A New Algorithmic Framework for Robot Dynamics Analysis with Application to Space Robots Dynamics Simulation

Amir Fijany and Antal K. Bejczy
Jet Propulsion Laboratory, California Institute of Technology

Abstract

In this paper a new algorithm for fast serial and parallel dynamics simulation of space robots as characterized by serial chain systems with a floating base is presented. This algorithm is derived by using a recently developed algorithmic framework based on a new *Schur Complement* factorization of the inverse of mass matrix, *M-l*. The new algorithm leads to optimal serial and parallel computation for the problem, that is, an $O(N)$ serial computation and an $O(Log\ N)$ parallel computation by using O(N) processors.

Key Words: Manipulator Dynamics, *Dynamics* Sire-, *ulation, Space Robots, Parallel Computation*

I. Introduction

The manipulator forward dynamics, or dynamics simulation, problem concerns the determination of the motion of the system which results from the application of a set of control forces. The problem is of paramount importance in the robotics field where simulation is a fundamental part of the analysis of manipulator systems. There is moreover considerable motivation for devising fast simulation algorithms, ranging from the need for extensive off-line simulation capabilities for design and evaluation purposes, to real-time implementation for controlled simulations and teleoperator training.

Robotic technology offers significant benefits for space exploration both for unmanned missions (e.g., Mars exploration) in the form of space teleoperation and manned mission for which it can reduce the risk to and increase the productivity of astronauts [1]. However, in order to realize these benefits certain technical problems need to be addressed. A major problem for space teleoperation is the unavoidable delay in information transfer. In the presence of such a delay, a faster-than-real-time dynamics simulation capability is extremely valuable since it allows a human operator to preview several scenarios before run-time [2]. A second problem arises in both manned and unmanned applications wherein a robot is mounted on a non-fixed or a floating-base system, such as space station, space shuttle, or satellite, since in this case there is a dynamic interaction between robot and the floating-base. In order to design efficient control schemes, accurate and fast algorithms are needed to analyze this interaction and to simulate the behavior of both base and robot.

In brief mathematical terms, the forward dynamics problem can be stated as the solution of a linear system

$$\mathcal{M}\ddot{Q} = \Gamma - b(Q, \dot{Q} = \mathcal{F}_T \qquad (1.1)$$

or

$$\ddot{Q} = \mathcal{M}^{-1}\mathcal{F}_T \qquad (2.1)$$

The vector $b(Q, \dot{Q})$ represents the contribution of nonlinear terms and can be computed by using the Newton-Euler (N-E) algorithm [3] while setting $\ddot{Q}$ to zero. The vector $\mathcal{F}_T = \mathrm{Col}\{F_{Ti}\}\varepsilon\Re^N$ in (1.1) represents the acceleration-dependent component of the control forces.

At present, it appears that the development of serial algorithms has reached a certain level of maturity. The O(N) algorithms [4,5,6] represent the asymptotically optimal serial solution for the problem. However, with the mat urity of serial algorithms, it is clear that any further significant improvement in the computational efficiency can only be achieved through exploitation of parallelism. Extensive analysis of efficiency of the existing algorithms for parallel computation has been reported in [2,7]. This analysis showed that the existing $O(N)$ algorithms are strictly serial, that is, parallelism in their computation is bounded. As a result, their parallelization leads to yet $O(N)$ parallel algorithms which are faster than their serial counterparts by only a small constant factor.

Motivated by this analysis, we have recently developed a new formulation for the problem, designated as the Constraint Force (CF) algorithm, based on a novel factorization of *M-'* in form of

Schur Complement [7]. This factorization indeed provides an algorithmic framework for the development of fast serial $O(N)$ algorithms for various dynamics problems. More importantly, however, it allows for the first time the development of both time- and processor-optimal parallel algorithms, that is, $O(Log\,N)$ parallel algorithms with $O(N)$ processors. In addition to application to serial chain systems [7], the *Schur Complement* factorization has also been applied for the control and simulation of closed-loop systems [8]. In this paper, we present the application of this new algorithmic framework to the dynamics simulation of space robots as characterized by serial chain systems with a floating base.

This paper is organized as follows. In §II, notation and some preliminaries are presented. A modified derivation of the CF algorithm for serial chain systems is briefly reviewed in §III. The purpose of this modification is to allow a more straightforward extension to floating base systems. The application of the CF algorithm to dynamics simulation of floating base systems is presented in §IV and its serial and parallel computational efficiency are also analyzed. Finally, some concluding remarks are made in §V.

II. Notation and Preliminaries

A. Spatial and Global Notation

In the following derivation, we use spatial and global notation, presented in the nomenclature section, which allow a compact representation of the algorithms. For the sake of clarity, the global quantities are shown by upper-case script letters. Here, only joints with one revolute DOF are considered here. However, the results can be extended to joints with different and/or more DOFS.

With any vector v, a matrix $\tilde{v}$ can be associated whose representation in any frame is a skew symmetric matrix:

$$\tilde{v} = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$$

where $v_=$, v_y, and $v_:$ are the components of v in the frame considered. The matrix $\tilde{v}$ has the properties that $\tilde{v}^t = -\tilde{v}$ and $\tilde{v}_1 v_2 = V_1$ x V_2, i.e., it is a vector cross-product operator. A matrix $\hat{v}$ associated to the vector v is defined as

$$\hat{v} = \begin{bmatrix} U & \tilde{v} \\ 0 & U \end{bmatrix} \quad \text{and} \quad \hat{v}^t = \begin{bmatrix} U & O \\ -\tilde{v} & U \end{bmatrix} \varepsilon \Re^{6\times 6} \quad (2.1)$$

where here (and through the rest of the paper) U and O stand for the identity and zero matrices of appropriate size. The spatial velocities of two rigidly connected points A and B are related as

$$V_A = \hat{P}^t_{A,B} V_B \quad (2.2)$$

where $P_{A,B}$ denotes the position vector from B to A. The matrix $\hat{P}_{A,B}$ has the properties as

$$\hat{P}_{A,B}\hat{P}_{B,C} = \hat{P}_{A,C} \text{ and } \hat{P}^{-1}_{A,B} = \hat{P}_{B,A} \quad (2.3)$$

which can be easily verified from Eq. (2.1). If the linear and angular velocities of point B are zero then

$$\dot{V}_A = \hat{P}^t_{A,B}\dot{V}_B \quad (2.4)$$

The spatial forces acting on points A and B are related as

$$F_B = \hat{P}_{A,B} F_A \quad (2.5)$$

The spatial inertia of body i about its center of mass, designated as $I_{i,Ci}$, is given by

$$I_{i,Ci} = \begin{bmatrix} J_i & 0 \\ 0 & m_i U \end{bmatrix} \varepsilon \Re^{6\times 6}$$

The spatial inertia of body i about point O_i, designated as Ii, is obtained as

$$I_i = \hat{S}_i I_{i,Ci} \hat{S}^t_i = \begin{bmatrix} k_i & \tilde{h}_i \\ \tilde{h}^t_i & m_i U \end{bmatrix} \quad (2.6)$$

Equation (2.6) represents the *parallel axis theorem* for propagation of spatial inertia.

A global bidiagonal block matrix $\mathcal{P}\varepsilon\Re^{6N\times 6N}$ is defined as

$$P= \begin{bmatrix} U & & & & \\ -\hat{P}_{N-1} & U & & 0 & \\ 0 & -\hat{P}_{N-2} & U & & \\ O & 0 & & & \\ \vdots & . & & & \\ O & O & & -\hat{P}_1 & u \end{bmatrix}$$

Note that, according to our notation, $P_{i+1,i} = Pi$.

B. Recursive Equations of Motion

From (1.1) the multibody system can be assumed as a system at rest which upon the application of $\mathcal{F}_T$ accelerates in space. Note that, (1.1) describes a global relationship between the set of active forces and the set of resulting accelerations. Alternatively, the equations of motion can be written in a recursive form by describing the propagation of spatial accelerations and forces among bodies of the serial chain as follows (Fig. 1).

$$\dot{V}_i = \hat{P}^t_{i-1}\dot{V}_{i-1} + H_i\ddot{Q}_i \quad (2.7)$$

$$F, = I_i\dot{V}_i + \hat{P}_i F_{i+1} \quad (2.8)$$

The acceleration-dependent acting forces, F_{T_i}, and the interbody forces, F_i, are related by

$$F_{T_i} = H_i^t F, \qquad (2.9)$$

Equations (2.7)-(2.9) are spatial representation of the simplified (with nonlinear terms being excluded) N-E algorithm.

III. Schur Complement Factorization of *M-l*

A. Interbody Force Decomposition Strategy

In this section we briefly review a recently developed factorization of M^{-1} [7] to establish the basis for its application to dynamics simulation of space robots. This new factorization is based on a rather unconventional decomposition of interbody force of the form:

$$F_i = H_i F_{Ti} + W_i F_{Si} \qquad (3.1)$$

where F_{Si} is the constraint force. The projection matrices *Hi* and W_i *are* taken to satisfy the following orthogonality conditions:

$$H_i^t H_i = U, W_i^t W_i = U, \; W_i^t H_i = O \qquad (3.2)$$

$$H_i H_i^t + W_i W_i^t = U \qquad (3.3)$$

The above assumes that the projection matrices are block diagonal in the rotational and translational coordinates. **This in turn implies that there is no** coupling between the degrees of freedom, thereby precluding dimensional inconsistency. Furthermore, the axes of articulation for each joint are orthogonal. For a joint i with multiple DOFs, say N_{fi} <6 DOFs, $H_i \varepsilon \Re^{6\times N_{fi}}$ and $W_i \varepsilon \Re^{6\times N_{ci}}$ with $N_{fi}+N_{ci} = 6$. It should be emphasized the CF algorithm can also be extended to more complex cases wherein the axes of articulation are not orthogonal [9,10].

B. Factorization of *M-l*

To derive the new factorization of *M-l*, let us first define following global matrix and vector for $i = N$ to 1:

$$\mathcal{W} = \text{Diag}\{W_i\}\varepsilon\Re^{6N\times 5N} \text{ and } \mathcal{F}_S = \text{Col}\{F_{Si}\}\varepsilon\Re^{5N}$$

Equations (3.1)-(3.3) can be now written in global form as

$$\mathcal{F} = \mathcal{H}\mathcal{F}_T + \mathcal{W}\mathcal{F}_S \qquad (3.4)$$

$$\mathcal{H}^t\mathcal{H} = U, \mathcal{W}^t\mathcal{W} = U, \textbf{ and } \mathcal{W}^t\mathcal{H} = O \qquad (3.5)$$

$$\mathcal{H}\mathcal{H}^t + \mathcal{W}\mathcal{W}^t = U \qquad (3.6)$$

Using matrices P and P^t, Eqs. (2.7)-(2.8) can be written **in a global** form as

$$\mathcal{P}^t\dot{\mathcal{V}} = \mathcal{H}\ddot{\mathcal{Q}} \qquad (3.7)$$

$$\mathcal{P}\mathcal{F} = \mathcal{I}\dot{\mathcal{V}} \qquad (3.8)$$

From (3.4) and (3.7) it follows that

$$\mathcal{W}^t\mathcal{P}^t\dot{\mathcal{V}} = 0 \qquad (3.9)$$

$$\mathcal{H}^t\mathcal{P}^t\dot{\mathcal{V}} = \ddot{\mathcal{Q}} \qquad (3.10)$$

From **(3.7)** and (3.4) we have

$$\dot{\mathcal{V}} = \mathcal{I}^{-1}\mathcal{P}\mathcal{F} = \mathcal{I}^{-1}\mathcal{P}(\mathcal{H}\mathcal{F}_T + \mathcal{W}\mathcal{F}_S) \qquad (3.11)$$

Substituting (3.11) into (3.9)-(3.10) we get

$$\mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W}\mathcal{F}_S + \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\mathcal{F}_T = O \qquad (3.12)$$

$$\mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W}\mathcal{F}_S + \mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\mathcal{F}_T = \ddot{\mathcal{Q}} \qquad (3.13)$$

where

$$A = \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{W}\varepsilon\Re^{5N\times 5N} \qquad (3.14)$$

$$B = \mathcal{W}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\varepsilon\Re^{5N\times N} \qquad (3.15)$$

are block diagonal matrices and

$$\mathcal{C} = \mathcal{H}^t\mathcal{P}^t\mathcal{I}^{-1}\mathcal{P}\mathcal{H}\varepsilon\Re^{N\times N} \qquad (3.16)$$

is a tridiagonal matrices. Furthermore, both matrices *A* and C are symmetric positive definite (see appendix). This garantees the existence of $\mathcal{A}^{-1}$. Equations (3.12)- (3.13) can now be written as

$$\mathcal{A}\mathcal{F}_S + \mathcal{B}\mathcal{F}_T = 0 \qquad (3.17)$$

$$\mathcal{B}^t\mathcal{F}_S + \mathcal{C}\mathcal{F}_T = \ddot{\mathcal{Q}} \qquad (3.18)$$

from which Q is obtained as

$$\ddot{\mathcal{Q}} = (C - \mathcal{B}^t\mathcal{A}^{-1}\mathcal{B})\mathcal{F}_T \qquad (3.19)$$

In compariosn with (1.2), a factorization of *M ˙1* in form of ***Schur*** *Complement is* then by

$$\mathcal{M}^{-1} = \mathcal{C} - \mathcal{B}^t\mathcal{A}^{-1}\mathcal{B} \qquad (3.20)$$

Efficient $O(N)$*serialand*O(Log *N)* parallel solution of (1.2) by using the factorization given by (3.20) is presented in detail in [7].

IV. Application of CF Algorithm to Dynamics Simulation of Space Robots

A. Extension of CF Algorithm to serial chain systems with Floating Base

A floating base (designated as body O) has six DOFS. This implies that $H_0 = U$ and $W_0 = O$. As a result, a direct application of the CF algorithm to the systems with a floating base is not possible since the resulting matrix A will be singular. However, it is possible to apply a modified version of the CF algorithm by treating the floating base separately as follows.

For a floating base, since $W_0 = \mathbf{O}$ then from (3. 1) we have $F_0 = F_{T0}$ and hence $F.$ is given. This additional information is used in modifying the CF algorithm. The spatial acceleration of floating base is given by

$$\dot{V}_0 = H_0\ddot{Q}_0 = \ddot{Q}_0 \qquad (4.1)$$

where $\ddot{Q}_0 \varepsilon \Re^6$ represents six DOFS of floating-base. Equation (2.8) for base (body O) is now written as

$$F. = I_o Q_o + \hat{P}_0 F_1 \qquad (4.2)$$

from which we have

$$\ddot{Q}_0 = I_0^{-1}(F_0 - \hat{P}_0 F_1) \qquad (4.3)$$

The spatial acceleration of body 1 is given by

$$\dot{V}_1 = \hat{P}_0^t \ddot{Q}_0 + H_1 \ddot{Q}_1 \qquad \mathbf{(4.4)}$$

For bodies N to 1, Eq. (3.7) can now be written as

$$\mathcal{P}^t \dot{\mathcal{V}} = \mathcal{H}\ddot{\mathcal{Q}} + \dot{\mathcal{V}}_b \qquad (4.5)$$

where $\dot{\mathcal{V}}_b = [0,0, \ldots 0, \ddot{Q}_0^t \hat{P}_0]^t$ **is** a bias acceleration term. The force propagation for bodies N to 1 is again given by *(3.8)*. Repeating the procedure of §III.A, given by (3.9)-(3.13) but by using (4.5) instead of (3.7), it then follows that

$$\mathcal{A}\mathcal{F}_S + \mathcal{B}\mathcal{F}_T = \mathcal{W}\dot{\mathcal{V}}_b \qquad (4.6)$$

$$\mathcal{B}^t\mathcal{F}_S + \mathcal{C}\mathcal{F}_T = \ddot{\mathcal{Q}} + \mathcal{H}\dot{\mathcal{V}}_b \qquad (4.7)$$

Using the decomposition in (3.1) for F_1 and from (4.3), the term $\mathcal{W}^t\dot{\mathcal{V}}_b$ after some manipulation is given by

$$\mathcal{W}^t\dot{\mathcal{V}}_b = -\mathcal{W}^t\mathcal{I}_b^{-1}\mathcal{W}\mathcal{F}_S + \dot{\mathcal{R}}_{T10} \qquad (4.8)$$

where

$$\mathcal{W}^t\mathcal{I}_b^{-1}\mathcal{W}\mathcal{F}_S = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ W_1^t \hat{P}_0^t I_0^{-1} \hat{P}_0 W_1 F_{S1} \end{bmatrix} \qquad \mathbf{(4.9)}$$

$$\dot{\mathcal{R}}_{T10} = \begin{bmatrix} 0 \\ 0 \\ \\ 0 \\ W_1^t \hat{P}_0^t I_0^{-1}(F. - \hat{P}_0 H_1 F_{T1}) \end{bmatrix} \qquad \mathbf{(4.10)}$$

and $\mathcal{I}_b^{-1} = \mathrm{Diag}\{0, O, \ldots \hat{P}_0^t I_0^{-1}$ fro}. Similarly, the term $\mathcal{H}^t\dot{\mathcal{V}}_b$ is given by

$$\mathcal{H}^t\dot{\mathcal{V}}_b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ H_1^t \hat{P}_0^t I_0^{-1}(F_0 - \hat{P}_0 F_1) \end{bmatrix} \qquad (4.11)$$

Substituting (4.8)-(4.10) into Eq. (4.6) gives

$$\mathcal{A}'\mathcal{F}_S + \mathcal{B}\mathcal{F}_T = \dot{\mathcal{R}}_{T10} \qquad (4.12)$$

where

$$\mathcal{A}' = \mathcal{A} + \mathcal{W}^t\mathcal{I}_b^{-1}\mathcal{W} \qquad (4.13)$$

The matrix A' is a rank one modification (in block sense) of A. More precisely, A' and A differ only in their last element, that is, $A_1 = W_1^t I_1^{-1} W_1$ whereas $A_1' = W_1^t(I_1^{-1} + \hat{P}_0^t I_0^{-1}\hat{P})W_1$. The matrix A_1' is symmetric and, as shown in the appendix, it is also positive definite and hence invertible.

B. Serial and Parallel Computational Complexity of CF Algorithm

Thus far, our derivation has been presented in a coordinate-free form. However, before the implementation of the algorithm, the tensors and vectors involved in the computation should be projected onto a suitable frame. The choice of optimal frame for the algorithm is discussed in detail in **[7]**. Here, suffice to mention that all required projections can be performed in $O(N)$ for serial computation and in $O(Log\,N)$ with $O(N)$ processors for parallel computation. Also, for efficient serial and parallel computation, the matrices B and $\mathcal{C}$ need not be computed explicitly and only the explicit computation of matrix A is needed. Multiplication of any vector by matrices B and C can be performed by using their factorization in terms of simpler operators given by (3.15)-(3.16).

A step-by-step description of the algorithm for simulation of space robots is then given as follows.

Step 1. Compute the $\mathcal{F}_T$ for i = N to O by using the N-E algorithm [3].

Step 2. Compute $\dot{\mathcal{R}}_{T10}$ from (4.10) and *B3T by* using (3.15).

Step **3.** Form matrix $\mathcal{A}'$ from (3.14) and (4.13) and solve $\mathcal{A}'\mathcal{F}_S = -\mathcal{B}\mathcal{F}_T + \dot{\mathcal{R}}_{T10}$ for $\mathcal{F}_S$.

Step 4. Compute the terms $\mathcal{H}^t\dot{\mathcal{V}}_b$ from (4.11), $\mathcal{B}^t\mathcal{F}_S$ by **using (3.15),** and $\mathcal{C}\mathcal{F}_T$ by using **(3.16).**

Step 5. Compute $\ddot{\mathcal{Q}}$ from (4.7) and $\ddot{Q}_0$ from (4.3).

1. Serial $O(N)$ *Computation*

By using the N-E algorithm the cost of step 1 is of O(N). The computation of $\dot{\mathcal{R}}_{T10}$ from (4. 10) can be performed in O(1) and the cost of computation of $\mathcal{B}\mathcal{F}_T$ by using (3.15) is of O(N). The matrix *A* ' can be formed from (3.14) and (4.13) with a cost of O(N) and $\mathcal{F}_S$ is obtained by solving a block tridiagonal system with a cost of $O(N)$. The cost of computation of $\mathcal{H}\dot{\mathcal{V}}_b$ from (4.11) is of O(l). The cost of computation of $\mathcal{B}^t\mathcal{F}_S$ from (3.15) and $\mathcal{C}\mathcal{F}_T$ from (3.16) is of O(N). Finally, the computation of $\ddot{\mathcal{Q}}$ from (4.7) and $\ddot{Q}_0$ from (4.3) can be performed with a cost of O(N) and 0(1), respectively. Therefore, the overall cost of serial implementation of the algorithm is of O(N). “

2. *Parallel* $O(Log\, N)$ *Computation*

The computation of the N-E algorithm can be performed with a cost of $O(Log$ N) by using O(N) processors [11]. The computation of $\dot{\mathcal{R}}_{T10}$ can be performed in O(1) on one processor. The computation of $\mathcal{B}\mathcal{F}_T$ can be also performed in $O(1)$ by using O(N) processors. The matrix *A'* can be formed in O(1) with O(N) processors and the block tridiagonal system solution for $\mathcal{F}_S$ can be performed in $O(Log\, N)$ with O(N) processors. The term $\mathcal{H}^t\dot{\mathcal{V}}_b$ can be computed in O(1) on one processor and the computation of $\mathcal{B}^t\mathcal{F}_S$ and $\mathcal{C}\mathcal{F}_T$ both can be performed in O(1) with O(N) processors. The vectors $\ddot{\mathcal{Q}}$ and $\ddot{Q}_0$ can be both computed *in O(1)* with O(N) and one processor, respectively. This implies an overall parallel complexity of $O(Log\, N)$ by using O(N) processors for the algorithm.

V. Conclusion

In this paper we presented a new algorithm for fast and optimal serial and parallel dynamics simulation of space robots as characterized by serial chain systems with a floating base. Due to the lack of space, our discussion was mainly focused on the mathematical foundation and computational complexity of the algorithm. It should be mentioned, however, that as for the serial chain systems [7,8] the use of *Schur Complement* factorization not only leads to fast serial and parallel algorithms but it also provides a deeper physical insight into the structure of computation. Such an insight can lead to better understanding of the dynamic interaction between floating base (e.g., space station, space shuttle, or satellite) and the robot arm and thus the design of more appropriate control schemes.

It should be also emphasized that the parallel algorithm presented in this paper is also highly efficient for practical implementation. In fact, the **practical implementation of the** CF algorithm on MIMD parallel architectures for a serial chain (fixed-base) system has shown that a significant speedup in the computation can be achieved [13]. The parallel algorithm of this paper has properties (in terms of **both** computation and communication) very similar to the algorithm in [13] and hence it should be also highly efficient for practical parallel implementation.

ACKNOWLEDGMENT

The research described **in this** *paper was performed at the Jet Propulsion Laboratory,* **California Insti-***tute of Technology, under contract with the National Aeronautics and Space Administration (NASA).*

Appendix

A. Proof of Positive Definiteness of Matrices *A* and C

The positive definiteness of matrices A, C, and $\mathcal{A}'$ **is established** based on the following theorem.

Theorem. If $B\varepsilon\Re^{n\times n}$ **is** positive definite and $X\varepsilon\Re^{n\times k}$ has rank *k,* then $X^tBX\varepsilon\Re^{k\times k}$ is also positive definite.

Proof. See [12, p.140]

The positive definiteness of matrix d, given by (3.14), follows from a repetitive application of the above theorem as follows. The matrix $\mathcal{I}^{-1}\varepsilon\Re^{6N\times 6N}$ is the global matrix of spatial inertia and hence it is positive definite. From its definition it follows that the matrix $\mathcal{P}\varepsilon\Re^{6N\times 6N}$ has full rank of 6N and hence the matrix $\mathcal{P}^t\mathcal{I}^{-1}\varepsilon\Re^{6N\times 6N}$ **is** positive definite. Since the axes of articulation of each joint are orthogonal it then follows that the matrix $\mathcal{W}\varepsilon\Re^{5N\times N}$ has full rank of N which implies that the matrix *A* is positive definite. The positive definiteness of matrix C, given by (3.16), follows from a similar argument.

B. Proof of Positive Definiteness of Matrix *A'*

The positive definiteness of matrix *A'* is established by showing that, for some nonzero vector $X\varepsilon\Re^{5N}$, we have $X^t\mathcal{A}'X > 0$. To this end, from (4.13) we have

$$X^t\mathcal{A}'X = X^t\mathcal{A}X + X^t\mathcal{W}^t\mathcal{I}_b^{-1}\mathcal{W}X$$

Since the matrix d is positive definite we then have $X^t\mathcal{A}X > 0$. Now consider a block representation of vector X as $X = \text{Col}\{X_i\}$, i = N to 1, with $X_i\varepsilon\Re^5$. From the definition of W and $\mathcal{I}_b^{-1}$ **it** then follows that

$$X^t W^t \mathcal{I}_b^{-1} W X = X_1^t W_1^t I_1^{-1} W_1 X_1$$

The positive definiteness of matrix $W_1^t I_1^{-1} W_1$ is *easily* established from the above theorem. It then follows that $X_1^t W_1^t I_1^{-1} W_1 X_1 > 0$ and hence $X^t A' X >$ *0* since it is a sum of two positive terms.

References

1. Y. XU and T. Kanade (Eels.), Space *Robotics: Dynamics and Control,* Kluwer Academic Pub., 1993.

2. A. Fijany and A.K. Bejczy," Techniques for Parallel Computation of Mechanical Manipulator Dynamics. Part II: Forward Dynamics," in *Advances in Control and Dynamic Systems, Vol. 40: Advances* in *Robotic Systems* Dynamics *and Control,* C.T. Leondes (Ed.), pp. 357-410, Academic Press, March 1991.

3. J.Y.S. Luh, M.W. Walker, and R.P.C. Paul,''On-Line Computational Scheme for Mechanical Manipulator," *ASME J.* Dynamic *Syst., Meas., and Control,* Vol. 102, pp. 69-76, June 1980.

4. A.F. Vereshchagin,''Computer Simulation of Dynamics of Complicated Mechanism of Robot-Manipulators," *Engineering Cybernetics, No.(6),* pp. 65-70, 1974.

5. R. Featherstone," The Calculation of Robot Dynamics Using Articulated-Body Inertia," *Int. J. Robotics Res.,* Vol. 2(l), pp. 13-30, 1983.

6. G. Rodriguez and K. Kreutz-Delgado,''Spatial Operator Factorization and Inversion of the Manipulator Mass Matrix," *IEEE Trans. Robotics & Automation,* Vol. 8(l), pp. 65-76, Feb. 1992.

7. A. Fijany, I. Sharf, and G.M.T. D'Eleuterio," Parallel *O(Log N)* Algorithms for Computation of Manipulator Forward Dynamics," *IEEE Trans. Robotics & Automation, Vol. 11(3),* pp. 389-400, June 1995.

8. A. Fijany,"Shur Complement Factorization and Parallel *O(Log N)* Algorithms for Computation of Operational Space Mass Matrix and its Inverse Forward Dynamics," *Proc. IEEE Int. Conf. Robotics & Automation,* pp. 2369-2376, 1994.

9. R. Featherstone, *Personal Communication,* 1996.

10. R. Featherstone and A. Fijany," A Technique for Analysing Constrained Rigid-Body Systems and its Application to the Constraint Force Algorithm.'' *In preparation.*

11. C.S.G. Lee and P.R. Chang," Efficient Parallel Algorithms for Robot Inverse Dynamics Computation," IEEE Trans. Syst., Man, and Cybern., Vol. 16(4), pp. 532-542, July/August 1986.

12. G.H. Golub and C.F. Van Loan, *Matrix Computations,* 2nd Ed., The John Hopkins Univ. Press, 1989.

13. A. Fijany, G. Kwan, and N. Bagherzadeh,"A Fast Algorithm for Parallel Computation of Multibody Dynamics on MIMD Parallel Architectures," Proc. *Computing in Aerospace 9 Conf.* pp. 1021-1032, San Diego, CA, Oct. 1993.

Nomenclature

N	Number of Degrees-Of-Freedom (DOF) of the system
$p_{i,j}$	Position vector from O_j to O_i, $p_{i+1,i} = p_i$
m,	Maas of link i
J_i	**Second moment of mass of link i about its center of mass**
hi, k_i	**First and Second Moment of mass of link i about point O_i**
$Q_i, \dot{Q}_i, \ddot{Q}_i$	**Position, velocity, and acceleration of joint i**
Γ_i	**Applied (control) force on joint i**
$\omega_i, \dot{\omega}_i \epsilon \Re^3$	**Angular velocity and acceleration of link i**
$v_i, \dot{v}_i \epsilon \Re^3$	**Linear velocity and acceleration** of **link i, point O_i**
$f_i, n_i \epsilon \Re^3$	**Force and moment of interaction between link i-1 and link i**
Hi	**Spatial axis (map matrix) of joint i, $H_i \epsilon \Re^{6 \times k}$ for a joint with k DOFS**
$I_{i,j} \epsilon \Re^{6 \times 6}$	**Spatial Inertia of body i about point** Oj, $I_{i,i} = I_i$ $I_i = \begin{bmatrix} k_i & \tilde{h}_i \\ \tilde{h}_i^t & m_i U \end{bmatrix}$ **(t denotes transpose)**
$V_i = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} \epsilon \Re^6$	**Spatial velocity of link i, point O_i**
$\dot{V}_i = \begin{bmatrix} \dot{\omega}_i \\ \dot{v}_i \end{bmatrix} \epsilon \Re^6$	**Spatial acceleration of link i, point O_i**

$F, = \begin{bmatrix} n_i \\ f_i \end{bmatrix} \epsilon \Re^6$	**Spatial force of interaction between link i-1 and link i**
$F_{N+1} \epsilon \Re^6$	**External spatial force acting on the End-Effecter (EE)**
$V_{N+1}, \dot{V}_{N+1} \epsilon \Re^6$	**EE Spatial velocity and acceleration, point O_{N+1}**

Global Quantities, i = N to 1

$\mathcal{M} \epsilon \Re^{N \times N}$	**Symmetric Positive Definite (SPD) mass matrix**
$\mathcal{J} \epsilon \Re^{6 \times N}$	**Jacobian Matrix**
$\mathcal{H} = \text{Diag}\{H_i\}$	**Global matrix of spatial axes, $\mathcal{H} \epsilon \Re^{6N \times N}$ for a system with 1 DOF joints.**
$\mathcal{I} = \text{Diag}\{I_i\} \epsilon \Re^{6N \times 6N}$	**Global matrix of spatial inertia**
$Q = \text{Col}\{Q_i\} \epsilon \Re^N$	**Global Vector of joint positions**
$Q = \text{Col}\{\dot{Q}_i\} \epsilon \Re^N$	**Global vector** of **joint velocities**
$Q = \text{Col}\{\ddot{Q}_i\} \epsilon \Re^N$	Global vector of joint accelerations
$\Gamma = \text{Col}\{\Gamma_i\} \epsilon \Re^N$	Global vector of applied joint forces
$\mathcal{V} = \text{Col}\{V_i\} \epsilon \Re^{6N}$	**Global vector of spatial velocities**
$\dot{\mathcal{V}} = \text{Col}\{\dot{V}_i\} \epsilon \Re^{6N}$	**Global vector of spatial accelerations**
$\mathcal{F} = \text{Col}\{F_i\} \epsilon \Re^{6N}$	**Global vector of spatial interaction forces**