

JPL PUBLICATION 86-48

IN-63-CR

52978

P.47

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

G. Rodriguez

(NASA-CR-180135) KALMAN FILTERING,
SMOOTHING AND RECURSIVE ROBOT ARM FORWARD
AND INVERSE DYNAMICS (Jet Propulsion Lab.)

47 p

CSCD 09B

N87-17451

Unclas

G3/63

43690

December 1, 1986

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL PUBLICATION 86-48

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

G. Rodriguez

December 1, 1986

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

The inverse and forward dynamics problems for multi-link 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 identical 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. In addition to providing techniques to compute joint accelerations from applied joint moments (and vice versa), the report provides an approach to evaluate recursively the composite multi-link system inertia matrix and its inverse. The report lays the foundation for the potential use of filtering and smoothing techniques in robot inverse and forward dynamics and in robot control design.

TABLE OF CONTENTS

1. INTRODUCTION.	1
2. CONFIGURATION AND PROBLEM STATEMENT	6
3. SPATIAL FORCE, ACCELERATION AND INERTIA	8
4. DYNAMICS: AN INWARD RECURSION FOR THE JOINT SPATIAL FORCES	11
5. KINEMATICS: AN OUTWARD RECURSION FOR THE JOINT VELOCITIES AND ACCELERATIONS	15
6. TWO-POINT BOUNDARY VALUE PROBLEM.	18
7. INVERSE DYNAMICS SOLUTION.	20
8. FORWARD DYNAMICS SOLUTION	23
9. SIMILARITIES TO KALMAN FILTER AND BRYSON-FRAZIER SMOOTHER	27
10. CLOSED FORM INERTIA MATRIX INVERSE	31
Solution of the State Equation	32
Solution of the Co-State Equation	33
Joint Angle Acceleration due to Joint Moments, Bias Spatial Forces, and Bias Spatial Accelerations.	35
11. RELATIONSHIP TO OTHER WORK.	36
12. CONCLUDING REMARKS AND FUTURE DIRECTIONS	39
REFERENCES.	40

Figures

2.1 N-Link Joint-Connected Mechanical System.	7
5.1 Inertial Locations of Joints k and $k-1$ and Link k Mass Center.	16
7.1 Illustration of Double Summation Order Reversal.	22

Table

6.1 Equivalence Between Two-Point Boundary-Value Problems in Optimal Estimation and in Recursive Robot Dynamics.	19
--	----

1. INTRODUCTION

The central theme of this report is to examine the use of filtering and smoothing techniques in studying robot dynamics. In particular, the report 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 discrete-time state space systems, can be used to solve the problems of serial manipulator inverse and forward dynamics. 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 solutions obtained are recursive in the sense that an inward recursion, which starts from the tip of the manipulator and proceeds sequentially from link to link to the base, is used to compute a sequence of constraint forces and moments. Similarly, an outward iteration from the base to the tip is used to determine a corresponding sequence of link/joint linear and angular accelerations. The recursive solutions are $O(N)$ in the sense that the number of required computations only grows linearly with the number of links.

The notions of spatial force, acceleration and inertia [4] are used to simplify the statement of these recursive equations. A spatial force acting on a link is defined here as a 6-dimensional vector whose first three components represent a 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 6-dimensional vector formed by an angular

acceleration and a linear acceleration. An additional concept introduced is that of spatial inertia. For any given link, the spatial inertia used here is a 6-by-6 matrix which very compactly embodies the mass and inertia properties of the link about its inner joint. It should be pointed out that there are minor differences between the definitions for spatial force, acceleration and inertia used here and those of [4].

One of the important steps in the report 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 [1]. The spatial force plays the role of the state. The link spatial interval, defined as the vector from the inner to the outer joint of a link, plays the role of the time interval between discrete time samples. This establishes a means to "propagate" the spatial force inwardly within a link from the outer 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 propagation mechanisms allows a complete link-to-link sequential propagation of the spatial force from the tip of the manipulator to its base. The difference equation generates the joint moments as an output. The transformation from spatial forces to joint moments is a projection operator that takes the 6-dimensional spatial force into the scalar that characterizes the applied joint 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 accelerations as inputs. The spatial accelerations play the role of the co-states (or adjoint variables) that are typical in optimal control and estimation problems [3]. This co-state equation reflects the kinematic relationship that exists between the spatial (angular and linear) acceleration at the outer joint of a link and the acceleration at its inner joint. Thus, the equation computes the spatial acceleration of a link at its outer joint given the acceleration of its inner joint. The difference equation can therefore be used to "propagate" in an outward direction the spatial accelerations within a link. A similar outward propagation across the joint at the interface between two adjacent links is obtained because the angular acceleration along the joint axis introduces a "jump" discontinuity in the angular acceleration component of the co-state (spatial acceleration) vector.

When combined, the above state and co-state 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 co-state vanishes at the base. These conditions arise because of the assumptions that the tip is unconstrained and the base is immobile (undergoes no accelerations). This boundary-value problem defined in terms of state and co-state variables is used as a pivotal step to develop the recursive inverse and forward dynamic 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 to the tip to obtain a set of co-states (spatial accelerations), using the set of prescribed joint accelerations and the boundary condition at the manipulator base; 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) and the required applied joint moments.

Still within the context of inverse dynamics, it is possible to use the above two-stage process to obtain the by now traditional dynamical equations for an N-link manipulator, expressed in terms of an N-by-N composite system inertia matrix. To arrive at this equation requires that the two difference equations of the two-point boundary-value problem be solved symbolically instead of numerically. In particular, the solution of the state equation is obtained in terms of a weighting pattern (or kernel) dependent on the transition matrix inherent in the state difference equation. Similarly, the co-state difference equation is solved in terms of the transpose of the transition matrix. Substitution of the solution for the co-state into that of the state leads to the desired form of the equations of motion. An interesting by-product of this process is a method for recursive computation of the inertia matrix itself by means of an inward iteration from the tip of the manipulator to its base. This recursive relationship for the inertia matrix is equivalent to those that describe the evolution of the covariance of the state of a linear discrete-time state space system that is driven by a white-noise process. With this result, the similarities between the statistical state 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.

Solution of the forward dynamics problem is also based on using 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 λ_k denote the state and co-state for link k. The symbol z_k denotes a 6-dimensional vector which turns out to play the role of the predicted state estimate whose evolution is described by the Kalman filter. The applied joint moments play the role of the measurement process. Similarly, P_k is a 6-by-6 matrix, with the units of spatial inertia, analogous to the predicted state estimation error covariance. This substitution, central to the "sweep method" referred to in [3],

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 as the difference between the actual and the predicted joint moment; 2) outward smoothing in which the innovations process resulting from the first stage is used to generate a sequence of co-states (spatial accelerations) and the desired joint 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 this correction step, the updated state estimate is obtained as a linear weighted combination of the predicted estimate and the innovations process. The weight associated with the innovations is a 6-by-1 matrix playing the role of the Kalman gain. This gain can be computed from the matrix P_k . The matrix P_k satisfies a difference equation analogous to the Riccati equation encountered in discrete-time systems. Propagation of this matrix from the outer to the inner joint of each link is achieved by using the system transition matrix. The spatial inertia associated with a link, playing the role of the process error covariance, 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.

The second stage is an outward recursion that aims to compute a sequence of co-states (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 by means of what are referred to as the Bryson-Frazier equations. These equations also have a predictor/corrector architecture. Prediction

occurs in outward propagation of the co-states within a link from the inner joint to the outer joint. Correction occurs in propagating the co-state from one link to the next across a joint.

If the same two-stage filtering/smoothing process is conducted symbolically (instead of numerically), it is possible to arrive at a closed-form expression for the inverse of the inertia matrix. To arrive at this equation, it is first necessary to solve the filtering equation for the predicted state estimate z_k in terms of the transition matrix associated with the Kalman filter. Similarly, the smoother equations for the co-states can be solved symbolically in terms of the transpose of this transition matrix. Substitution of the solution for the states into the solution for the co-states leads to the desired equation of motion. An interesting by-product of the above process is that it is possible to obtain recursive relationships for direct, non-numerical evaluation of the inertia matrix inverse. The recursive relationships are identical to those necessary to compute the covariance of the smoothed state estimate 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 co-states.

The remaining sections of the report 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, similarities to the Kalman filter and the Bryson-Frazier smoother, closed form inversion of the inertia matrix, relationship to other work, and concluding remarks.

2. CONFIGURATION AND PROBLEM STATEMENT

Consider a mechanical system of N links numbered $1, \dots, N$ connected together by N joints numbered $1, \dots, N$ to form a branch-free kinematic chain. The system is illustrated in Fig. 2.1.

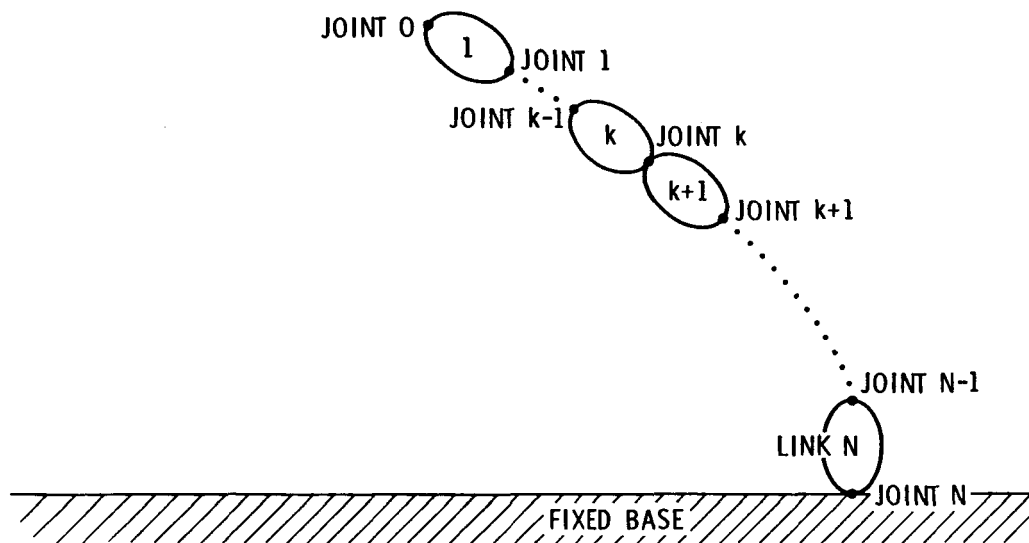


Fig. 2.1 N-Link Joint-Connected Mechanical System

The links and joints are numbered in an increasing order that goes from the tip of the system toward the base. Joint N is the last in the sequence, and it connects link N to an immobile 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. 2.1 allows a simpler description of the recursive algorithms contained in the report.

Let link k be characterized by an inertia tensor I_k about joint k, a mass m_k , a vector L_k from joint k to k-1, and a vector p_k from joint k to the link k mass center.

Let joint k be characterized by a unit vector h_k along its axis of rotation. Let τ_k be the active moment applied about the axis of joint k. Let u_k be the corresponding joint angle which is positive in the right hand sense about h_k .

The objective is to outline a recursive method for computation of the joint accelerations \ddot{u}_k , given the values of I_k , m_k , L_k , p_k , h_k and τ_k . A secondary objective is to solve the closely related inverse problem of computing τ_k from the desired accelerations \ddot{u}_k .

3. 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. The definitions used here are closely related but not identical to those of [4]. Generally, the term spatial force for a given link k will be used here to refer to a 6-by-1 vector whose first three components are pure moments and whose last three components are forces. Similarly, the term spatial acceleration will be used to describe a 6-by-1 vector consisting of three angular acceleration and three linear acceleration components. The link k spatial inertia is a 6-by-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-by-1 vectors representing, respectively, the constraint moment and force acting on link $k+1$ at joint k . The spatial force x_k^+ is the 6-by-1 composite vector defined by $x_k^+ = [-T_k^+ | -f_k^+]$, where the "+" superscript indicates that the corresponding variable is evaluated at a point on link $k+1$ that is immediately adjacent and on the "positive" side, toward the base, of joint k .
- T_k^- and f_k^- are 3-by-1 vectors representing, respectively, the constraint moment and force acting on link k at joint k . The spatial force x_k^- is the 6-by-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, toward the tip, 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.

- ω_k^+ and v_k^+ are 3-by-1 vectors representing, respectively, the angular and linear velocity of link k+1 at joint k. The corresponding spatial velocity is defined as $V_k^+ = [\omega_k^+ | v_k^+]$. It is assumed that both the angular and linear velocities associated with a link are specified in a coordinate frame attached to the link.
- ω_k^- and v_k^- are 3-by-1 vectors representing, respectively, the angular and linear velocity of link k at joint k. The corresponding spatial velocity is $V_k^- = [\omega_k^- | v_k^-]$. The "-" superscript indicates that the corresponding variable is to be evaluated on the negative side of joint k.
- ω_k and v_k are 3-by-1 vectors representing, respectively, the angular and linear velocity of link k at its mass center.
- $D^2 R_k^+ / Dt^2$ is a 3-by-1 vector representing the linear inertial acceleration of link k+1 at joint k, where the operator $D(\cdot)/Dt$ denotes time differentiation in an inertial reference frame. Similarly, $D^2 R_k^- / Dt^2$ is the acceleration of link k at joint k, and $D^2 R_k / Dt^2$ is the acceleration of the link k mass center.

Subsequently, it will be convenient to express these inertial accelerations in a local coordinate frame attached to link k. To this end, observe that $v_k = DR_k / Dt$, the linear velocity of the link k mass center, is a 3-by-1 vector expressed in link k coordinates. Hence,

$$D^2 R_k / Dt^2 = \dot{v}_k + \omega_k \times v_k$$

where $\dot{v}_k = dv_k / dt$ denotes that the time derivative of the velocity is performed in a coordinate frame attached to link k. Similar relationships can be obtained for the accelerations $D^2 R_k^+ / Dt^2$ and $D^2 R_k^- / Dt^2$. This allows the following definition of the notion of spatial acceleration.

- The spatial acceleration $\lambda_k^- = [\dot{\omega}_k^- | \dot{v}_k^-]$ is a 6-by-1 vector of angular and linear accelerations of link k at the negative side of joint k. $\lambda_k^+ = [\dot{\omega}_k^+ | \dot{v}_k^+]$ is a vector of accelerations of link k+1 on the positive side of joint k. $\lambda_k = [\dot{\omega}_k | \dot{v}_k]$ is the acceleration of the link k mass center. These accelerations are expressed in link k coordinates.

The spatial inertia matrix M_k for link k is defined as

$$M_k = \begin{bmatrix} I_k & m_k \tilde{p}_k \\ -m_k \tilde{p}_k & m_k U \end{bmatrix} \quad (3.1)$$

where I_k is the inertia dyadic 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-by-3 matrix equivalent to the cross-product operation $p_k \times$; and U is the 3-by-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 T^k associated with link k can be expressed as

$$T^k = 1/2 V_k^- \cdot M_k \cdot V_k^- \quad (3.2)$$

where V_k^- is the 6-by-1 vector of link k spatial velocities at joint k. This follows because

$$T^k = 1/2 \begin{bmatrix} \omega_k & v_k \end{bmatrix} \begin{bmatrix} \tilde{I}_k & 0 \\ 0 & m_k U \end{bmatrix} \begin{bmatrix} \omega_k \\ v_k \end{bmatrix} \quad (3.3)$$

where \tilde{I}_k is the link k inertia about its mass center; ω_k is the link k angular velocity; and v_k is the link k mass center linear velocity. However,

$$\begin{bmatrix} \omega_k \\ v_k \end{bmatrix} = \begin{bmatrix} U & 0 \\ -\tilde{p}_k & U \end{bmatrix} \begin{bmatrix} \omega_k^- \\ v_k^- \end{bmatrix} \quad (3.4)$$

where ω_k^- and v_k^- are, respectively, the angular and linear velocities of link k at joint k. Finally, substitute (3.4) in (3.3) and observe that $V_k^- = [\omega_k^- \ v_k^-]$ to obtain (3.2).

For later reference, it is also convenient to define the following 6-by-6 matrix

$$\Phi_{k,m} = \begin{bmatrix} U & \tilde{L}_{k,m} \\ 0 & U \end{bmatrix} \quad (3.5)$$

where $\tilde{L}_{k,m}$ is the 3-by-3 matrix equivalent to $L_{k,m} \times$; and $L_{k,m}$ is the vector from joint m to joint k. This matrix has the following properties usually associated with a "transition" matrix for a discrete linear state space system [1]:

$$\Phi_{k,m} = \Phi_{k,i} \Phi_{i,m}; \quad \Phi_{k,k} = U; \quad \Phi_{k,m}^{-1} = \Phi_{m,k} \quad (3.6)$$

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 can be inverted by reversing its two arguments.

4. DYNAMICS: AN INWARD RECURSION FOR THE JOINT SPATIAL FORCES

The main objective of this section is to establish that the spatial forces x_k satisfy

$$\mathbf{x}_k^- = \Phi_{k,k-1} \mathbf{x}_{k-1}^+ + M_k \lambda_k^- + \xi_k, \quad k = 1, \dots, N \quad (4.1)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^-, \quad k = 1, \dots, N \quad (4.2)$$

$$\mathbf{x}_0^+ = 0 \quad (4.3)$$

where λ_k^- is the 6-by-1 vector of joint k spatial accelerations and

$$\xi_k = \begin{bmatrix} \omega_k \times L_k \cdot \omega_k + m_k p_k \times (\omega_k \times v_k^-) - T_k - p_k \times f_k \\ m_k \omega_k \times v_k^- - f_k \end{bmatrix} \quad (4.4)$$

The arguments required to establish this result can be summarized as follows. Equation (4.1) is based on the rigid-body equations of rotational and translational motion for link k . Equation (4.2) reflects the equivalence of action/reaction moments and forces at joint k . Equation (4.3) states that the initial joint 0, which can be specified at any location on link 1, is not under the influence of any external moments and forces. A more detailed proof of this result is outlined below.

Proof. Consider the equations of rotational motion for link k about its mass center

$$\tilde{I}_k \cdot \dot{\omega}_k + \omega_k \times \tilde{I}_k \cdot \omega_k = T_k^- + T_{k-1}^+ + T_k + (L_k - p_k) \times f_{k-1}^+ - p_k \times f_k^- \quad (4.5)$$

where ω_k is the link k angular velocity; \tilde{I}_k is the link inertia about its mass center; T_k^- and f_k^- are the moment and force acting at joint k ; T_{k-1}^+ and f_{k-1}^+ are the moment and force acting at joint $k-1$; and T_k and f_k are the moment and force acting at the link mass center.

The translation of the link k mass center is described by

$$\mathbf{f}_k^- = m_k (\dot{\mathbf{v}}_k + \boldsymbol{\omega}_k \times \mathbf{v}_k) - \mathbf{f}_{k-1}^+ - \mathbf{f}_k \quad (4.6)$$

where $\dot{\mathbf{v}}_k + \boldsymbol{\omega}_k \times \mathbf{v}_k$ is the inertial acceleration of the mass center. Substitution of this on the right side of (4.5) leads to

$$\tilde{\mathbf{I}}_k \cdot \dot{\boldsymbol{\omega}}_k + \boldsymbol{\omega}_k \times \tilde{\mathbf{I}}_k \cdot \boldsymbol{\omega}_k + m_k \mathbf{p}_k \times (\dot{\mathbf{v}}_k + \boldsymbol{\omega}_k \times \mathbf{v}_k) = \quad (4.7)$$

$$\mathbf{T}_k^- + \mathbf{T}_{k-1}^+ + \mathbf{L}_k \times \mathbf{f}_{k-1}^+ + \mathbf{T}_k + \mathbf{p}_k \times \mathbf{f}_k$$

However, recall that $\mathbf{v}_k = \mathbf{v}_k^- + \boldsymbol{\omega}_k \times \mathbf{p}_k$. Therefore, $\dot{\mathbf{v}}_k + \boldsymbol{\omega}_k \times \mathbf{v}_k = \dot{\mathbf{v}}_k^- + \boldsymbol{\omega}_k \times \mathbf{v}_k^- + \dot{\boldsymbol{\omega}}_k \times \mathbf{p}_k + \boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \mathbf{p}_k)$, and

$$\dot{\mathbf{v}}_k = \dot{\mathbf{v}}_k^- + \boldsymbol{\omega}_k \times \mathbf{p}_k \quad (4.8)$$

Substitution of (4.8) in (4.7) and use of well-known cross-product expansion identities lead to

$$\begin{aligned} \mathbf{T}_k^- = & -\mathbf{T}_{k-1}^+ - \mathbf{L}_k \times \mathbf{f}_{k-1}^+ + \mathbf{I}_k \cdot \dot{\boldsymbol{\omega}}_k + m_k \mathbf{p}_k \times \dot{\mathbf{v}}_k^- + \\ & \boldsymbol{\omega}_k \times \mathbf{I}_k \cdot \boldsymbol{\omega}_k + m_k \mathbf{p}_k \times (\boldsymbol{\omega}_k \times \mathbf{v}_k^-) - \mathbf{T}_k - \mathbf{p}_k \times \mathbf{f}_k \end{aligned} \quad (4.9)$$

where $\mathbf{I}_k = \tilde{\mathbf{I}}_k + m_k [(p_k \cdot p_k) \mathbf{U} - p_k p_k]$ is the link k inertia about joint k .

Now, substitute (4.8) in (4.6) to obtain

$$\mathbf{f}_k^- = -\mathbf{f}_{k-1}^+ + m_k \boldsymbol{\omega}_k \times \mathbf{p}_k + m_k \mathbf{v}_k^- + m_k \boldsymbol{\omega}_k \times \mathbf{v}_k^- - \mathbf{f}_k \quad (4.10)$$

Equations (4.9) and (4.10) combine into

$$\begin{bmatrix} \mathbf{T}_k^- \\ \mathbf{f}_k^- \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \tilde{\mathbf{L}}_k \\ 0 & \mathbf{U} \end{bmatrix} \begin{bmatrix} -\mathbf{T}_{k-1}^+ \\ -\mathbf{f}_{k-1}^+ \end{bmatrix} + \begin{bmatrix} \mathbf{L}_k & m_k \tilde{\mathbf{p}}_k \\ -m_k \tilde{\mathbf{p}}_k & m_k \mathbf{U} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}}_k^- \\ \dot{\mathbf{v}}_k^- \end{bmatrix} + \begin{bmatrix} \boldsymbol{\omega}_k \times \mathbf{L}_k \cdot \boldsymbol{\omega}_k + m_k \tilde{\mathbf{p}}_k \times (\boldsymbol{\omega}_k \times \mathbf{v}_k^-) - \mathbf{T}_k - \mathbf{p}_k \times \mathbf{f}_k \\ m_k \boldsymbol{\omega}_k \times \mathbf{v}_k^- - \mathbf{f}_k \end{bmatrix} \quad (4.11)$$

Recall that $\mathbf{x}_k^- = [\mathbf{T}_k^- | \mathbf{f}_k^-]$, $\mathbf{x}_{k-1}^+ = [-\mathbf{T}_{k-1}^+ | -\mathbf{f}_{k-1}^+]$ and $\lambda_k^- = [\dot{\boldsymbol{\omega}}_k^- | \dot{\mathbf{v}}_k^-]$ to show that (4.11) is a more detailed version of (4.1).

To establish Eq. (4.2), it is enough to observe that $\mathbf{x}_k^- = \mathbf{x}_k^+$ implies (and is implied by)

$$\mathbf{T}_k^- = -\mathbf{T}_k^+; \quad \mathbf{f}_k^- = -\mathbf{f}_k^+ \quad (4.12)$$

which state that the moment \mathbf{T}_k^- and force \mathbf{f}_k^- acting on link k at joint k are equal to the negatives of the reaction moment \mathbf{T}_k^+ and force \mathbf{f}_k^+ acting on link $k+1$ at joint k .

Finally, Eq. (4.3) reflects the assumed lack of constraints (due to loads, for example) at the tip of the system on link 1.

5. KINEMATICS: AN OUTWARD RECURSION FOR THE JOINT VELOCITIES AND ACCELERATIONS

The sequence of spatial velocities satisfy

$$V_{k-1}^+ = \Phi_{k,k-1}^T V_k^-, \quad k=N, \dots, 1 \quad (5.1)$$

$$V_k^- = V_k^+ + H_k^T \dot{u}_k, \quad k=N, \dots, 1 \quad (5.2)$$

$$V_N^+ = 0 \quad (5.3)$$

where V_k^- and V_{k-1}^+ are, respectively, the spatial velocities of link k at joints k and $k-1$; $H_k = [h_k^T \ 0]$ is a 1-by-6 vector based on the unit vector h_k along the joint axis; and \dot{u}_k is the relative angular velocity at joint k .

The accelerations satisfy the closely related recursion

$$\lambda_{k-1}^+ = \Phi_{k,k-1}^T \lambda_k^- \quad (5.4)$$

$$\lambda_k^- = \lambda_k^+ + H_k^T \ddot{u}_k + \eta_k, \quad \lambda_N^+ = 0 \quad (5.5)$$

where η_k is the "bias" acceleration

$$\eta_k = \begin{bmatrix} \omega_k^+ \times h_k & \dot{u}_k \\ v_k^+ \times h_k & \dot{u}_k \end{bmatrix} \quad (5.6)$$

Proof: Consider the location of the joints k and $k-1$ with respect to an inertial reference, as illustrated in Fig. 5.1.

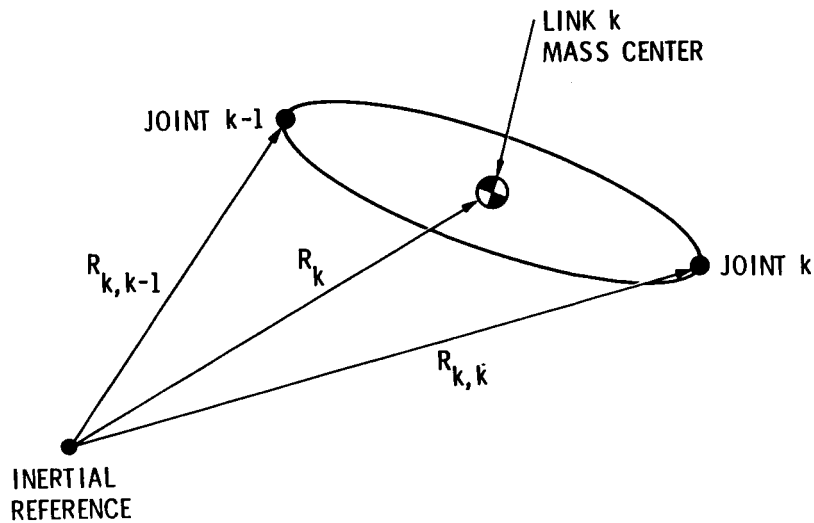


Fig. 5.1 Inertial Locations of Joints k and k-1 and Link k Mass Center

Observe that

$$\mathbf{R}_{k-1}^+ = \mathbf{R}_k^- + \mathbf{L}_k \quad (5.7)$$

where \mathbf{L}_k is the vector from joint k to k-1. Hence,

$$\mathbf{v}_{k-1}^+ = \mathbf{v}_k^- + \boldsymbol{\omega}_k^- \times \mathbf{L}_k \quad (5.8)$$

where $\mathbf{v}_{k-1}^+ = D\mathbf{R}_{k-1}^+/Dt$ and $\mathbf{v}_k^- = D\mathbf{R}_k^-/Dt$ are the inertial velocities of the two joints.

In addition, the angular velocity of link k at the two joints is common, since the link is assumed to be rigid. Hence,

$$\begin{bmatrix} \boldsymbol{\omega}_{k-1}^+ \\ \mathbf{v}_{k-1}^+ \end{bmatrix} = \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ -\tilde{\mathbf{L}}_k & \mathbf{U} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_k^- \\ \mathbf{v}_k^- \end{bmatrix} \quad (5.9)$$

which is a more detailed version of (5.1).

To establish (5.2), observe that

$$\omega_k^- = \omega_k^+ + h_k \dot{u}_k \quad (5.10)$$

$$v_k^- = v_k^+ \quad (5.11)$$

which state that the linear velocities on both sides of joint k are equal to each other and that the relative angular velocity between links k and $k+1$ equals the joint rotation $h_k \dot{u}_k$.

The boundary condition (5.3) is based on the assumption that the manipulator is attached to an immobile base.

The recursive relationships (5.4) - (5.6) for the accelerations can be established by appropriate time differentiation of (5.1) - (5.3), or of the equivalent equations (5.8), (5.10) and (5.11). Observe that (5.8) implies

$$\begin{aligned} \dot{v}_{k-1}^+ + \omega_{k-1}^+ \times v_{k-1}^+ &= \dot{v}_k^- + \omega_k^- \times v_k^- + \\ &\omega_k^- \times L_k + \omega_k^- \times (\omega_k^- \times L_k) \end{aligned} \quad (5.12)$$

However, (5.8) and $\omega_k^- = \omega_{k-1}^+$ imply

$$\dot{v}_{k-1}^+ = \dot{v}_k^- + \omega_k^- \times L_k \quad (5.13)$$

This establishes (5.4). Similarly (5.10) and (5.11) imply

$$\dot{\omega}_k^- = \dot{\omega}_k^+ + h_k \ddot{u}_k + \omega_k^+ \times h_k \dot{u}_k \quad (5.14)$$

$$\dot{v}_k^- + \omega_k^- \times v_k^- = \dot{v}_k^+ + \omega_k^+ \times v_k^+ \quad (5.15)$$

Use of (5.10) and (5.11) in (5.15) leads to

$$\dot{\mathbf{v}}_k^- = \dot{\mathbf{v}}_k^+ + \mathbf{v}_k^+ \times \mathbf{h}_k \dot{\mathbf{u}}_k \quad (5.16)$$

Equations (5.14) and (5.16) together imply (5.5).

Finally, the boundary condition $\lambda_N^+ = 0$ follows from (5.3).

6. TWO-POINT BOUNDARY-VALUE PROBLEM

The sequences of spatial forces \mathbf{x}_k and spatial accelerations λ_k satisfy the following two-point boundary-value problem:

$$\mathbf{x}_k^- = \Phi_{k,k-1} \mathbf{x}_{k-1}^+ + M_k \lambda_k^- + \xi_k \quad (6.1)$$

$$\mathbf{x}_k^+ = \mathbf{x}_k^- \quad (6.2)$$

$$\lambda_{k-1}^+ = \Phi_{k,k-1}^T \lambda_k^- \quad (6.3)$$

$$\lambda_k^- = \lambda_k^+ + H_k^T \ddot{\mathbf{u}}_k + \eta_k \quad (6.4)$$

$$\tau_k = H_k \mathbf{x}_k^+ \quad (6.5)$$

$$\mathbf{x}_0^+ = \lambda_N^+ = 0 \quad (6.6)$$

where τ_k is the active moment at joint k , and $\ddot{\mathbf{u}}_k$ are the corresponding joint accelerations.

The above is a two point boundary-value problem in the sense that the boundary conditions (6.6) are satisfied at the two points: the manipulator tip and its base.

Equations (6.1) and (6.2) are inward recursive equations which determine the sequence of spatial forces, given the spatial accelerations λ_k . The term ξ_k , defined by (4.4), appears as a bias term in this equation.

Equations (6.3) and (6.4) constitute an outward propagation and update that determines the sequence of spatial accelerations λ_k , given the joint accelerations u_k and the "bias" acceleration term η_k .

Equation (6.5) can be viewed as an output equation in the sense that it relates the state x_k and the output τ_k by means of the state-to-output map H_k .

This two-point boundary-value problem is analogous to those encountered in optimal control and estimation theory for linear systems based on quadratic criteria [3]. Such problems have been investigated extensively to develop filtering and smoothing solutions for dynamical systems. The equivalence between (6.1) - (6.6) and the problems in optimal control and estimation can be outlined by means of Table 6.1.

TABLE 6.1

Equivalence Between Two-Point Boundary-Value Problems in Optimal Estimation and in Recursive Robot Dynamics

ESTIMATION		ROBOT DYNAMICS
States	x_k	Spatial Forces
Co-States	λ_k	Spatial Accelerations
Measurements	τ_k	Active Moments
Transition Matrix	$\phi_{k,k-1}$	Spatial Transition Matrix
Process Error Covariance	M_k	Spatial Inertia Matrix
Known Deterministic Input	ξ_k	Bias Spatial Force
State to Output Map	H_k	Projection from State to Joint Axis

A more complete investigation of this equivalence is contained in Section 9.

The above problem can be used to solve the following two closely related problems: obtain the moment sequence τ_k , given knowledge of the joint accelerations \ddot{u}_k ; obtain the joint accelerations \ddot{u}_k from knowledge of the active moments τ_k . These are referred to, respectively, as the inverse and forward dynamics problems and are solved in the following two sections.

7. INVERSE DYNAMICS SOLUTION

The solution to the inverse dynamics problem consists of a two-stage process of outward recursion based on the co-state difference equation followed by an inward recursion based on the state equation.

The first stage involves an outward sequential process to determine a sequence of spatial accelerations. This outward recursion is based on Eqs. (6.3) and (6.4) and assumes that the spatial bias accelerations η_k have been previously determined from the spatial and joint velocities V_k and \dot{u}_k in (5.1) - (5.3). Equation (6.4) describes an operation by which the spatial acceleration λ_k^+ on the positive side of joint k, the joint acceleration \ddot{u}_k , and the bias acceleration η_k are combined in order to obtain the updated spatial acceleration λ_k^- at the negative side of joint k. Although this has not been explicitly shown, in practice, in going from one link to the next, a coordinate transformation is typically performed that converts the spatial accelerations into the coordinate frame of the next link. The updated acceleration is then operated on as in Eq. (6.3) by the matrix $\phi_{k,k-1}^T$ to arrive at the spatial acceleration λ_{k-1}^+ at joint k-1. These two steps, consisting of update at the joints and propagation from the inner to the outer joint of a link, provide a sequential process that generates the spatial accelerations λ_k . This process is started with the terminal condition $\lambda_N^+ = 0$.

The second stage in the inverse dynamics solution involves an inward sequential process to generate the spatial forces and the required joint moments. This second stage is based on Equation (6.1) and (6.2).

Equation (6.1) involves propagation of the spatial force at the outer joint of a link to the inner joint by using the spatial accelerations made available by the first stage and the bias term ξ_k . Equation (6.2) expresses continuity of the spatial force in going across a joint between two adjacent links. The process starts with the boundary condition $x_0^+ = 0$, indicating that the initial (and fictitious joint) is unconstrained in motion. The process continues inward from the tip to the base until the full sequence x_k^+ of spatial forces has been generated. The active moments τ_k at the joints are obtained as an output of this process by means of the state-to-output transformation H_k .

The boundary-value problem (6.1) - (6.6) can also be used to arrive at the traditional second-order matrix equation

$$M(u) \ddot{u} + V(u, \dot{u}) = \tau \quad (7.1)$$

where $M(u)$ is the inertia matrix, and $V(u, \dot{u})$ is a term that comprises nonlinear velocity and gravity dependent effects. To this end, observe that the co-state equations (6.3) and (6.4) imply

$$\lambda_j^- = \sum_{i=j}^N \phi_{i,j}^T (H_i^T \ddot{u}_i + \eta_i) \quad (7.2)$$

Similarly, the state equations (6.1) and (6.2) imply

$$x_k^- = \sum_{j=1}^k \phi_{k,j} (M_j \lambda_j^- + \xi_j) \quad (7.3)$$

Substitution of (7.2) in (7.3) leads to

$$x_k^- = \sum_{j=1}^k \sum_{i=j}^N \phi_{k,j} M_j \phi_{i,j}^T (H_i^T \ddot{u}_i + \eta_i) + \sum_{j=1}^k \phi_{k,j} \xi_j \quad (7.4)$$

However, observe the identity

$$\sum_{j=1}^k \sum_{i=j}^N = \sum_{i=1}^N \sum_{j=1}^{\min(i,k)} \quad (7.5)$$

which can be established by inspection of Fig. 7.1.

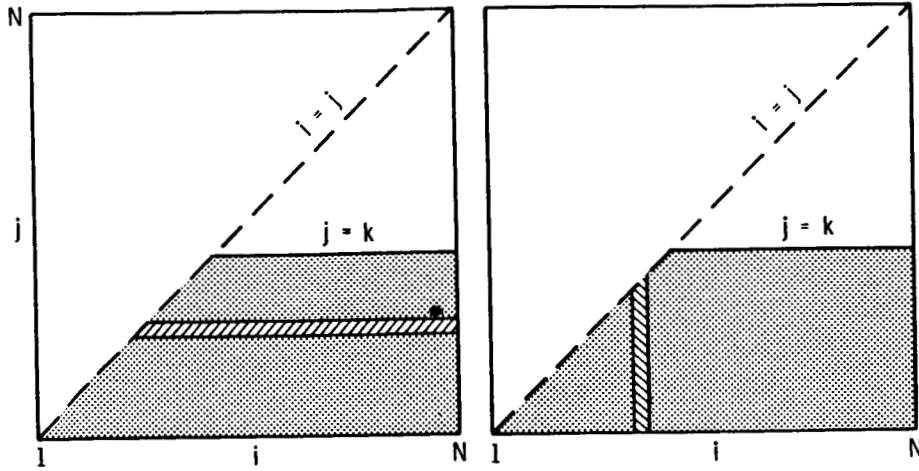


Fig. 7.1 Illustration of Double Summation Order Reversal

Use of this in (7.4) implies

$$\bar{x}_k = \sum_{i=1}^N r_{k,i} (H_i^T \ddot{u}_i + \eta_i) + \sum_{i=1}^k \phi_{k,i} \xi_i \quad (7.6)$$

where $r_{k,i}$ is the 6-by-6 matrix

$$r_{k,i} = \sum_{j=1}^{\min(i,k)} \phi_{k,j} M_j \phi_{i,j}^T \quad (7.7)$$

Note that $r_{k,i}$ can be specified as

	$1 \leq i \leq k \leq N$	$N \geq i \geq k \geq 1$
$r_{k,i}$	$\phi_{k,i} r_{i,i}$	$r_{k,k} \phi_{i,k}^T$

where

$$r_{k,k} = \sum_{j=1}^k \phi_{k,j} M_j \phi_{k,j}^T \quad (7.8)$$

Observe also that $r_{k,k}$ satisfies the recursive relationship

$$r_{k+1,k+1} = \phi_{k+1,k} r_{k,k} \phi_{k+1,k}^T + M_{k+1} \quad (7.9)$$

This formula, in filtering and prediction theory, is that satisfied by the covariance of the state of a linear discrete-time system subject to a process error with covariance M_k [3].

Use $\tau_k = H_k x_k^-$ in (7.6) to obtain

$$\sum_{i=1}^N m_{k,i} \ddot{u}_i + V^k(u, \dot{u}) = \tau_k, \quad k = 1, \dots, N \quad (7.10)$$

where

$$m_{k,i} = H_k r_{k,i} H_i^T, \quad V^k = \sum_{i=1}^N H_k r_{k,i} \eta_i + \sum_{i=1}^k H_k \phi_{k,i} \xi_i \quad (7.11)$$

This is the scalar version of the matrix equation (7.1), where $V = [V^1, \dots, V^N]$; M is the N -by- N inertia matrix whose general element $m_{k,i}$ is specified by (7.11); and $\tau = [\tau_1, \dots, \tau_N]$ is the vector of all active joint moments.

8. FORWARD DYNAMICS SOLUTION

The forward dynamics problem is to find the accelerations \ddot{u}_k at the joints, given the active moments τ_k . This problem can be solved using (6.1) - (6.6) and what is referred to as the "sweep method" in [3]. The sweep method begins by assuming that the state x_k and the co-state λ_k are related by

$$x_k = z_k + P_k \lambda_k \quad (8.1)$$

where z_k and P_k are to be determined by means of recursive formulas that emerge upon substitution of (8.1) in equations (6.1) - (6.6). In these formulas, 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 8.1 The joint accelerations \ddot{u}_k can be computed from the applied moments τ_k by means of the following two-stage process of inward filtering and outward smoothing.

Filtering

Initial Condition $z_0^+ = 0$ (8.2)

State Prediction $z_k^- = \Phi_{k,k-1} z_{k-1}^+ + \xi_k$ (8.3)

Spatial Inertia Prediction $P_k^- = \Phi_{k,k-1} P_{k-1}^+ \Phi_{k,k-1}^T + M_k$ (8.4)

Gain Computation $D_k = H_k P_k^- H_k^T, G_k = P_k^- H_k^T D_k^{-1}$ (8.5)

Innovations $e_k^- = \tau_k - H_k z_k^-$ (8.6)

State Update $z_k^+ = z_k^- + G_k e_k^- + P_k^+ \eta_k$ (8.7)

Residuals $e_k^+ = D_k^{-1} e_k^-$ (8.8)

Spatial Inertia Update $P_k^+ = P_k^- - P_k^- H_k^T D_k^{-1} H_k P_k^-$ (8.9)

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 links $1, \dots, k$.

Smoothing

$$\text{Terminal Condition} \quad \lambda_N^+ = 0 \quad (8.10)$$

$$\begin{array}{l} \text{Co-State} \\ \text{Propagation} \end{array} \quad \lambda_{k-1}^+ = \Phi_{k,k-1}^T \lambda_k^- \quad (8.11)$$

$$\text{Joint Accelerations} \quad \ddot{u}_k = e_k^+ - G_k^T (\lambda_k^+ + \eta_k) \quad (8.12)$$

$$\text{Co-State Update} \quad \lambda_k^- = \lambda_k^+ + \eta_k + H_k^T \ddot{u}_k = (I - G_k H_k)^T (\lambda_k^+ + \eta_k) + H_k^T e_k^+ \quad (8.13)$$

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 inertias have been updated in (8.7) and (8.9). A similar transformation from link $k+1$ to link k is performed after the co-state update in (8.13).

Proof: Substitute (8.1) in (6.5) to obtain

$$e_k^- = H_k P_k \lambda_k^- \quad (8.14)$$

where e_k^- is the innovations process defined by

$$e_k^- = \tau_k - H_k z_k^- \quad (8.15)$$

Now, substitute (6.4) in (8.14) to obtain

$$D_k \ddot{u}_k = e_k^- - H_k P_k^- (\lambda_k^+ + \eta_k) \quad (8.16)$$

where $D_k = H_k P_k^- H_k^T$. Hence,

$$\ddot{u}_k = e_k^+ - G_k^T (\lambda_k^+ + \eta_k) \quad (8.17)$$

where G_k is the Kalman gain $G_k = P_k^- H_k^T D_k^{-1}$. Substitute this in (6.4) to obtain

$$\lambda_k^- = (I - G_k H_k)^T (\lambda_k^+ + \eta_k) + H_k^T e_k^+ \quad (8.18)$$

which verifies (8.13). Substitute (8.1) in (6.1) to obtain

$$z_k^- + P_k^- \lambda_k^- = \Phi_{k,k-1} z_{k-1}^+ + \Phi_{k,k-1} P_{k-1}^+ \lambda_{k-1}^+ + M_k \lambda_k^- + \xi_k \quad (8.19)$$

However, in view of (6.3),

$$z_k^- + P_k^- \lambda_k^- = \Phi_{k,k-1} z_{k-1}^+ + \Phi_{k,k-1} P_{k-1}^+ \Phi_{k,k-1}^T \lambda_{k-1}^- + M_k \lambda_k^- + \xi_k \quad (8.20)$$

The state and spatial inertia propagation equations (8.3) and (8.4) are sufficient conditions to satisfy (8.20).

To obtain the state and inertia update equations, use (8.1) in the identity (6.2):

$$z_k^+ + P_k^+ \lambda_k^+ = z_k^- + P_k^- \lambda_k^- \quad (8.21)$$

However, substitute (8.18) on the right side of (8.21) to obtain

$$z_k^+ + P_k^+ \lambda_k^+ = z_k^- + P_k^- (I - G_k H_k)^T (\lambda_k^+ + \eta_k) + P_k^- H_k^T e_k^+ \quad (8.22)$$

and

$$P_k^+ = P_k^- - P_k^- H_k^T D_k^{-1} H_k P_k^- \quad (8.23)$$

$$z_k^+ = z_k^- + G_k e_k^- + P_k^+ \eta_k \quad (8.24)$$

These are the inertia and state update equations.

As an aside, note that the updated spatial inertia satisfies the following alternative formulas:

$$P_k^+ = (I - G_k H_k) P_k^- = P_k^- (I - G_k H_k)^T = (I - G_k H_k) P_k^- (I - G_k H_k)^T \quad (8.25)$$

as is well-known in Kalman filtering [1,3]. These two equations can be obtained routinely from (8.23).

9. SIMILARITIES TO KALMAN FILTER AND BRYSON-FRAZIER SMOOTHER

The two-point boundary-value problem of Section 6 and the filtering and smoothing equations of Section 8 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:

$$x_k = \phi_{k,k-1} x_{k-1} + w_k \quad (9.1)$$

$$\tau_k = H_k x_k \quad (9.2)$$

where x_k is the state; τ_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

$$E(w_k) = \xi_k; \quad E[(w_k - \xi_k)(w_k - \xi_k)^T] = M_k \quad (9.3)$$

where ξ_k is the link k bias spatial force, and M_k is the link k spatial inertia. To

simplify the discussion, it is assumed in this section that the acceleration bias term η_k has been set to zero.

The filtering problem consists of finding

$$z_k^- = E(x_k / \tau_1, \dots, \tau_{k-1}) \quad (9.4)$$

the best estimate of the state given all of the previous measurements $\tau_1, \dots, \tau_{k-1}$. Closely related to this filtered estimate is the innovations process defined by

$$e_k^- = \tau_k - H_k z_k^- \quad (9.5)$$

and the filtered state estimation error covariance

$$E[(x_k^- - z_k^-)(x_k^- - z_k^-)^T] = P_k^- \quad (9.6)$$

which is known to satisfy the discrete-time Riccati equation. The covariance of the innovations is also known to be

$$E[(e_k^-)(e_k^-)^T] = H_k P_k^- H_k^T = D_k \quad (9.7)$$

Note that (9.7) is obtained from the more general formula $D_k = H_k P_k^- H_k^T + R_k$ by setting the measurement noise covariance R_k to zero. The equations for the Kalman gain and the updated covariance are

$$G_k = P_k^- H_k^T D_k^{-1} \quad (9.8)$$

$$P_k^+ = P_k^- - P_k^- H_k D_k^{-1} H_k^T P_k^- \quad (9.9)$$

The updated state estimation error covariance P_k^+ can be shown to be

$$E[(x_k - z_k^+) (x_k - z_k^+)^T] = P_k^+ \quad (9.10)$$

where

$$z_k^+ = z_k^- + G_k e_k^- \quad (9.11)$$

is the updated state estimate

$$z_k^+ = E(x_k / \tau_1, \dots, \tau_k) \quad (9.12)$$

The smoothing problem associated with (9.1) and (9.2) is to find

$$\hat{x}_k = E(x_k / \tau_1, \dots, \tau_N) \quad (9.13)$$

the best estimate of the state, given all of the data τ_1, \dots, τ_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

$$\hat{x}_k = z_k + P_k \lambda_k \quad (9.14)$$

where λ_k are the co-states specified by

$$\lambda_k^+ = \Phi_{k+1,k}^T \lambda_{k+1}^- \quad (9.15)$$

$$\lambda_k^- = (I - G_k H_k)^T \lambda_k^+ + H_k^T e_k^+ \quad (9.16)$$

The state estimate error covariance S_k associated with the smoothed state estimate \hat{x}_k is defined as

$$S_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \quad (9.17)$$

This matrix is given by [3]

$$S_k = P_k - P_k \Lambda_k P_k \quad (9.18)$$

where Λ_k is the covariance of the co-states defined as

$$\Lambda_k = E(\lambda_k \lambda_k^T) \quad (9.19)$$

It is known also that Λ_k satisfies the recursive relationships

$$\Lambda_k^+ = \Phi_{k+1,k}^T \Lambda_{k+1}^- \Phi_{k+1,k} \quad (9.20)$$

$$\Lambda_k^- = (I - G_k H_k)^T \Lambda_k^+ (I - G_k H_k) + H_k^T D_k^{-1} H_k \quad (9.21)$$

This is a backward recursion consisting of propagation in (9.20) followed by an update in (9.21). The boundary condition $\Lambda_N^+ = 0$ is valid at the N^{th} sample.

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 Λ_k above. This is done in the next section.

10. CLOSED FORM INERTIA MATRIX INVERSE

The central objective is to obtain the following equation:

$$\ddot{u}_k = \alpha_k^1 + \alpha_k^2 + \alpha_k^3 \quad (10.1)$$

where α_k^1 , α_k^2 and α_k^3 are the joint angle accelerations due to the applied moments τ_k , the bias spatial forces ξ_k , and the bias spatial accelerations η_k . The three acceleration components are given by

$$\alpha_k^1 = c_k \tau_k + d_k \sum_{i=1}^{k-1} \psi(k^-, i^+) G_i \tau_i + G_k^T \sum_{i=k+1}^N \psi^T(i^-, k^+) d_i^T \tau_i$$

$$\alpha_k^2 = d_k \xi_k + d_k \sum_{i=1}^{k-1} \psi(k^-, i^-) \xi_i + G_k^T \sum_{i=k+1}^N \psi^T(i^-, k^+) \Lambda_i^- \xi_i \quad (10.2)$$

$$\alpha_k^3 = G_k^T (\Lambda_k^+ P_k^+ - D) \eta_k + d_k \sum_{i=1}^{k-1} \psi(k^-, i^+) P_i^+ \eta_i + G_k^T \sum_{i=k+1}^N \psi^T(i^+, k^+) (\Lambda_i^+ P_i^+ - D) \eta_i$$

where c_k and d_k are, respectively, the scalar and the 1-by-6 vector

$$c_k = D_k^{-1} + G_k^T \Lambda_k^+ G_k; d_k = G_k^T \Lambda_k^+ - c_k H_k \quad (10.3)$$

and $\psi(k^-, i^-) = \psi(k^-, i^+) (I - G_i H_i)$ and $\psi(i^+, k^+) = (I - G_i H_i) \psi(i^-, k^+)$, with $\psi(i^-, k^+)$ defined in (10.6) below.

Recall that (7.1) implies that $\ddot{u} = M^{-1}(u) [\tau - V(u, \dot{u})]$, where M is the inertia matrix.

Hence, the elements of its inverse can be obtained by inspection of α_k^1 in (10.2).

The overall approach used to arrive at (10.1) is based on solving both the state and co-state difference equations in terms of their corresponding "weighting" kernels. Substitution of the co-state solution into the state solution leads to the desired results. This is now performed in detail.

Solution of the State Equation

The aim here is to show that the sequences of "predicted" spatial forces z_k^- and residuals e_k^+ are specified by:

$$z_k^- = \xi_k + \sum_{i=1}^{k-1} \psi(k^-, i^+) \tilde{\xi}_i \quad (10.4)$$

$$e_k^+ = D_k^{-1} [\tau_k - H_k \xi_k - H_k \sum_{i=1}^{k-1} \psi(k^-, i^+) \tilde{\xi}_i] \quad (10.5)$$

where $\psi(k^-, i^+)$ and $\tilde{\xi}_i$ are defined as

$$\psi(k^-, i^+) = \Phi_{k,k-1} [(I - G_{k-1} H_{k-1}) \Phi_{k-1,k-2} \dots (I - G_{i+1} H_{i+1}) \Phi_{i+1,i}] \quad (10.6)$$

$$\tilde{\xi}_i = (I - G_i H_i) \xi_i + G_i \tau_i + P_i^+ \eta_i \quad (10.7)$$

Note that $\psi(k^-, i^+)$ is the transition matrix for the Kalman filter. This matrix is evaluated at its two arguments k^- and i^+ , representing, respectively, the negative (outboard, toward the tip) side of joint k and the positive (inboard, toward the base) side of joint i .

Proof: Recall that

$$z_k^- = \xi_k + \Phi_{k,k-1} z_{k-1}^+ \quad (10.8)$$

$$z_k^+ = (I - G_k H_k) z_k^- + G_k \tau_k + P_k^+ \eta_k \quad (10.9)$$

Substitution of (10.8) in (10.9) leads to

$$z_k^+ = \psi(k^+, k-1^+) z_{k-1}^+ + \tilde{\xi}_k \quad (10.10)$$

where

$$\psi(k^+, k-1^+) = (I - G_k H_k) \phi_{k, k-1} \quad (10.11)$$

This implies that

$$z_{k-1}^+ = \sum_{i=1}^{k-1} \psi(k-1^+, i^+) \tilde{\xi}_i \quad (10.12)$$

where

$$\psi(k-1^+, i^+) = \psi(k-1^+, k-2^+) \dots \psi(i+1^+, i^+) \quad (10.13)$$

To obtain (10.4), substitute (10.12) in (10.8) and observe that $\psi(k^-, i^+) = \phi_{k, k-1} \psi(k-1^+, i^+)$. This and (8.8) imply (10.5).

Solution of the Co-State Equation

The aim here is to show that the sequence of spatial accelerations λ_k^+ is specified by

$$\lambda_k^+ = \sum_{i=k+1}^N \psi^T(i^-, k^+) \tilde{\eta}_i \quad (10.14)$$

where

$$\tilde{\eta}_i = (I - G_i H_i)^T \eta_i + H_i^T e_i^+ \quad (10.15)$$

To establish (10.14), begin by observing that

$$\lambda_k^- = (I - G_k H_k)^T (\lambda_k^+ + \eta_k) + H_k^T e_k^+ \quad (10.16)$$

$$\lambda_k^+ = \Phi_{k+1,k}^T \lambda_{k+1}^- \quad (10.17)$$

with the terminal condition $\lambda_N^+ = 0$. Substitution of (10.17) in (10.16) leads to

$$\lambda_k^- = \Psi^T(k+1^-, k^-) \lambda_{k+1}^- + \tilde{\eta}_k \quad (10.18)$$

where

$$\Psi(k+1^-, k^-) = \Phi_{k+1,k} (I - G_k H_k) \quad (10.19)$$

Hence,

$$\lambda_{k+1}^- = \sum_{i=k+1}^N \Psi^T(i^-, k+1^-) \tilde{\eta}_i \quad (10.20)$$

where

$$\Psi(i^-, k+1^-) = \Phi_{i,i-1} (I - G_{i-1} H_{i-1}) \dots \Phi_{k+2,k+1} (I - G_{k+1} H_{k+1}) \quad (10.21)$$

To obtain (10.14), substitute (10.20) in (10.17) and observe that $\Psi^T(i^-, k^+) = \Phi_{k+1,k}^T \Psi^T(i^-, k+1^-)$.

The joint accelerations \ddot{u}_k can be obtained by substitution of (10.14) in (8.12):

$$\ddot{u}_k = (e_k^+ - G_k^T \eta_k) - G_k^T \sum_{i=k+1}^N \Psi^T(i^-, k^+) \tilde{\eta}_i \quad (10.22)$$

Joint Angle Accelerations due to Joint Moments, Bias Spatial Forces, and Bias Spatial Accelerations

The objective here is to obtain (10.1). To this end, observe that (10.5) and (10.22) imply that

$$\ddot{u}_k = D_k^{-1} [\tilde{\tau}_k - H_k \sum_{i=1}^{k-1} \psi^{(k^-, i^+)} \tilde{\xi}_i - H_k P_k^- \sum_{j=k+1}^N \psi^{T(i^-, k^+)} \tilde{\eta}_j] \quad (10.23)$$

where $\tilde{\tau}_k$ is defined as

$$\tilde{\tau}_k = \tau_k - H_k \xi_k - H_k P_k^- \eta_k \quad (10.24)$$

However,

$$\sum_{i=k+1}^N \psi^{T(i^-, k^+)} \tilde{\eta}_i = \sum_{i=k+1}^N [\psi^{T(i^+, k^+)} \eta_i + \psi^{T(i^-, k^+)} H_i^T e_i^+] \quad (10.25)$$

where $\psi^{T(i^+, k^+)} = \psi^{T(i^-, k^+)} (I - G_i H_i)^T$. In view of (10.5),

$$\sum_{i=k+1}^N \psi^{T(i^-, k^+)} H_i^T e_i^+ = \sum_{i=k+1}^N \psi^{T(i^-, k^+)} H_i^T D_i^{-1} [\tau_{i^-} - H_i \xi_{i^-} - \sum_{j=1}^{i-1} H_i \psi^{(i^-, j^+)} \tilde{\xi}_j] \quad (10.26)$$

Recall the identity

$$\sum_{i=k+1}^N \sum_{j=1}^{i-1} = \sum_{j=1}^{N-1} \sum_{i=\max(k+1, j+1)}^{N-1} \quad (10.27)$$

to obtain

$$\sum_{i=k+1}^N \sum_{j=1}^{i-1} \psi^{T(i^-, k^+)} H_i^T D_i^{-1} H_i \psi^{(i^-, j^+)} \tilde{\xi}_j = \Lambda_k^+ \tilde{\xi}_{k^+}$$

$$\sum_{j=k+1}^{k-1} \Lambda_k^+ \psi(k^+, j^+) \tilde{\xi}_j + \sum_{j=1}^N \psi^T(j^+, k^+) \Lambda_j^+ \tilde{\xi}_j \quad (10.28)$$

where

$$\Lambda_k^+ = \sum_{i=k+1}^N \psi^T(i^-, k^+) H_i^T D_k^{-1} H_i \psi(i^-, k^+) \quad (10.29)$$

In arriving at the upper limit N of summation for the last term in (10.28), use has been made of the terminal condition $\Lambda_N^+ = 0$ implied by (10.29). Finally, use of (10.28) in (10.26), (10.26) in (10.25), and (10.25) in (10.23) leads to (10.1).

Observe that (10.29) implies that the sequence Λ_k satisfies the recursive equations

$$\Lambda_k^+ = \Phi_{k+1, k}^T \Lambda_{k+1}^- \Phi_{k+1, k}, \quad \Lambda_N^+ = 0 \quad (10.30)$$

$$\Lambda_k^- = (I - G_k H_k^T) \Lambda_k^+ (I - G_k H_k^T) + H_k^T D_k^{-1} H_k \quad (10.31)$$

These equations are identical to the ones satisfied by the co-state variable covariance of the fixed-time smoother in Section 9.

11. RELATIONSHIP TO OTHER WORK

The basic reference on filtering is, of course, Kalman's original report [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]. Reference [3] provides a summary exposition of both filtering and smoothing, as well as the sweep method for solution of two-point boundary-value problems. The main contribution of the present report 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 [4-10].

Because they address forward dynamics (instead of the more common problems of inverse dynamics), Refs. [4,5] are very close in spirit to the present report. In fact, the recursive equations of [4] are very similar to the filtering and smoothing solutions of Section 8. The solutions advanced here expand on the results of [4] 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 [4] 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 ξ_k and η_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 report is to introduce a framework that, in addition to solving the forward dynamics problem of [4], 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 10 of this report)

or in the filtering and smoothing formulas which provide a constructive procedure for determining joint accelerations from applied moments.

Many of the references [6-9] presenting recursive solutions focus primarily on the inverse dynamics problem. These recursive methods lead either to the evaluation of required joint applied moments from desired joint accelerations or to evaluation of an inertia matrix for an equation of the form (7.1). 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 [10], 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 matrices 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 [11,12] 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 depend on the instantaneous value of the joint angles (reflecting the manipulator configuration) can be grouped together and 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 [11,12] 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 10 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 Eq. (10.1) symbolically, as opposed to numerically. The end result would be a set of equations of the form (10.1) where the accelerations α_k^1 , α_k^2 and α_k^3 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 [11,12] for explicit evaluation of the inertia matrix could be achieved for a similarly explicit evaluation of its inverse.

12. CONCLUDING REMARKS AND FUTURE DIRECTIONS

The pivotal step presented in this report is that the dynamics and kinematics of an N-link serial manipulator can be described by a two-point (in space) boundary-value problem. This observation allows the solution of inverse and forward dynamics within a single unified framework based on recursive techniques from the theory of optimal filtering and smoothing.

The results of this report suggest several areas for future research.

- 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. This would simplify system simulation as well as implementation of increasingly important exact linearization techniques for control design [13, 14].

- Extensive numerical studies with the proposed methods to establish the same level of confidence as exists for current methods.
- 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.
- Use of the forward dynamics solutions presented here for control design.

A full investigation of these areas will require much work and will be quite interesting.

REFERENCES

1. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," ASME Transactions, Journal of Basic Engineering, Vol. D, pp. 35-45, March 1960.
2. Bryson, A. E., Jr., and M. Frazier, "Smoothing for Linear and Nonlinear Dynamic Systems," Proc. Optimum Sys. Synthesis Conf., U. S. Air Force Tech. Rept. ASD-TDR-63-119, Feb. 1963.
3. Bryson, A. E., Jr., and Y. C. Ho, Applied Optimal Control, Blaisdell Publishing Co., 1969.
4. Featherstone, R., "The Calculation of Robot Dynamics Using Articulated-Body Inertias," The International Journal of Robotics Research, Vol. 2, No. 1, Spring 1983.
5. Armstrong, W. W., "Recursive Solution to the Equations of Motion for an N-Link Manipulator," Proc. 5th World Congress on Theory of Machines and Mechanisms, Vol. 2, Montreal, 1979.
6. Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," IEEE Trans. Sys. Man Cyber., SMC-10(11), pp. 730-736, Nov. 1980.
7. Walker, M. W., and D. E. Orin, "Efficient Dynamics Computer Simulation of Robotic Mechanisms," Journal of Dynamic Systems, Measurement, and Control, Vol. 104, pp. 205-211, Sept. 1982.
8. Luh, J.Y.S., et al., "On-Line Computational Scheme for Mechanical Manipulators," Journal of Dynamic Systems, Measurement, and Control, Vol. 102, pp. 103-110, June 1980.
9. Silver, W. M., "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", The International Journal of Robotics Research, Vol.1, No. 2, pp. 118-128, Summer 1982.

10. Kane, T., and D. Levinson, "The Use of Kane's Equations in Robotics," The International Journal of Robotics Research, Vol. 2, No. 3, Fall 1983.
11. Burdick, J. W., "An Algorithm for Generation of Efficient Manipulator Dynamic Equations," Proc. IEEE International Conference on Robotics and Automation, San Francisco, CA, April 1986.
12. Burdick, J. W., "The Explicit Model and Inertial Parameters of the PUMA 560 Arm," Proc. IEEE International Conference on Robotics and Automation, San Francisco, CA, April 1986.
13. Tarn, T. J., et al., "Nonlinear Feedback in Robot Arm Control", Proc. 23rd IEEE Conf. on Decision and Control, Las Vegas, NV, Dec. 1984.
14. Isidori, A., et al., "Nonlinear Decoupling via Feedback: A Differential Geometric Approach", IEEE Trans. Aut. Control, AC-26, 1981.

1. Report No. 86-48	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics		5. Report Date December 1, 1986	6. Performing Organization Code
7. Author(s) G. Rodriguez		8. Performing Organization Report No.	
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91109		10. Work Unit No.	11. Contract or Grant No. NAS7-918
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546		13. Type of Report and Period Covered External Report JPL Publication	
15. Supplementary Notes		14. Sponsoring Agency Code RE159 BK-549-01-11-05-00	
16. Abstract The inverse and forward dynamics problems for multi-link 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 identical 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. In addition to providing techniques to compute joint accelerations from applied joint moments (and vice versa), the report provides an approach to evaluate recursively the composite multi-link system inertia matrix and its inverse. The report lays the foundation for the potential use of filtering and smoothing techniques in robot inverse and forward dynamics and in robot control design.			
17. Key Words (Selected by Author(s)) Computer Systems; Cybernetics; Numerical Analysis; Theoretical Mathematics		18. Distribution Statement Unclassified; unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price