation and dynamics is outlined in Table I. A more complete investigation of this equivalence is contained in Section IX.

The above problem can be used as a starting point to solve the following two closely related problems: obtain the moment sequence $\tau(k)$, given knowledge of the joint accelerations $a(k)$; obtain the joint accelerations $a(k)$ from knowledge of the active joint moments $\tau(k)$. These are referred to, respectively, as the inverse and forward dynamics problems and are solved, respectively, in the following two sections.

VII. Inverse Dynamics

The solution to the inverse dynamics problem consists of a two-stage process of outward recursion based on the costate difference equation followed by an inward recursion based on the state equation. This is fundamentally a state-space formulation of the recursive techniques of [7]–[11].

The first stage involves an outward sequential process to determine a sequence of spatial accelerations. This outward recursion is based on (18) and (19) and assumes that the spatial bias accelerations $n(k)$ have been determined previously from the spatial and joint angle velocities $V(k)$ and $\dot{u}(k)$ in (11) and (12). Equation (19) describes an operation by which the spatial acceleration $\lambda^+(k)$ on the positive side of joint k, the joint acceleration $a(k)$, and the bias acceleration $n(k)$ are combined to obtain the updated spatial acceleration $\lambda^-(k)$ at the negative side of joint k. Not shown is a coordinate transformation, that is typically performed in crossing a joint, to convert the spatial accelerations into the coordinate frame of the next link. After the joint is crossed and the acceleration is updated, a propagation step based on (18) is conducted. This step determines the acceleration at the outer joint of link k, given the acceleration at the inner joint k. These two steps of update at a joint followed by propagation from the inner to the outer joint define a sequential technique that generates all of the spatial accelerations. This process is started with the boundary condition $\lambda^+(N)=0$.

TABLE I
EQUIVALENCE BETWEEN TWO-POINT BOUNDARY-VALUE PROBLEMS IN OPTIMAL ESTIMATION AND RECURSIVE ROBOT DYNAMICS

Estimation		Robot Dynamics
States	$x(k)$	spatial forces
Costates	$\lambda(k)$	spatial accelerations
Measurements	$\tau(k)$	active joint moments
Transition matrix	$\phi(k,\,k-1)$	spatial jacobian
Process error covariance	$M(k)$	spatial inertia
Known input	$b(k)$	spatial bias force
State-to-output map	$H(k)$	state-to-joint-axis map

The second stage in the inverse dynamics solution involves an inward sequential process to generate the spatial forces and the applied joint moments. The second stage is based on (16) and (17). Equation (16) involves propagation of the spatial force to the inner joint of a link from the outer joint. Use is made of the previously determined spatial accelerations and bias force. Equation (12) expresses continuity of the magnitude of the spatial force in crossing a joint between two adjacent links. The process starts with the boundary condition that the spatial force vanishes at the initial joint. The process continues inwardly from the tip to the base until a full sequence of spatial forces $x(k)$ has been generated. The active joint moments $\tau(k)$ for $k=1,\cdots,N$ are obtained from the spatial forces by means of the output equation (20).

The boundary-value problem (16)–(21) can also be used to arrive at the traditional second-order matrix equation

$$M(u)a+V(u,\,\dot{u})=\tau \tag{22}$$

where M is the composite multilink system inertia matrix; $a=[a(1),\cdots,a(N)]$ is the vector of joint accelerations; and $\tau=[\tau(1),\cdots,\tau(N)]$ is the vector of joint moments. To derive (22), begin by observing that (18) and (19) imply

$$\lambda^-(j)=\sum_{i=j}^{N}\phi^T(i,\,j)[H^T(i)a(i)+n(i)]. \tag{23}$$

Similarly, the state equations (16) and (17) imply that

$$x^-(k)=\sum_{j=1}^{k}\phi(k,\,j)[M(j)\lambda^-(j)+b(j)]. \tag{24}$$

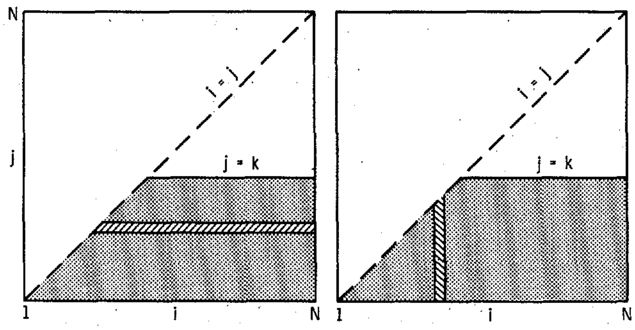


Fig. 2. Illustration of double summation order reversal.

Substitution of (23) in (24) leads to

$$x^{-}(k)=\sum_{j=1}^{k}\sum_{i=j}^{N}\phi(k,j)M(j)\phi^{T}(i,j)[H^{T}(i)a(i)+n(i)]$$
$$+\sum_{j=1}^{k}\phi(k,j)b(j). \quad (25)$$

However, observe the identity

$$\sum_{j=1}^{k}\sum_{i=j}^{N}=\sum_{i=1}^{N}\sum_{j=1}^{\min(i,k)} \quad (26)$$

which can be established by inspection of Fig. 2.

Use of this in (25) implies that

$$x^{-}(k)=\sum_{i=1}^{N}r(k,i)[H^{T}(i)a(i)+n(i)]+\sum_{i=1}^{k}\phi(k,i)b(i) \quad (27)$$

where $r(k, i)$ is the 6×6 matrix

	$1\leq i\leq k\leq N$	$N\geq i\geq k\geq 1$
$r(k, i)$	$\phi(k, i)r(i)$	$r(k)\phi^{T}(i, k)$

$$r(k)=\sum_{j=1}^{k}\phi(k,j)M(j)\phi^{T}(k,j). \quad (28)$$

Observe also that $r(k)$ satisfies the recursive relationship

$$r(k)=\phi(k,k-1)r(k-1)\phi^{T}(k,k-1)+M(k) \quad (29)$$

with the initial condition $r(0) = 0$. This formula is similar to that satisfied by the covariance of the state of a linear discrete-time system driven by a process error with covariance $M(k)$ [3]. Use $\tau(k) = H(k)x^{-}(k)$ in (27) to obtain

$$\sum_{i=1}^{N}m(k,i)a(i)+V^{k}(u,\dot{u})=\tau(k) \quad (30)$$

where $m(k, i) = H(k)r(k, i)H^{T}(i)$ and

$$V^{k}(u,\dot{u})=\sum_{i=1}^{N}H(k)r(k,i)n(i)+\sum_{i=1}^{k}H(k)\phi(k,i)b(i). \quad (31)$$

This is the scalar version of the matrix equation (22). The quantities $m(k, i)$ above are the scalar elements of the inertia matrix M in (22). The elements of the vector $V = [V^1, \cdots, V^N]$ are given by (31).

Physical Interpretation

It is of interest to provide a physical interpretation of the foregoing result. This is done by expressing the above recursions in terms of the changes undergone by the mass, mass center location, and inertia of the composite body outboard of joint $k - 1$ as link k is added to it. The aim is to establish that the inverse dynamics solutions outlined above are equivalent to the composite rigid body method advanced in [11], [12] for its computational efficiency.

It has been shown that the elements $m(k, i)$ of the system inertia matrix M can be determined by

$$\begin{aligned}&\text{initialize state covariance } r(0)=0\\ &\text{propagate state covariance } (1\leq i\leq N)\\ &\qquad r(i)=\phi(i,i-1)r(i-1)\phi^{T}(i,i-1)+M(i)\\ &\text{initialize state } (1\leq i\leq N)\ x(i)=r(i)H^{T}(i)\\ &\text{propagate state } (i\leq k\leq N)\ x(k)=\phi(k,k-1)x(k-1)\\ &\text{inertia matrix element } (i\leq k\leq N)\ m(k,i)=H(k)x(k).\end{aligned} \quad (32)$$

This sequence of steps computes the elements of the inertia matrix in the triangular region $i \leq k \leq N$.

To show that this is equivalent to the composite rigid body method, assume that $r(i)$ can be expressed as

$$r(i)=\begin{pmatrix} J(i) & \tilde{C}(i)\rho(i) \\ -\rho(i)\tilde{C}(i) & \rho(i)U \end{pmatrix} \quad (33)$$

in which $\rho(i)$, $C(i)$, and $J(i)$ are the mass, mass center location, and the inertia of the body formed by links $1, \cdots, i$. The mass center location and the inertia are determined from joint i. Substitution of (33) in (32) leads to the recursions: $\rho(i) = \rho(i-1) + m(i)$ for the mass; $C(i)\rho(i) = [C(i-1) + L(i)]\rho(i-1) + m(i)p(i)$ for the mass center location; and $J(i) = J(i-1) + I(i) + \rho(i-1)[(C(i-1) + L(i))^{T}(C(i-1) + L(i))U - (C(i-1) + L(i))(C(i-1) + L(i))^{T}] - \rho(i-1)[C^{T}(i-1)C(i-1)U - C(i-1)C^{T}(i-1)]$ for the inertia. These recursions are started with the initial conditions $\rho(0) = 0$, $C(0) = 0$, and $J(0) = 0$.

The diagonal elements of the composite system inertia matrix are computed by $m(i, i) = h^{T}(i)J(i)h(i)$. The off-diagonal elements $m(k, i)$ in the triangular region $i \leq k \leq N$ are computed by the last two equations in (32).

One of the advantages of (33) is that the spatial inertia matrix is parameterized with a minimal number of ten parameters (one for the mass, three for the mass center location, and six for the rotational inertia). The number of parameters is reduced from the maximum number of 21 required to characterize a 6×6 symmetric matrix. This reduces the number of computations. This is an illustration of one of the important points of the paper. The state space formulation to spatial recursions is conceptually very simple as illustrated in (32). Computational efficiency can be obtained by using a suitable parametric characterization for the equations.

VIII. Forward Dynamics

The forward dynamics problem is to find the accelerations $a(k)$ at the joints, given the applied joint moments $\tau(k)$. The problem is solved using (16)–(21) and what is referred to as the *sweep method* in [3]. The sweep method begins by assuming that the state $x(k)$ and the costate $\lambda(k)$ are related by

$$x(k)=z(k)+P(k)\lambda(k) \tag{34}$$

where $z(k)$ and $P(k)$ are to be determined by means of recursive equations that emerge upon substitution of (34) in (16)–(21). There, $z(k)$ will play the role that the predicted state estimate plays in the Kalman filter, and $P(k)$ will play the role of the corresponding state estimation error covariance.

Result

The joint accelerations $a(k)$ can be computed from the applied joint moments $\tau(k)$ by means of the two-stage process of inward filtering and outward smoothing.

Filtering:

$$\text{initial conditions } z^+(0)=0;\ P^+(0)=0 \tag{35}$$

$$\text{state prediction } z^-(k)=\phi(k,\ k-1)z^+(i)+b(k) \tag{36}$$

inertia prediction $P^-(k)$

$$=\phi(k,\ k-1)P^+(k-1)\phi^T(k,\ k-1)+M(k) \tag{37}$$

$$\text{joint axis inertia } D(k)=H(k)P^-(k)H^T(k) \tag{38}$$

$$\text{gain } G(k)=P^-(k)H^T(k)/D(k) \tag{39}$$

$$\text{innovations } e^-(k)=\tau(k)-H(k)z^-(k) \tag{40}$$

$$\text{state update } z^+(k)=z^-(k)+G(k)e^-(k)+P^+(k)n(k) \tag{41}$$

$$\text{residuals } e^+(k)=e^-(k)/D(k) \tag{42}$$

inertia update $P^+(k)=P^-(k)$

$$-P^-(k)H^T(k)H(k)P^-(k)/D(k). \tag{43}$$

The residuals $e^+(k)$ and the gains $G(k)$ are stored in this stage. The scalar $D(k)$, whose inversion is required to compute the gain $G(k)$, represents the inertia along the joint k axis of the composite body formed by the links outboard of this joint. All of the joints outboard of joint k are unlocked in defining the inertia $D(k)$.

Smoothing:

$$\text{terminal costate } \lambda^+(N)=0 \tag{44}$$

$$\text{costate propagation } \lambda^+(k-1)=\phi^T(k,\ k-1)\lambda^-(k) \tag{45}$$

$$\text{joint accelerations } a(k)=e^+(k)-G^T(k)[\lambda^+(k)+n(k)] \tag{46}$$

$$\text{costate update } \lambda^-(k)=\lambda^+(k)+n(k)+H^T(k)a(k). \tag{47}$$

Not shown explicitly is a link k to link $k+1$ coordinate transformation that is performed immediately after a joint has been crossed and the state and spatial inertia have been updated in (41) and (43). A similar transformation from link $k+1$ to link k is performed after the costate propagation in (45).

Proof: Substitute (34) in (20) to obtain $e^-(k) = H(k)P^-(k)\lambda^-(k)$ where $e^-(k)$ is the innovations process defined by (40). Now, substitute (19) in this to obtain (46). Use of this in (19) leads to

$$\lambda^-(k)=[I-G(k)H(k)]^T[\lambda^+(k)+n(k)]+H^T(k)e^+(k) \tag{48}$$

which provides an alternative to (47) in updating the costate at joint k. Substitute (34) in (16) to obtain

$$\begin{aligned}z^-(k)+P^-(k)\lambda^-(k)&=\phi(k,\ k-1)[z^+(k-1)\\&\quad+P^+(k-1)\lambda^+(k-1)]+M(k)\lambda^-(k)+b(k).\end{aligned} \tag{49}$$

Use (18) and observe that the state and spatial inertia propagation equations (36) and (37) are sufficient conditions to satisfy (49). To obtain the state and inertia update equations, observe that (17) and (34) imply $z^+(k) + P^+(k)\lambda^+(k) = z^-(k) + P^-(k)\lambda^-(k)$. Finally, substitute (48) on the right side of this. As an aside, note that the spatial inertia can also be updated by

$$P^+(k)=[I-G(k)H(k)]P^-(k)=P^-(k)[I-G(k)H(k)]^T \tag{50}$$

$$P^+(k)=[I-G(k)H(k)]P^-(k)[I-G(k)H(k)]^T \tag{51}$$

as is well-known in Kalman filtering [3]. These two equations can be obtained easily from the inertia update equation (43).

IX. Similarities to Kalman Filtering and Bryson–Frazier Smoothing

The two-point boundary-value problem of Section VI and the filtering and smoothing equations of Section VIII are analogous to those typically used to obtain the best smoothed state estimate of a discrete-time state space system with discrete measurements (for the special case of no measurement noise). To examine this analogy more closely, consider the following system:

$$x(k)=\phi(k,\ k-1)x(k-1)+w(k) \qquad \tau(k)=H(k)x(k) \tag{52}$$

where $x(k)$ is the state; $\tau(k)$ is the observation; $\phi(k,\ k-1)$ is the transition matrix; $H(k)$ is the state-to-measurement map; and $w(k)$ is a white-Gaussian process with mean and covariance specified by

$$E[w(k)]=b(k) \qquad E[\tilde{w}(k)\tilde{w}^T(k)]=M(k) \tag{53}$$

with $\tilde{w}(k) = w(k) - b(k)$. To simplify the discussion, it is assumed in this section that the acceleration bias term $n(k)$ has been set to zero.

The filtering problem consists of finding

$$z^-(k)=E[x(k)/\tau(1), \cdots, \tau(k-1)]. \tag{54}$$

This is the best estimate of the state at time k given all of the previous measurements. Closely related to this filtered estimate is the innovations process defined by

$$e^-(k)=\tau(k)-H(k)z^-(k). \tag{55}$$

The filtered state estimation error covariance is

$$E[x(k)-z^-(k)][x(k)-z^-(k)]^T=P^-(k) \tag{56}$$

which is known [3] to satisfy the discrete Riccati equation. The covariance of the innovations is known to be

$$E[e^-(k)e^-(k)]=D(k). \tag{57}$$

Note that (57) is obtained from the more general formula $D(k) = H(k)P^-(k)H^T(k) + R(k)$ by setting the measurement noise covariance $R(k)$ to zero. The equations for the Kalman gain and the updated covariance are (39) and (43). The updated state estimation error covariance $P^+(k)$ can be shown to be

$$E[x(k)-z^+(k)][x(k)-z^+(k)]^T=P^+(k) \tag{58}$$

where $z^+(k) = z^-(k) + G(k)e^-(k)$ is the updated state estimate

$$z^+(k)=E[x(k)/\tau(1), \cdots, \tau(k)]. \tag{59}$$

The smoothing problem associated with (52) and (53) is to find

$$x(k)=E[x(k)/\tau(1), \cdots, \tau(N)] \tag{60}$$

the best estimate of the state, given all of the data $\tau = [\tau(1), \cdots, \tau(N)]$ at the N measurement locations. It is known [3] that the best smoothed estimate can be generated by means of the Bryson–Frazier equations

$$\hat{x}(k)=z(k)+P(k)\lambda(k) \tag{61}$$

where $\lambda(k)$ are the costates specified by

$$\lambda^+(k-1)=\phi^T(k, k-1)\lambda^-(k) \tag{62}$$

$$\lambda^-(k)=[I-G(k)H(k)]^T\lambda^+(k)+H^T(k)e^+(k). \tag{63}$$

The error covariance $S(k)$ associated with the smoothed state estimate is defined as

$$S(k)=E[x(k)-\hat{z}(k)][x(k)-\hat{x}(k)]^T. \tag{64}$$

This matrix is given by [3]

$$S(k)=P(k)-P(k)\Delta(k)P(k) \tag{65}$$

where $\Delta(k)$ is the costate covariance defined as

$$\Delta(k)=E[\lambda(k)\lambda^T(k)]. \tag{66}$$

It is known also that $\Delta(k)$ satisfies the recursive relationships

$$\Delta^+(k-1)=\phi^T(k, k-1)\Delta^-(k)\phi(k, k-1) \tag{67}$$

$$\begin{aligned}\Delta^-(k)=[I-G(k)H(k)]\Delta^+(k)[I-G(k)H(k)]^T\\ +H^T(k)H(k)/D(k).\end{aligned} \tag{68}$$

This is a backward recursion consisting of propagation in (67) followed by an update in (68). The boundary condition $\Delta^+(N) = 0$ is valid at sample N.

It is possible to obtain the closed-form inverse of the inertia matrix in terms of a pair of matrices analogous to $P(k)$ and $\Delta(k)$ above. This is done in the next section.

X. Closed-Form Inertia Matrix Inverse

The central objective is to obtain the following equation:

$$a(k)=a^1(k)+a^2(k)+a^3(k) \tag{69}$$

in which $a^1(k)$, $a^2(k)$, and $a^3(k)$ are the joint angle accelerations due to the applied joint moments $\tau(k)$, the bias spatial forces $b(k)$, and the bias spatial accelerations $n(k)$. The three acceleration components are given by

$$\begin{aligned}a^1(k)&=c(k)\tau(k)+d(k)\sum_{i=1}^{k-1}\psi(k^-, i^+)G(i)\tau(i)\\ &\quad+G^T(k)\sum_{i=k+1}^{N}\psi^T(i^-, k^+)d^T(i)\tau(i)\\ a^2(k)&=d(k)b(k)+d(k)\sum_{i=1}^{k-1}\psi(k^-, i^-)b(i)\\ &\quad+G^T(k)\sum_{i=k+1}^{N}\psi^T(i^-, k^+)\Delta^-(i)b(i)\\ a^3(k)&=G^T(k)[\Delta^+(k)P^+(k)-I]n(k)\\ &\quad+d(k)\sum_{i=1}^{k-1}\psi(k^-, i^+)P^+(i)n(i)\\ &\quad+G^T(k)\sum_{i=k+1}^{N}\psi^T(i^+, k^+)\\ &\quad\cdot[\Delta^+(i)P^+(i)-I]n(i)\end{aligned} \tag{70}$$

where $c(k)$ and $d(k)$ are the scalar and the 1×6 vector

$$\begin{aligned}c(k)&=D^{-1}(k)+G^T(k)\Delta^+(k)G(k)\\ d(k)&=G^T(k)\Delta^+(k)-c(k)H(k).\end{aligned} \tag{71}$$

The transition matrix $\psi(k^-, i^+)$ is defined as

$$\psi(k^-, i^+)=\phi(k, k-1)\prod_{j=i+1}^{k-1}[I-G(j)H(j)]\phi(j, j-1). \tag{72}$$

Note the definitions $\psi(k^-, i^-) = \psi(k^-, i^+)[I - G(i)H(i)]$ and $\psi(k^+, i^+) = [I - G(k)H(k)]\psi(k^-, i^+)$. The matrix $\psi(k^-, i^+)$ is the transition matrix for the Kalman filter. Its two arguments k^- and i^+ represent, respectively, the negative side of joint k and the positive side of joint i. The sequence of multiplications on the right side of (72) is taken along the path from joint i to joint k.

Recall that (22) implies that $a = M^{-1}(u)[\tau - V(u, \dot{u})]$, where M is the inertia matrix. Hence the elements of its inverse can be obtained by inspection of $a^1(k)$ in (70).

The overall approach used to arrive at (69) is based on solving both the state and costate difference equations in terms of their corresponding, and mutually adjoint, weighting kernels. Substitution of the costate solution into the state solution leads to the desired result. This is now performed in detail for the acceleration component in (69) due to the applied joint moments. The other two components, due to bias forces and accelerations, can be evaluated in a similar fashion.

Solution of the State Equation

The aim here is to show that the sequences of "predicted" spatial forces and residuals are specified by

$$z^-(k) = \sum_{i=1}^{k-1} \psi(k^-, i^+)G(i)\tau(i) \tag{73}$$

$$e^+(k) = D^{-1}(k)\left[\tau(k) - H(k)\sum_{i=1}^{k-1} \psi(k^-, i^+)G(i)\tau(i)\right]. \tag{74}$$

To this end, observe that substitution of (36) in (41) implies $z^+(k) = \psi(k^+, k-1^+)z^+(k-1) + G(k)\tau(k)$. Hence $z^+(k) = \Sigma_{j=1}^{k} \psi(k^+, j^+)G(j)\tau(j)$. Now use (36) to obtain (73). This, together with (40) and (42), implies (74).

Solution of the Costate Equation

Use of similar arguments can be made to show that (44)–(47) imply

$$\lambda^+(k) = \sum_{i=k+1}^{N} \psi^T(i^-, k^+)H^T(i)e^+(i) \tag{75}$$

$$a^1(k) = e^+(k) - G^T(k)\sum_{i=k+1}^{N} \psi^T(i^-, k^+)H^T(i)e^+(i). \tag{76}$$

Joint Accelerations Due to Joint Moments

The objective here is to obtain $a^1(k)$ in (69). To this end, substitute (74) in (76) to obtain

$$a^1(k) = e^+(k) - G^T(k)\sum_{i=k+1}^{N} \psi^T(i^-, k^+)H^T(i)D^{-1}(i) \cdot \left[\tau(i) - H(i)\sum_{j=1}^{i-1} \psi(i^-, j^+)G(j)\tau(j)\right]. \tag{77}$$

Recall the identity

$$\sum_{i=k+1}^{N}\sum_{j=1}^{i-1} = \sum_{j=1}^{N-1}\sum_{i=m}^{N-1} \tag{78}$$

with $m = \max(k+1, i+1)$. Observe that

$$\begin{aligned}\sum_{i=k+1}^{N}\sum_{j=1}^{i-1} &\psi^T(i^-, k^+)H^T(i)D^{-1}(i)H(i)\psi(i^-, j^+)G(j)\tau(j)\\ &= \Delta^+(k)G(k)\tau(k) + \Delta^+(k)\sum_{j=1}^{k-1}\psi(k^+, j^+)G(j)\tau(j)\\ &\quad + \sum_{j=k+1}^{N}\psi^T(j^+, k^+)\Delta^+(j)G(j)\tau(j)\end{aligned} \tag{79}$$

where

$$\Delta^+(k) = \sum_{i=k+1}^{N} \psi^T(i^-, k^+)H^T(i)D^{-1}(i)H(i)\psi(i^-, k^+). \tag{80}$$

In arriving at the upper limit of summation for the last term in (79), use has been made of the terminal condition $\Delta^+(N) = 0$ implied by (80). Finally, use of (74) and (79) in (76) leads to $a^1(k)$ in (69). This evaluates the acceleration component due to the applied joint moments. The components due to bias forces and accelerations can be obtained similarly.

Observe that (80) implies that the sequence $\Delta(k)$ satisfies the recursive equations

$$\Delta^+(k-1) = \phi^T(k, k-1)\Delta^-(k)\phi(k, k-1) \tag{81}$$

$$\Delta^-(k) = [I - G(k)H(k)]^T\Delta^+(k)[I - G(k)H(k)] + H^T(k)H(k)/D(k). \tag{82}$$

These equations are identical to the ones satisfied by the costate variable covariance of the fixed-time smoother in Section IX.

Recursive Evaluation of Inertia Matrix Inverse

The above results imply that the inverse of the inertia matrix can be computed recursively by means of

initial inertia $P^+(0) = 0$

inertia prediction $(1 \le i \le N)$

$$P^-(i) = \phi(i, i-1)P^+(i-1)\phi^T(i, i-1) + M(i)$$

inertia update $(1 \le i \le N)$

$$P^+(i) = P^-(i) - P^-(i)H^T(i)H(i)P^-(i)/D(i)$$

terminal costate covariance $\Delta^+(N) = 0$

costate covariance update $(N \ge i \ge 1)$

$$\Delta^-(i) = [I - G(i)H(i)]^T\Delta^+(i)[I - G(i)H(i)] + H(i)H^T(i)/D(i)$$

costate covariance propagation $(N-1\geq i\geq 1)$

$$\Delta^{+}(i)=\phi(i+1,\, i)\Delta^{-}(i+1)\phi(i+1,\, i)$$

inertia inverse diagonal element $(N\geq i\geq 1)$

$$m^{-1}(i,\, i)=D^{-1}(i)+G^{T}(i)\Delta^{+}(i)G(i)$$

terminal costate $(k=i)$

$$\alpha^{-}(k)=\Delta^{+}(i)G(i)-m^{-1}(i,\, i)H^{T}(i)$$

costate propagation $(i-1\geq k\geq 1)$

$$\alpha^{+}(k)=\phi(k+1,\, k)\alpha^{-}(k+1)$$

inertia inverse off-diagonal element

$$m^{-1}(k,\, i)=G^{T}(k)\alpha^{+}(k)$$

costate update $(i-1\geq k\geq 1)$

$$\alpha^{-}(k)=[I-G(k)H(k)]^{T}\alpha^{+}(k). \quad (83)$$

The above steps determine the elements $m^{-1}(k,\, i)$ of the inverse of the inertia matrix in the triangular region $k\leq i\leq N$.

The above computations are analogous to those in (32), which determine the elements of the inertia matrix recursively. However, for the inverse dynamics problem of Section VII it was possible to use a minimum number of ten parameters to characterize (32). These parameters corresponded to the mass, mass center, and inertia of the composite body outboard of joint k. It is of interest to investigate the consequences of using a similar set of parameters to characterize (83). For example, use of the same parameter set for the solution $P(k)$ to the discrete Riccati equation leads to

mass propagation $\rho(k)\rightarrow\rho(k-1)+m(k)U$

mass update $\rho(k)\rightarrow\rho(k)$

$$-\rho(k)C^{T}(k)h(k)h^{T}(k)C(k)\rho(k)/D(k)$$

mass center propagation $C(k)\rho(k)\rightarrow[C(k-1)$

$$+\tilde{L}(k)]\rho(k-1)+m(k)\tilde{p}(k)$$

mass center update $C(k)\rho(k)$

$$\rightarrow[I-J(k)h(k)h^{T}(k)/D(k)]C(k)\rho(k)$$

inertia propagation $J(k)\rightarrow J(k-1)+I(k)$

$$+(C(k-1)+\tilde{L}(k))\rho(k-1)(C(k-1)+\tilde{L}(k))^{T}-C(k-1)\rho(k-1)C^{T}(k-1)$$

inertia update $J(k)\rightarrow[I-J(k)h(k)h^{T}(k)/D(k)]J(k)$ (84)

in which $D(k)\;=\;h^{T}(k)\,J(k)h(k)$. These equations can be viewed as the extensions to the forward dynamics case of the composite rigid body method for inverse dynamics. However, in contrast to the earlier results for the inverse dynamics problem, use of the mass, mass center, and inertia parameters of the composite rigid body does not appear to reduce the number of computations. The main reason for this is that the mass $\rho(k)$ is a full symmetric 3×3 matrix, whereas in the inverse dynamics problem, this matrix was reduced to a simple scalar. Similarly, the mass center $C(k)$ appears to be a full 3×3 matrix, which may not be symmetric. Hence the number of independent parameters in (84) is 21 (six each for the mass and inertia, and nine for the mass center). Use of (84) in place (83), while providing some physical insight, does not appear to lead to computational savings. Equation (83) appears to provide a more efficient means to evaluate the inertia matrix inverse recursively. However, this requires that $P(k)$ be characterized by the 21 independent parameters required to specify an arbitrary, symmetric 6×6 matrix. A similar number is also required for the matrix $\Delta(k)$. More investigation is required to determine if there is a better parameter set to characterize (83).

XI. Physical Interpretation

This section explores more completely the underlying reasons for the analogy between estimation theory and robot dynamics. This is done by examining two areas: 1) the use of a spatially random state space model for the dynamics of a typical link; and 2) description of the filtering and smoothing equations in terms of physical quantities.

Spatially Random Model

First, observe that the robot arm under study here is a serial structure in which the links and joints are numbered in increasing order. Because of this, the link numbering approach allows definition of the notion of past, present, and future. For example, a sequence that starts at the tip and goes to the base is analogous to a dynamical process that evolves forward from an initial to a terminal time. If a given link lies closer to the base than another link, then the given link can be said to be in the future. Similarly, if a link lies closer to the tip, then it can be said to be in the past. This notion of causality in turn allows for definition of the notion of prediction, which in this case means to estimate the spatial forces in the future given only past and present information.

Now, develop a causal model for one-step prediction of the spatial force at the inner joint of a link, given the spatial force at the outer joint. Such a model, contained in (52), is based on the equation $x(k)\;=\;\phi(k,\;k\;-\;1)x(k\;-\;1)\;+\;w(k)$. The model has a built-in error represented by the term $w(k)$ in this equation. This error is due to the fact that the link is in general moving and undergoing accelerations. There is a force acting on the link that is due to the acceleration and mass distribution of the link. This force is referred to typically as a D'Alembert force. One of the fundamental assumptions made in this paper is that this force is initially (before estimation occurs) modeled as a spatially distributed white-noise process with mean and covariance given by the spatial bias force $b(k)$ and the spatial inertia matrix $M(k)$. This takes the point of view of a local observer whose perception is by definition confined to the location of joint k itself. This observer is assumed to know nothing about the applied moments acting at the joints or about the accelerations acting on any of the adjoining links. The observer assumes that its own acceleration at joint k is uncorrelated with the acceleration at the remaining joints. It should be pointed out that the white-noise model is only an *a* *priori* model. There is uncertainty in the inertial forces only before the estimation process is conducted. After estimation takes place, this uncertainty no longer exists. The estimation error covariance of the joint angle accelerations that emerge from the filter and smoother vanishes.

To complete the state space model requires an output or measurement equation, to model the relationship between the state of the system and its output. This equation is $\tau(k) = H(k)x^+(k)$ which states that the measurements can be obtained from the state by projection into the appropriate joint axis. The measurement equation has no uncertainty (no measurement error) because the active moments are assumed to be deterministic quantities.

The above ingredients are identical to those that occur in state estimation theory. The system is modeled as a combination of a deterministic model to propagate states based on the transition matrix, an additive random noise model for uncertainty, and a measurement equation to model the relationship between states and measurements. The estimation problem consists of estimating the states (and the inertial forces), given the set of measurements. The corresponding dynamics problem is to find the forces and accelerations, given the active moments at the joints. The equations that solve these two equivalent problems are analyzed below from the mutually complementary points of view of dynamics and estimation theory.

Filtering and Smoothing

The filtering process starts at a fictitious joint 0 attached to the tip body. The state and spatial inertia initial conditions at this joint are both zero. The initial state estimate is zero, because there are no external forces and moments acting at the initial joint. Since this knowledge is precise, there is no uncertainty, and the corresponding estimation error covariance is zero. The initial condition for the Riccati equation is therefore zero. This is from the point of view of estimation theory. From the point of view of mechanics, the spatial inertia at the initial joint is zero because by definition there is no inertia outboard of this joint (the initial joint is assumed to be on the surface of the initial link). From this initial condition, the filtering equations proceed with the by now classical predictor–corrector architecture of the Kalman filter (Fig. 3).

The step of prediction in (36) and (37) involves crossing link k from joint $k-1$ to joint k. A prediction step ① is used to propagate the state from the outer to the inner joint. The bias force ② is added as a deterministic input in this step. The spatial inertia is also propagated by means of the prediction step in the Riccati equation. This equation reflects the increase in the spatial inertia due to the addition of a link. From the viewpoint of estimation theory, the prediction step reflects increases in the state estimation error covariance, which are due to the buildup of uncertainty that occurs because the inertial forces are assumed to be random. Knowledge of the covariance allows for compensation of this uncertainty. This is done in the correction step that follows.

Correction occurs in crossing a joint from one link to the next. It involves updating the state estimate and spatial inertia.

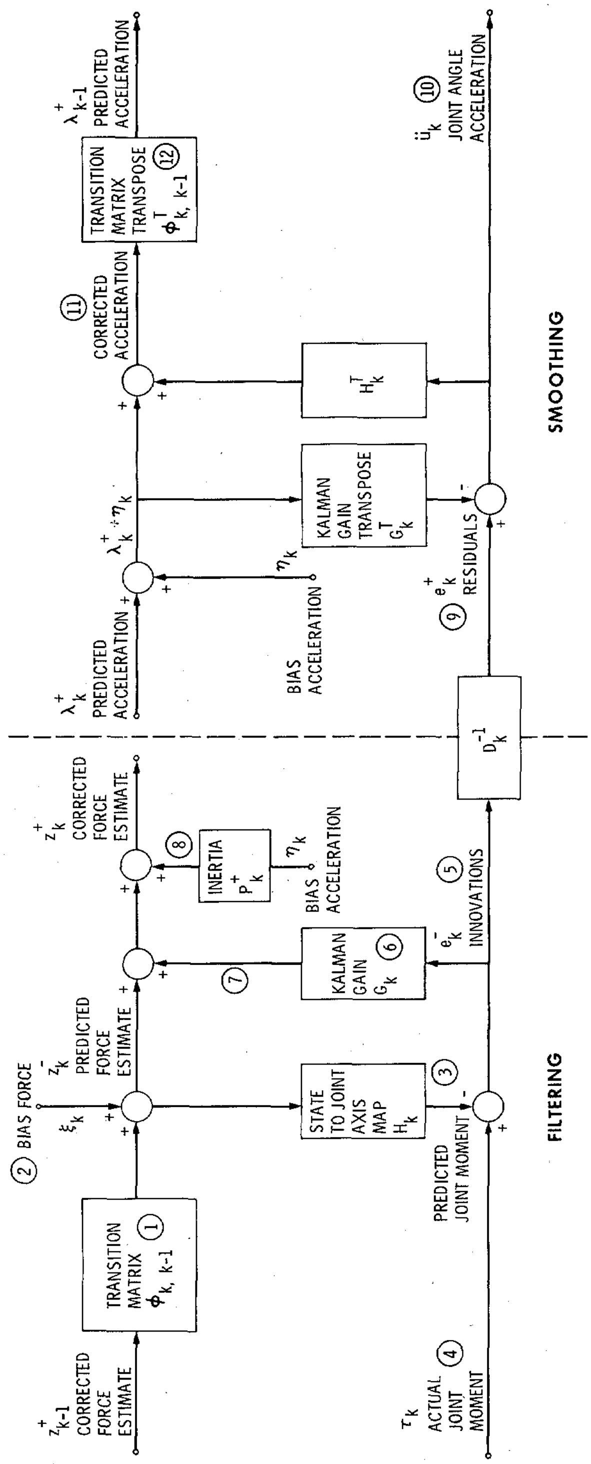


Fig. 3. Filtering and smoothing architecture.

Before the correction step occurs, there are two estimates for the moment about the joint axis. One of the estimates ③ comes from the previously conducted prediction step. This estimate has an inherent error with a known covariance. The second estimate ④ comes from the measurement itself and has no error. These two estimates are combined optimally in the state update equation. An error (innovations) term ⑤ is first created that represents the difference between the actual moment (measurement) and the predicted moment. Note that the innovations process has the units of a pure moment. The innovations process is multiplied by the Kalman gain ⑥ to form the correction term ⑦ that is then added to the predicted state estimate. The state update also involves addition of the bias term ⑧ due to the bias acceleration. Note that this term has the physical units of a spatial force and is the product of the updated spatial inertia at a joint and the bias acceleration at the same joint.

Central to the above correction step is the Kalman gain determined from the propagated inertia by means of (38). The Kalman gain governs the relative weighting between the predicted state estimate and the correction term. In approximate terms, if the spatial inertia of the body outboard of joint k is large, then the uncertainty in the predicted estimate is large. The resulting gain is large, and the correction term is weighted more heavily than the prediction term. In contrast, if the inertia of the same composite body is small, the uncertainty in the predicted state estimate is also small. More reliance is then placed on this estimate. Note that there is a gain associated with each of the joints. Each gain is a 6×1 vector. For each joint, the component of the gain along the joint axis is the identity. This can be observed from the condition $H(k)P^-(k) = 0$. The inertia along the joint axis is zero after the update occurs. Equivalently, the covariance of the state estimation error along the joint axis is zero after the update. This means that the updated state estimate in the joint axis has no error. This is as expected because the measurement equation has no noise. From the point of view of mechanics, the spatial force in the direction of the joint axis is the active joint moment. Since this moment is known precisely (and is in fact imposed by the actuating device assumed at the joint), then the component of the state estimate in the joint axis direction is simply set equal to this active moment.

An additional outcome of the filtering equations is the residual process ⑨. The residual at joint k is an estimate of the joint k angular acceleration, under the assumption that all of the future joints are locked. It is a causal estimate in the sense that it is based only on past and present joint moments. It does not depend on future joint moments. There is an inherent potential error in this estimate, because the assumption of no future acceleration may not be valid. However, this error is compensated for in the smoothing stage that follows.

The inputs to the smoothing stage are a set of residuals. The Kalman gains are also assumed to be known. The outputs are a set of joint angle accelerations ⑩, which are the final result of the forward dynamics computations. The acceleration that is computed at any given joint is that due to past, present, and future joint moments. This is in contrast to the joint accelerations (residuals) computed in the filtering stage, in which only past and present joint moments are used in computing the residual acceleration at a given joint.

The smoothing stage also has an architecture that involves prediction and correction. The correction step is used to cross a joint, while the prediction step is used to cross a link. In the update step, a joint angle acceleration is computed, and a new spatial acceleration ⑪ is determined at the inner joint of the next link. In the prediction step, spatial accelerations are propagated outward to the outer joint. The transpose ⑫ of the transition matrix is used to do this. The smoothing stage proceeds sequentially from the base to the tip. At the end the filtering and smoothing computations, all of the joints have been crossed twice, once in the inward direction and again in the outward direction. In the second crossing, the correction due to applied joint moments in the future takes place.

Observe in Fig. 3 that the filtering and smoothing stages can be viewed as mirror images of each other about a vertical line that cuts the diagram in two. This is a graphical illustration of the result that the filtering and smoothing algorithms factor the inverse of the composite multibody system inertia matrix as $M^{-1} = (I - L^*)D^{-1}(I - L)$ in which L is a lower triangular matrix, and L^* is its upper triangular transpose. The matrix D is diagonal. This result is developed in [5].

XII. Planar Chain Example

To illustrate the foregoing ideas, consider a simple example in which only planar motion is allowed. The example illustrates the relative ease with which the filter and smoother are set up. It also provides a means to estimate the number of arithmetic operations required. This is done parametrically in terms of the number of links in the chain.

Consider a configuration in which the system in Fig. 1 lies on a plane, and the axis of rotation for each of the joints is orthogonal to this plane. Define a coordinate frame attached to an arbitrary link k. This frame has the x axis along the direction of rotation, the y axis in the direction of a vector from joint $k - 1$ to joint k, and the z axis in a direction orthogonal to both the x and y axes. The state of the system at joint k is a three-dimensional vector $x = [x(1), x(2), x(3)]$, in which $x(1)$ is a moment in the x direction and $x(2)$ and $x(3)$ are forces in the y and z directions, respectively. For simplicity, the argument k, which identifies the joint at which the state is defined, is omitted. The corresponding costate is $\lambda = [\lambda(1), \lambda(2), \lambda(3)]$ where $\lambda(1)$ is a rotational acceleration along the x axis, and $\lambda(2)$ and $\lambda(3)$ are linear accelerations in the y and z directions.

The spatial transition and inertia matrices are

$$\phi = \begin{pmatrix} 1 & 0 & -l \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad M = \begin{pmatrix} I & 0 & -mp \\ 0 & m & 0 \\ -mp & 0 & m \end{pmatrix} \tag{85}$$

in which l is the length of link k, m is the link k mass, I is the link k inertia about joint k, and p is the scalar distance from joint k to the link k mass center. For simplicity, the bias force and accelerations terms b and n are set to zero.

State propagation to cross a link is achieved by means of the equation $z \to \phi z$ which in x, y, z component notation becomes

$z(1) \rightarrow z(1) - lz(3)$. Observe that the components $z(2)$ and $z(3)$ do not change in the propagation step. Spatial inertia propagation is achieved by the matrix equation $P \rightarrow \phi P \phi^T + M$ in which the spatial inertia (covariance) matrix is

$$P = \begin{pmatrix} P(1) & P(2) & P(4) \\ P(2) & P(3) & P(5) \\ P(4) & P(5) & P(6) \end{pmatrix}. \tag{86}$$

Since P is symmetric, it can be characterized by a total of six parameters (in the planar case). In terms of these parameters, the spatial inertia propagation equations become $P(1) \rightarrow l^2P(6) + I$, $P(2) \rightarrow -lP(5)$, $P(3) \rightarrow P(3) + m$, $P(4) \rightarrow -lP(6) - mp$, $P(6) \rightarrow P(6) + m$. Note that the preceding computations are simplified by the conditions that $P(1) = P(2) = P(4) = 0$ just before the propagation step is conducted. This follows because of $HP^+ = 0$ and $P^+H^T = 0$, which imply that the components of the spatial inertia P^+ in the joint axis direction vanish. The Kalman gain is computed by means of the equations $D \rightarrow HPH^T$ and $G = PH^TD^{-1}$ which in scalar notation becomes $D \rightarrow P(1)$, $g(1) = 1$, $g(2) \rightarrow P(2)/P(1)$, $g(3) \rightarrow P(4)/P(1)$.

State update to cross a joint is achieved by means of the equation $z \rightarrow z + Ge^-$ in which e^- is the innovations process. In scalar notation, this becomes $z(1) \rightarrow \tau$, $z(2) \rightarrow z(2) + g(2)e^-$, $z(3) \rightarrow z(3) + g(3)e^-$. The residual process is computed by $e^+ = e^-/D$. The spatial inertia update is accomplished by means of $P \rightarrow P - PH^TD^{-1}HP$ which in terms of component notation becomes $P(1), P(2), P(4) \rightarrow 0$, $P(3) \rightarrow P(3) - g(2)P(2)$, $P(5) \rightarrow P(5) - g(3)P(2)$, $P(6) \rightarrow P(6) - g(3)P(4)$. Note that $P(1) = P(2) = P(4) = 0$ after the update step, because the absence of measurement noise allows the Kalman filter to produce an estimate that has no error in the direction of rotation at any given joint. A coordinate transformation is performed immediately after the update step in order to transform the state estimate into the coordinate frame of the next link.

In the smoothing stage, the costates are propagated backward by means of $\lambda(3) \rightarrow \lambda(3) - l\lambda(1)$. After the propagation step, the costates are transformed to the coordinate frame of the next link. The joint acceleration is then computed by $a \rightarrow e^+ - \lambda(1) - g(2)\lambda(2) - g(3)\lambda(3)$. This acceleration is then used to update the costate $\lambda(1) \rightarrow \lambda(1) + a$. Note that only the joint-axis component of the costates needs to be updated at any given joint, and that the components in the other two axes remain unchanged.

The number of arithmetic operations required per link in an N link system is summarized in Table II.

The count for the update equations for the state, the inertia, and the costate includes the number of operations required for the coordinate transformations required to cross joints. Based on the above count, an approximate estimate of $27N$ flops is obtained for an N link system.

It is of interest to compare the above number of operations with those required to compute the multilink system inertia matrix, using (32), and to invert this matrix numerically. To do this, assume that the state covariance matrix is characterized in the planar case by a minimal set of four parameters representing the mass, mass center location, and rotational inertia.

TABLE II
NUMBER OF ARITHMETIC OPERATIONS PER LINK REQUIRED BY BRYSON–FRAZIER SMOOTHER FOR A PLANAR CHAIN

	Add	Multiply	Divide
State propagation	1	1	0
Inertia propagation	4	3	0
Gain	0	0	2
Innovations	1	0	0
Residuals	0	0	1
State update	5	6	0
Inertia update	10	9	0
Costate propagation	1	1	0
Joint acceleration	3	2	0
Costate update	3	4	0
Total	28	26	3

TABLE III
NUMBER OF ARITHMETIC OPERATIONS REQUIRED FOR ASSEMBLY OF N-BY-N INERTIA MATRIX

	Add	Multiply	Repetitions
Spatial inertia propagation			
Mass	1	0	N
Mass center	2	1	N
Rotational inertia	3	3	N
State propagation	1	1	$N(N+1)/2$
State update	2	4	$N(N+1)/2$

To form the diagonal elements of the inertia matrix, all of the above steps except the last two (state propagation and update) must be repeated N times. The last two steps must be repeated a total number of $N(N+1)/2$, to compute the off-diagonal elements. This leads to an estimate of $2N(N+5)$ flops to assemble the full inertia matrix (see Table III). It is assumed that the inertia matrix is then expressed as the product of two mutually adjoint triangular matrices. This typically requires $N^3/3$ flops (using Cholesky decomposition, for instance). Then, N^2 flops are assumed to be needed to obtain the solution of two triangular systems of linear equations. The combination of the foregoing three steps of inertia matrix assembly, decomposition as the product of two triangular matrices, and solution of two triangular set of equations is estimated to require a total $N^3/3 + N^2 + 2N(N+5)$ flops.

A comparison between the above two operation count estimates is shown in Fig. 4. The number of operations is shown for the two approaches, as a function of the number of links. Notice that the filtering and smoothing approach requires more computations than the other method for a small number of links. This is to be expected because there is an inherent overhead (Riccati equation, outward coordinate transformations, etc.) required to set up the filtering/smoothing computations. The benefits, in terms of operation count, of the filtering and smoothing approach become apparent for large N. For $N \geq 5$, the filtering and smoothing approach requires fewer computations.

The above comparison is not intended to be definitive. Understanding of the numerical properties of the filtering and

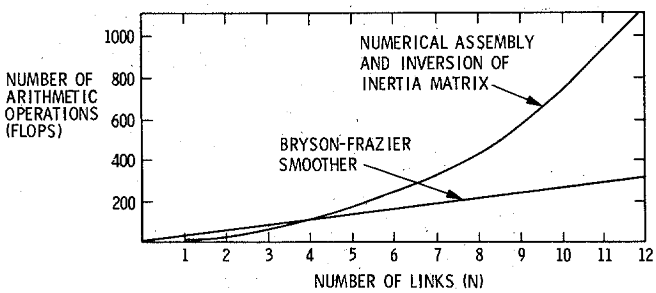


Fig. 4. Operation count for two methods.

smoothing approach is in its infancy. Much work is therefore required before a more complete comparison can be made. The above initial comparison, however, provides motivation required to more completely explore the potential of the approach.

XIII. Relationship to Other Work

The basic reference on filtering is, of course, Kalman's original paper [1], which derives the filter for discrete-time systems with discrete data and which, in addition, introduces a global framework (Riccati equation, Kalman gain, prediction/correction, covariances, etc.) that underlies much of today's linear filtering and prediction theory. A similarly basic reference for smoothing is [2]. A summary exposition of both filtering and smoothing, as well as the sweep method for solution of two-point boundary-value problems is provided in [3]. The main contribution of the present paper is to recognize that these filtering and smoothing techniques provide a unified framework to solve recursively the fundamental robotics problems of inverse and forward dynamics. This complements many of the recursive and nonrecursive techniques currently used to solve these problems [6]–[13].

Because they address forward dynamics (instead of the more common problems of inverse dynamics), [6], [12], [13] are very close in spirit to the present paper. In fact, the recursive equations of [6] are very similar to the filtering and smoothing solutions of Section VIII. The solutions advanced here expand on the results of [6] in two areas: 1) recognizing similarities with filtering and smoothing, and 2) providing what is believed to be a more appropriate way to account for the bias spatial forces and accelerations due to coriolis, centrifugal, gyroscopic, and gravitational effects. Reference [6] suggests that these effects be accounted for by conducting an inverse dynamics computation prior to the forward dynamics solution. This has the possible drawback of requiring that certain calculations (link-to-link coordinate transformations, spatial force and acceleration propagation, etc.) be performed twice: once for inverse dynamics and again for the forward dynamics problem. Hence two full recursions along the entire span of the manipulator appear to be required. In contrast, the recursive techniques advanced here embody these effects in the bias terms $b(k)$ and $n(k)$ of the filter and smoother equations. No advance inverse dynamics solution is required, and a single inward/outward iteration is sufficient to solve the problem. An additional contribution of the present paper is to introduce a framework that, in addition to solving the forward dynamics problem of [6], also provides inverse dynamics solutions.

Another result which is believed to be unique is the closed-form evaluation of the inertia matrix and its inverse in terms of estimation error covariances. This result suggests that numerical inertia matrix inversion can be avoided (or at least performed recursively). This can be done if the emphasis is placed instead on direct matrix-symbolic evaluation of the inertia matrix inverse (as in Section X of the paper) or on the filtering and smoothing formulas, which provide a constructive procedure for determining joint accelerations from applied moments.

Many of the works [7]–[11] presenting recursive solutions focus primarily on the inverse dynamics problem. These recursive methods lead either to the evaluation of required joint moments from desired joint angle accelerations or to evaluation of an inertia matrix for an equation of the form (22). The forward dynamics problem is not addressed directly. Instead, the usual approach requires a numerical inversion of the inertia matrix. This causes the resulting forward dynamics algorithms to be $O(N^3)$, i.e., the number of computations is proportional to the cube of the number of links. This means that for large N the computations required may be dominated by the matrix inversion process.

Yet another point of view with regards to robot dynamics is that initiated by [14], which advances the notion that explicit scalar equations of motion can be obtained for common manipulators such as the JPL/Stanford and PUMA arms. These equations are explicit in the sense that the scalar elements of the inertia matrix (as well as other terms accounting for coriolis, centrifugal, and other effects) are evaluated symbolically in terms of link mass and inertia, mass center offsets, etc. The end results of this approach are algebraic expressions [15], [16] for each of the inertia matrix elements.

Such explicit equations can lead to substantial computational savings. One key reason for this is that terms in the inertia matrix which do not depand on the instantaneous value of the joint angles (reflecting the manipulator configuration) can be grouped together and simplified. These terms need be evaluated only once at the beginning of the model application. The same value of those terms is then retained after this initialization. This is a feature that less explicit equations do not have. However, because of the complexity of the trigonometric and algebraic operations required, manual derivation methods cannot be used easily, and symbolic manipulation programs [15], [16] that conduct machine differentiation of the Lagrangian are typically used. One of the challenges that remains after symbolic evaluation of the inertia matrix elements is the numerical inertia matrix inversion required to solve the forward dynamics problem.

The recursive equations developed in Section X can, in principle, be used to arrive at direct explicit evaluation of the scalar elements of the inertia matrix inverse. A symbolic manipulation program could be set up to conduct the operations in (83) symbolically, as opposed to numerically. The end result would be a set of equations of the form (69) where the accelerations $a^1(k)$, $a^2(k)$, and $a^3(k)$ would be determined as explicit functions of the joint angles, the link masses and inertias, the link dimensions, etc. Such results would eliminate the need to invert the inertia matrix numerically, and could lead to significant computational savings. Savings comparable to those achieved in [15], [16] for explicit evaluation of the inertia matrix could be achieved for a similarly explicit evaluation of its inverse.

XIV. Concluding Remarks and Future Directions

The primary objective of the present paper has been to point out the equivalence between recursive robot dynamics methods and the filtering and smoothing techniques from state estimation theory. In the view of the author, establishing relationships between ideas and concepts that had been previously thought of as being unrelated is one of the more interesting efforts that can be made. This typically leads to the discovery of new physical and mathematical insights that would otherwise be very difficult to discover. An interesting example of this is the equivalence of spatial inertia and covariance. Another example, which is currently under investigation and which will be reported in the near future, is the relationship between manipulator redundancy (having more than six joints) and the notions of observability and controllability arising in state estimation and control theory.

A closely related objective has been to show that the robot arm dynamics computations can be organized with the very well understood and highly developed framework initially introduced in [1], [2]. Extensive analytical and computational experience exists with such an architecture. The architecture is very easy to understand both mathematically and physically. This is one of the primary reasons for its popularity. In contrast, robot dynamics solutions are usually not as easily understood because they are not cast within an architecture that is as recognizable. In addition, standardized software [17] is widely available to implement the filtering and smoothing techniques. The results of this paper make it relatively easy to use this software to set up dynamical models for simulation and control design for arbitrary robot arms.

The paper does not claim to advance techniques that are numerically superior to the existing ones. Rather, one of the results is to show that the existing methods can be derived and analyzed within the framework of estimation theory. This leads to a better understanding of the methods. In specific cases, the recursions outlined here can be shown to be equivalent to very efficient computational methods. This point is illustrated for example in Section VII by showing that the inverse dynamics solutions presented here are equivalent to the composite rigid body method advanced in [11] for its efficiency. The point is further illustrated in Section X by showing that the forward dynamics approach of [6] can be viewed as an extension of this method.

Although no particular attempt is made to advance the filtering and smoothing solutions for their numerical efficiency, these techniques could ultimately lead to better numerical algorithms for robot dynamics. The Kalman filter is quite popular, but it is not always the fastest recursive algorithm for state estimation. A wide variety of "fast" algorithms exist [17], [18] that are faster and more stable numerically than the Kalman filter. These algorithms however are also rooted in estimation theory. It would be of interest to investigate the computational improvements that may result from application of fast algorithms to the recursive robot dynamics problem. While such an investigation is not within the scope of this paper, the necessary analytical foundation has been developed so that such an investigation can be conducted.

The approach outlined here has been extended to rigid multibody configurations more general than the serial manipulator analyzed in this paper [19], [20]. Recursive dynamics solutions to systems of multiple joint-connected rigid bodies forming a topological tree are provided in [19]. Closed-chain systems arising in the problem of robotic dual-arm dynamics are analyzed in [20]. This is a step in the right direction. Future research might lead to extensions to more general and possibly flexible multibody systems.

The results of this paper suggest several areas for future research: 1) development of methods for symbolic evaluation of the scalar elements in the inverse of the inertia matrix, as opposed to the current ones that focus on the elements of the inertia matrix itself, to simplify system simulation as well as control design; 2) extensive numerical studies with the proposed methods to establish the same level of confidence as exists for current methods; and 3) development of forward and inverse dynamics solutions based on "fast" filtering and smoothing techniques which involve direct propagation of the filter gain as opposed to indirect methods requiring covariance propagation. A full investigation of these areas will require much work and will be quite interesting.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Trans. J. Basic Eng.*, vol. D, pp. 35–45, Mar. 1960.

[2] A. E. Bryson, Jr., and M. Frazier, "Smoothing for linear and nonlinear dynamic systems," in *Proc. Optimum Sys. Synthesis Conf.*, U.S. Air Force Tech. Rep. ASD-TDR-63-119, Feb. 1963.

[3] A. E. Bryson, Jr., and Y. C. Ho, *Applied Optimal Control.* Blaisdell, 1969.

[4] G. Rodriguez, "Kalman filtering, smoothing and recursive robot arm forward and inverse dynamics," Jet Propulsion Lab., Pasadena, CA, JPL Publication 86-48, Dec. 1986.

[5] G. Rodriguez, "Spatially random models, estimation theory and robot arm dynamics," in *Proc. JPL Workshop Space Telerobotics,* Jet Propulsion Lab., Pasadena, CA, JPL Publication 87-32, July 1987.

[6] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. Robotics Res.*, vol. 2, no. 1, Spring 1983.

[7] J. M. Hollerbach, "A recursive lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-10, pp. 730–736, Nov. 1980.

[8] M. W. Walker and D. E. Orin, "Efficient dynamics computer simulation of robotic mechanisms," *J. Dynamic Syst., Meas., Contr.*, vol. 104, pp. 205–211, Sept. 1982.

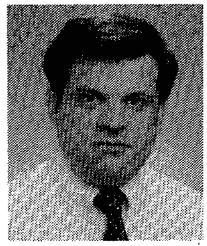
[9] J. Y. S. Luh *et al.*, "On-line computational scheme for mechanical manipulators," *J. Dynamic Syst., Meas., Contr.*, vol. 102, pp. 103–110, June 1980.

[10] W. M. Silver, "On the equivalence of Lagrangian and Newton–Euler dynamics for manipulators," *Int. J. Robot. Res.*, vol. 1, no. 2, pp. 118–128, Summer 1982.

[11] C. G. S. Lee and P. R. Chang, "Efficient parallel algorithms for robot inverse dynamics computations," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, 1986.

[12] ——, "Efficient algorithms for robot forward dynamics computation," in *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC, Mar. 1987.

[13] W. W. Armstrong, "Recursive solution to the equations of motion for an N-link manipulator," in *Proc. 5th World Congress Theory of Machines and Mechanisms,* vol. 2, Montreal, PQ, Canada, 1979.
[14] T. Kane and D. Levinson, "The use of Kane's equations in robotics," *Int. J. Robotics Res.,* vol. 2, no. 3, Fall 1983.
[15] J. W. Burdick, "An algorithm for generation of efficient manipulator dynamic equations," in *Proc. IEEE Int. Conf. Robotics and Automation,* San Francisco, CA, Apr. 1986.
[16] ——, "The explicit model and inertial parameters of the PUMA 560 arm," in *Proc. IEEE Int. Conf. Robotics and Automation,* San Francisco, CA, Apr. 1986.
[17] G. Bierman, *Factorization Methods for Discrete Sequential Estimation.* New York: Academic, 1977.
[18] B. Friedlander *et al.,* "Levinson and Chandrasekhar equations for a general discrete-time estimation problem," *IEEE Trans. Automat. Contr.,* vol. AC-23, 1978.
[19] G. Rodriguez, "Recursive multirigid body topological tree dynamics via Kalman filtering and Bryson–Frazier smoothing," presented at the *6th VPI/SU Symp. Control of Large Structures,* Blacksburg, VA, June 1987.
[20] ——, "Recursive forward dynamics for two robot arms in a closed chain based on Kalman filtering and Bryson–Frazier smoothing," in *Robotics,* M. Jamshidi, Ed. Amsterdam, The Netherlands: North-Holland, 1986.



Guillermo Rodriguez received the Ph.D. degree in control theory and applied mathematics from the University of California at Los Angeles in 1974.

Since then, he has been at the Jet Propulsion Laboratory. In the late 1970's, he participated in the development of on-board guidance and control systems flown in unmanned autonomous spacecraft for planetary missions. These included the completed Viking mission to Mars, the on-going Voyager mission to Jupiter, Saturn, Uranus and Neptune, and the planned Galileo mission to Jupiter. From 1979 to 1985, he supervised a group responsible for conducting research and technology development in the area of control of future flexible spacecraft and space systems (such as space station). He is currently supervisor of a group responsible for research in machine intelligence with applications to space telerobotics. His research interests include estimation theory, dynamics and control of flexible nonlinear multibody systems, and control architectures for autonomous robots.