SYMBOLIC FACTORIZATION OF INERTIA MATRIX FOR SPACE ROBOT SIMULATION

Subir Kumar Saha
R&D Center, Toshiba Corporation
4-1 Ukishima-cho, Kawasaki-ku, Kawasaki 210, Japan
E-mail: saha@mel.uki.rdc.toshiba.co.jp

Abstract: In simulation of a space robot, factorization of its inertia matrix is necessary to solve for the joint accelerations. Usually, this factorization is done numerically. For that, computational complexity is cubic in the degrees of freedom (DOF) of the system at hand. In this paper, a symbolic factorization of the inertia matrix is presented. This allows to calculate the joint accelerations with the complexity that is linear in the DOF. The approach is originally introduced for the simulation of fixed-base industrial manipulators. Here, the method is extended to serial type space robots.

Keywords: Space robot, Reverse Gaussian elimination, Symbolic factorization, Simulation.

1 Introduction

The inertia matrix of a space robot results from its dynamic equations of motion. In order to simulate the robot's motion, the dynamic equations are solved for the joint accelerations while the external forces and torques are given. The step is same as to solve a set of linear algebraic equations. Since the inertia matrix is symmetric positive definite (SPD), the solution is obtained by, first, factoring the inertia matrix using the Cholesky decomposition and, then, calculating the joint accelerations with forward and backward substitutions[1]. Note here that instead of the Cholesky decomposition Gaussian elimination (GE) can also be used, which has the same complexity for a SPD matrix.

The conventional way to perform simulation is to follow the above steps *numerically*, i.e., the elements of the inertia matrix are evaluated as *numbers* before the rules of the GE are applied followed by the substitutions. Now, a space robot consisting of an n degrees of freedom (DOF) mounted on a free-base, e.g., a spacecraft, as shown in Fig. 1, the total DOF of the system is $(n+6)$. Therefore, either Cholesky decomposition or Gaussian elimination requires computations of order $(n+6)^3/6$. As a result, complexity of the resulting simulation algorithm is in the order of $(n+6)^3$, i.e., $\mathcal{O}[(n+6)^3]$. For a fixed-base manipulator, however, the complexity is $\mathcal{O}(n^3)$, as in 2), because the base is fixed and the DOF is n.

In 3), an approach, called the *articulated-body inertia* (ABI), was presented for the simulation of fixed-base robotic systems. This calculates the joint accelerations of an n-DOF fixed-base robots recursively with $\mathcal{O}(n)$ computations. Later, in 4) and the subsequent publications, many recursive $\mathcal{O}(n)$ simulation algorithms for different robotic systems were presented without any exact complexity count. The approach is based on the Kalman filtering and smoothing techniques, arising in the state estimation theory.

Comparison of conventional and recursive approaches shows that the ABI based approach[3], in contrast to the scheme in 2), leads to more efficient simulation algorithm when DOF$\geq$12. Thus, for space robots having a 6-DOF manipulator mounted on the spacecraft that also has 6-DOF, it is natural that the recursive scheme[3] is efficient for the 12-DOF space robot. This is reflected in the results of 5), where a recursive simulation algorithm for the space robot is presented that is better than the scheme of 2).

In this paper, an approach based on the Gaussian elimination (GE) is proposed to develop a recursive $\mathcal{O}(n)$ simulation algorithm. Thus, the method is different from previously reported $\mathcal{O}(n)$ algorithms[3),4)]. Here, instead of numerical calculations, symbolic operations are performed for the GE. That is, contrary to find the ele-

PR0001-3/95/0000-1137 $4.00

ments of the inertia matrix as numbers, they are obtained as expressions. Then, the rules of the GE are applied to those symbolic elements. Consequently, a unique characteristic is noted among the elements of the resulting matrices.

The elements resulting from the symbolic GE show recursive relations. The observation is not obvious if the GE is done numerically, as in 2). As a result, the joint accelerations, as required in simulation, can be calculated recursively that has the computational complexity of $\mathcal{O}(n)$. This approach is originally proposed in 6) for the fixed-base robots whose computational complexity matches to that of 3), i.e., the scheme is efficient if DOF$\geq$12. Thus, it can be predicted that the proposed approach, compared to the one in 2), will lead to an efficient simulation algorithm for the space robots under study. Moreover, the symbolic GE provides many physical interpretations[6], as in 3) and 4). The present approach is, however, simple to understand.

The paper is organized as follows: First, Section 2 derives a suitable representation of the inertia matrix for symbolic GE. In Section 3, the symbolic factorization is performed, and a simulation algorithm is proposed in Section 4. Finally, conclusions are given in Section 5.

2 Inertia Matrix

As shown in Fig. 1, a space robot under study is composed of $n+1$ rigid bodies coupled by n joints. For simplicity, all joints are assumed to be revolute. However, any other type can be assumed. Now, the rigid bodies of the robot are indicated in Fig. 1 by $\#0, \cdots, \#n$, whereas the revolute joints are located at $O_1, \cdots, O_n$. Note the unconventional numbering of the rigid bodies and the joints. This is to compare the derivations for the space robot under study to those for the fixed-base robot, as done in 6).

Next, associated to a rigid body, say, $\#i$ of Fig. 1, two terms, the 6-dimensional *twist* vector, $\mathbf{t}_i$, and the 6×6 *extended mass* matrix, $\mathbf{M}_i$, with respect to the mass center of the ith body, C_i, are defined as follows:

$$\mathbf{t}_i \equiv \begin{bmatrix} \boldsymbol{\omega}_i \\ \dot{\mathbf{c}}_i \end{bmatrix} \quad \text{and} \quad \mathbf{M}_i \equiv \begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i\mathbf{1} \end{bmatrix} \qquad (1)$$

where

$\boldsymbol{\omega}_i$ and $\dot{\mathbf{c}}_i$: the 3-dimensional vectors of angular velocity and the velocity of C_i, respectively;

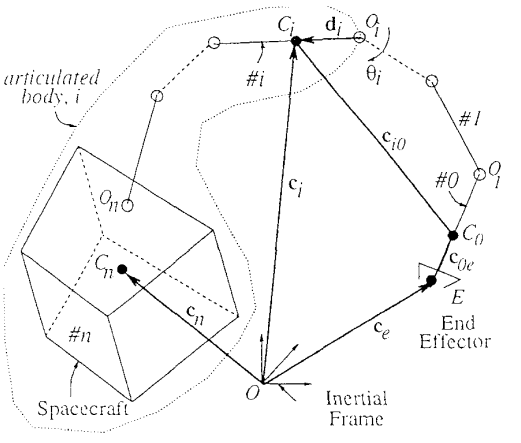

Fig. 1 A schematic diagram of a space robot.

$\mathbf{I}_i$ and m_i: the 3×3 inertia tensor about C_i and the mass of the ith body, respectively;

$\mathbf{1}$ and $\mathbf{O}$: the 3×3 identity and zero matrices, respectively.

Similar to the definitions of eq.(1), the $6(n+1)$-dimensional vector, $\mathbf{t}$, and the $6(n+1)\times 6(n+1)$ matrix, $\mathbf{M}$, for the whole system are given as

$$\mathbf{t} \equiv [\mathbf{t}_0^T, \cdots, \mathbf{t}_n^T]^T, \ \mathbf{M} \equiv \mathrm{diag}(\mathbf{M}_0, \cdots, \mathbf{M}_n) \quad (2)$$

Since the space robot at hand has $n+6$ degrees of freedom (DOF), its $(n+6)$ independent scalar dynamic equations are written from the Newton-Euler equations of the uncoupled rigid bodies and the *natural orthogonal complement* (NOC) matrix[7] as

$$\mathbf{I}\dot{\upsilon} = \boldsymbol{\phi}, \quad \text{where} \quad \mathbf{I} \equiv \mathbf{T}^T\mathbf{M}\mathbf{T} \qquad (3)$$

in which

$\mathbf{I}$: the $(n+6)\times(n+6)$ inertia matrix.

υ and $\boldsymbol{\phi}$: the $(n+6)$-dimensional vectors of independent generalized speeds and forces, respectively. The generalized forces are due to the external moments and forces, and those arising from the gravity, and centrifugal and coriolis accelerations. Vectors υ and $\boldsymbol{\phi}$ are defined as

$$\upsilon \equiv [\mathbf{t}_e^T, \ \dot{\boldsymbol{\theta}}^T]^T \text{ and } \boldsymbol{\phi} \equiv [\mathbf{w}_e^T, \ \boldsymbol{\tau}^T]^T \quad (4)$$

The vectors, $\boldsymbol{\theta}$ and $\boldsymbol{\tau}$, are the n-dimensional vectors of joint angles and torques, respectively, namely,

$$\boldsymbol{\theta} \equiv [\theta_1, \ \cdots, \ \theta_n]^T, \ \boldsymbol{\tau} \equiv [\tau_1, \ \cdots, \ \tau_n]^T$$

On the contrary, $\mathbf{t}_e \equiv [\boldsymbol{\omega}_e^T,\ \dot{\mathbf{c}}_e^T]^T$ is the twist of the end-effector, E of Fig. 1—$\boldsymbol{\omega}_e$ and $\dot{\mathbf{c}}_e$ are the angular velocity of the end-effector, E, and its velocity, respectively—, and $\mathbf{w}_e$ is the 6-dimensional vector consisting of moments about E and the forces acting at E, respectively.

$\mathbf{T}$: the $6(n+1)\times(n+6)$ NOC matrix of the system at hand that relates $\boldsymbol{\upsilon}$ with $\mathbf{t}$[6),7)], i.e,

$$\mathbf{t} = \mathbf{T}\boldsymbol{\upsilon}$$

Finally, the $(n+6)\times(n+6)$ inertia matrix, I of eq.(3), is derived using the steps reported in 6) as

$$\mathbf{I} \equiv \begin{bmatrix} \mathbf{I}_e & \mathbf{I}_{em} \\ \mathbf{I}_{em}^T & \mathbf{I}_m \end{bmatrix} \tag{5}$$

where

$$\mathbf{I}_e \equiv \mathbf{B}_{0e}^T \tilde{\mathbf{M}}_0 \mathbf{B}_{0e} \tag{6}$$

and the 6-dimensional ith row vector of the $n\times 6$ matrix, $\mathbf{I}_{em}^T$, denoted by $\mathbf{i}_i^T$, and the scalar elements of the $n\times n$ symmetric matrix, $\mathbf{I}_m$, i_{ij}, are expressed as

$$\mathbf{i}_i^T \equiv \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{B}_{ie} \quad \text{and} \quad i_{ij} \equiv \mathbf{p}_i^T \tilde{\mathbf{M}}_i \mathbf{B}_{ij} \mathbf{p}_j \tag{7}$$

for $i = 1,\cdots,n$; $j = 1,\cdots,i$. Note that the term, i_{ij}, represents the diagonal and the below-diagonal elements of $\mathbf{I}_m$ that is associated to the n-DOF manipulator[6)]. Moreover, in eqs.(6) and (7), the 6×6 matrix, $\tilde{\mathbf{M}}_i$, for $i = 0,\cdots,n$, is given as

$$\tilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \mathbf{B}_{i+1,i}^T \tilde{\mathbf{M}}_{i+1} \mathbf{B}_{i+1,i} \tag{8}$$

Matrix $\tilde{\mathbf{M}}_i$ is the extended mass of the *composite body*, i, with respect to the mass center of the ith body, C_i. A composite body i is defined as the rigidly connected bodies, $\#i$, $\cdots$, $\#n$. Referring to Fig. 1, if the joints of the system inside the dotted line are locked then the system becomes the composite body, i. Furthermore, $\mathbf{B}_{ij}$ and $\mathbf{p}_i$ are the 6×6 matrix and the 6-dimensional vector, respectively. They are associated to the velocity relation between two successive rigid bodies, namely, i and j, which is written[6)] as

$$\mathbf{t}_i = \mathbf{B}_{ij}\mathbf{t}_j + \mathbf{p}_i\dot{\theta}_i \tag{9}$$

where $\mathbf{B}_{ij}$ and $\mathbf{p}_i$ are as follows:

$$\mathbf{B}_{ij} \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{C}_{ij} & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{p}_i \equiv \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_i \times \mathbf{d}_i \end{bmatrix} \tag{10}$$

$\mathbf{C}_{ij}$ being the 3×3 cross-product tensor associated to the vector, $\mathbf{c}_{ij}$, as shown in Fig. 1 for $j = 0$. The tensor, $\mathbf{C}_{ij}$, is defined such that

$$\mathbf{C}_{ij}\mathbf{x} \equiv \mathbf{c}_{ij} \times \mathbf{x}$$

for any arbitrary 3-dimensional vector $\mathbf{x}$. The term, $\mathbf{B}_{ie}$ of eq.(6), is, however, associated to the vector, $\mathbf{c}_{ie}$, as shown in Fig. 1 for $i = 0$. Moreover, in eq.(10), $\mathbf{e}_i$ is the 3-dimensional unit vector parallel to the axis of the ith revolute joint and the vector, $\mathbf{d}_i$, is shown in Fig. 1.

3 Symbolic Factorization

The proposed symbolic factorization is based on the Gaussian elimination (GE)[1)] of the inertia matrix, I of eq.(5). The elimination is, however, done here in the *reverse* order. Thus the name, reverse GE (RGE) is used. It helps to systematically interpret the associated matrices[6)].

In the RGE, the elementary upper triangular matrices (EUTM)[6)] are premultiplied to I, eq.(5), to reduce it to a lower triangular matrix. Note here that due to the difficulties in writing each element of $\mathbf{I}_e$ and $\mathbf{I}_{em}^T$ symbolically, as obvious from eqs.(6) and (7), respectively, the symbolic RGE of I cannot be completed. So, it is done for the last n rows of I, eq.(5). This reduces $\mathbf{I}_m$ to a lower triangular matrix, $\hat{\mathbf{L}}_m$, and I to a lower *block* triangular matrix, say, L, that is obtained as

$$\mathbf{EI} = \mathbf{L} \quad \text{where} \quad \mathbf{E} \equiv \mathbf{E}_1,\ \cdots,\ \mathbf{E}_n \tag{11}$$

In eq.(11), the $(n+6)\times(n+6)$ EUTM, $\mathbf{E}_k$, for $k = 1,\cdots,n$, is shown in Appendix A.1, whereas the $(n+6)\times(n+6)$ matrix, L, is represented as

$$\mathbf{L} \equiv \begin{bmatrix} \hat{\mathbf{I}}_e & \mathbf{O} \\ \hat{\mathbf{I}}_{em}^T & \hat{\mathbf{L}}_m \end{bmatrix} \tag{12}$$

where $\mathbf{O}$ is the $6\times n$ matrix of zeros, and $\hat{\mathbf{I}}_e$, $\hat{\mathbf{I}}_{em}^T$, and $\hat{\mathbf{L}}_m$ are obtained from eqs.(28)-(30), as derived in Appendix A.2.

Note in eq.(12) that the $n\times n$ matrix, $\hat{\mathbf{L}}_m$, is a lower triangular matrix associated to the n-DOF manipulator, as derived in 6), whose scalar elements can be obtained symbolically. On the contrary, the elements of $\hat{\mathbf{I}}_e$ and $\hat{\mathbf{I}}_{em}^T$ cannot be suitably written in symbolic forms.

From eq.(11), $\mathbf{I} \equiv \mathbf{E}^{-1}\mathbf{L}$, where $\mathbf{E}^{-1}$ is the $(n+6)\times(n+6)$ upper traingular matrix that is

denoted by $\mathbf{U}$ and given as

$$\mathbf{U} \equiv \mathbf{E}^{-1} \equiv \begin{bmatrix} 1 & \boldsymbol{\alpha}_{e1} & \boldsymbol{\alpha}_{e2} & \cdots & \boldsymbol{\alpha}_{en} \\ & 1 & \alpha_{11} & \cdots & \alpha_{1n} \\ & & 1 & \cdots & \alpha_{2n} \\ & & & \ddots & \vdots \\ & 0's & & & 1 \end{bmatrix} \quad (13)$$

where "$0's$" denote zeros, and the 6-dimensional vectors, $\boldsymbol{\alpha}_{ej}$, and the scalars, α_{ij}, for $i = 1, \cdots, j$; $j = 1, \cdots, n$, are obtained in eqs.(23)–(25) of Appendix A.1.

Now, for the SPD matrix, $\mathbf{I}$ of eq.(5), it can be shown[1] that

$$\mathbf{I} \equiv \mathbf{U}\mathbf{D}\mathbf{U}^T \quad (14)$$

where $\mathbf{D}\mathbf{U}^T = \mathbf{L}$, $\mathbf{L}$ being shown in eq.(12). The $(n+6) \times (n+6)$ matrix, $\mathbf{D}$, is as follows:

$$\mathbf{D} \equiv \begin{bmatrix} \hat{\mathbf{I}}_e^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0}^T & \frac{1}{\hat{m}_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \mathbf{0}^T & 0 & \cdots & \frac{1}{\hat{m}_n} \end{bmatrix} \quad (15)$$

where $\mathbf{0}$ is the 6-dimensional zero vector.

4 Simulation Algorithm

For simulation, the right hand side of eq.(3), vector $\boldsymbol{\phi}$, is known. Then, the vector of accelerations, $\dot{\boldsymbol{\upsilon}}$, is solved from the following three sets of equations:

$$\mathbf{U}\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}, \quad \mathbf{D}\tilde{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}}, \quad \text{and} \quad \mathbf{U}^T\dot{\boldsymbol{\upsilon}} = \tilde{\boldsymbol{\phi}} \quad (16)$$

where eq.(14) is substituted into eq.(3). Using eq.(16), vector $\dot{\boldsymbol{\upsilon}}$ is obtained recursively as

1. *Solution for* $\hat{\boldsymbol{\phi}}$: Let, $\hat{\boldsymbol{\phi}} \equiv [\hat{\mathbf{w}}_e^T, \hat{\boldsymbol{\tau}}^T]^T$. Then, the the 6-dimensional vector, $\hat{\mathbf{w}}_e$, and the n-dimensional vector, $\hat{\boldsymbol{\tau}}$, are obtained from $\mathbf{U}\hat{\boldsymbol{\phi}} = \boldsymbol{\phi}$ in two different steps, namely,

   (M) Find $\hat{\boldsymbol{\tau}}$: For $i = n-1, \cdots, 1$

   $$\hat{\tau}_i = \tau_i - \mathbf{p}_i^T \tilde{\boldsymbol{\eta}}_{i,i+1} \quad (17)$$

   where $\hat{\tau}_n \equiv \tau_n$, and the 6-dimensional vector, $\tilde{\boldsymbol{\eta}}_{i,i+1}$, is given as

   $$\tilde{\boldsymbol{\eta}}_{i,i+1} \equiv \mathbf{B}_{i+1,i}^T \tilde{\boldsymbol{\eta}}_{i+1} \quad (18)$$

   In eq.(18), $\tilde{\boldsymbol{\eta}}_{i+1} \equiv \hat{\boldsymbol{\psi}}_{i+1}\hat{\tau}_{i+1} + \tilde{\boldsymbol{\eta}}_{i+1,i+2}$, and $\tilde{\boldsymbol{\eta}}_{n,n+1} = \mathbf{0}$.

   (E) Find $\hat{\mathbf{w}}_e$: It is obtained as

   $$\hat{\mathbf{w}}_e = \mathbf{w}_e - \mathbf{B}_{1e}^T(\hat{\boldsymbol{\psi}}_1\hat{\tau}_1 + \tilde{\boldsymbol{\eta}}_{12})$$

   where $\tilde{\boldsymbol{\eta}}_{12}$ is obtained from eq.(18).

2. *Solution for* $\tilde{\boldsymbol{\phi}}$: Here $\tilde{\boldsymbol{\phi}} \equiv [\tilde{\mathbf{w}}_e^T, \tilde{\boldsymbol{\tau}}^T]^T$. Vectors $\tilde{\mathbf{w}}_e$ and $\tilde{\boldsymbol{\tau}}$ are calculated from $\mathbf{D}\tilde{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}}$, where $\tilde{\mathbf{w}}_e$ needs to be evaluated first. Thus,

   (E) Find $\tilde{\mathbf{w}}_e$: The soluion is given as

   $$\tilde{\mathbf{w}}_e = \hat{\mathbf{I}}_e^{-1}\hat{\mathbf{w}}_e \quad (19)$$

   Equation (19) is solved numerically using the GE, and the backward and forward substitutions[1] because the elements of matrix $\hat{\mathbf{I}}_e$, eq.(6), cannot be represented symbolically. For eq.(19), the numerical algorithm has the complexity of $\mathcal{O}(6^3)$, which is a constant value.

   (M) Find $\tilde{\tau}$: For $i = 1, \cdots, n$

   $$\tilde{\tau}_i = \hat{\tau}_i / \hat{m}_i$$

3. *Solution for* $\dot{\boldsymbol{\upsilon}}$: Since $\dot{\boldsymbol{\upsilon}} \equiv [\dot{\mathbf{t}}_e^T, \ddot{\boldsymbol{\theta}}^T]^T$, $\dot{\mathbf{t}}_e$ and $\ddot{\boldsymbol{\theta}}$ are calculated as

   (E) Find $\dot{\mathbf{t}}_e$: Due to the structure of $\mathbf{U}$, as in eq.(13), vector $\dot{\mathbf{t}}_e$ is nothing but $\tilde{\mathbf{w}}_e$, i.e.,

   $$\dot{\mathbf{t}}_e \equiv \tilde{\mathbf{w}}_e$$

   where $\tilde{\mathbf{w}}_e$ is calculated in eq.(19).

   (M) Find $\ddot{\boldsymbol{\theta}}$: For $i = 1, \cdots, n$

   $$\ddot{\theta}_i = \tilde{\tau}_i - \hat{\boldsymbol{\psi}}_i^T \tilde{\boldsymbol{\mu}}_{i,i-1} \quad (20)$$

   where the 6-dimensional vector, $\tilde{\boldsymbol{\mu}}_{i,i-1}$, is obtained from

   $$\tilde{\boldsymbol{\mu}}_{i,i-1} \equiv \mathbf{B}_{i,i-1}\tilde{\boldsymbol{\mu}}_{i-1} \quad (21)$$

   in which, $\tilde{\boldsymbol{\mu}}_0 \equiv \mathbf{B}_{0e}\dot{\mathbf{t}}_e$, and, for $i > 1$, $\tilde{\boldsymbol{\mu}}_{i-1} \equiv \mathbf{p}_{i-1}\ddot{\theta}_{i-1} + \tilde{\boldsymbol{\mu}}_{i-1,i-2}$.

The above algorithm has two distinct parts: one marked with (M) and another with (E). The parts with (M) are associated to the manipulator that are same as for the fixed-base manipulators[6], except in the expression for $\ddot{\theta}_1$. The algorithm of 6) for fixed-based manipulators requires $(201n - 335)$ multiplications or divisions and $(193n - 361)$ additions or subtractions, i.e., the order of complexity is n, $\mathcal{O}(n)$.

Now, for the space robot, if the steps marked with (E) and the additional computations to find $\ddot{\theta}_1$, as in (M) of *Solution for* $\dot{\boldsymbol{\upsilon}}$ above, are added the overall complexity of the proposed algorithm also becomes of $\mathcal{O}(n)$.

5 Conclusions

Based on the symbolic factorization of the inertia matrix of the space robot, as shown in Fig 1, a recursive $\mathcal{O}(n)$ algorithm is presented in Section 4. The exact complexity count for the space robot and the comparison with other recursive algorithms, e.g., 5), are, however, not reported here. This will be done next.

Note that the proposed factorization based on the Gaussian elimination (GE) has the following characteristics: (i) The derivations are easy to understand because they are based on the simple rules of the GE. (ii) The methodology is very good for analytical investigations since $\tilde{\mathbf{M}}_i$, $\hat{m}_i$, and $\hat{\mathbf{M}}_{i,i+1}$, as appear in eqs.(7), (24), and (26), respectively, have physical interpretations. They are: (a) $\tilde{\mathbf{M}}_i$ is the extended mass of the composite body i with respect to the mass center of the ith link, C_i; (b) $\hat{\mathbf{M}}_{i,i+1}$ is the extended mass of *articulated body* i—a system comprising of links $\#i, \cdots, \#n$ that are coupled by joints $i+1, \cdots, n$, as shown in Fig. 1—with respect to C_i. The matrix is same as the *articulated-body inertia*[3], and the *state estimation error covariance*[4]; and (c) $\hat{m}_i$ is the moment of inertia of *articulated body* i about the axis of rotation of the ith revolute joint.

Finally, since no classification is made about the type of space robot, for example, free-flying, the proposed factorization is applicable to any type of space robots.

6 References

1) G.E. Stewart, *Introduction to Matrix Computations*, Academic Press, Inc., NY (1973).

2) M.W. Walker, and D.E. Orin, "Efficient dynamic computer simulation of robotic mechanisms," *ASME J. of Dyn. Sys., Meas., and Cont.*, V. 104, Sept., pp. 205–211 (1982).

3) R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. of Rob. Res.*, V. 2, N. 1, pp. 13–30 (1983).

4) G. Rodriguez, "Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics," *IEEE Trans. on R&A*, Vol. RA-3, N. 6, pp. 624–639 (1987).

5) K. Yamada, and K. Tsuchiya, "Efficient simulation algorithm for a space manipulator," *Proc. of the 7th Int. Symp. on Space Tech. and Sc.*, Tokyo, pp. 1197–1204 (1990).

6) S.K. Saha, "The $\mathbf{UDU}^T$ decomposition of manipulator inertia matrix," accepted for presentation and publication in the *Proc. of the IEEE Conf. on R&A*, Nagoya, Japan, May 21–27 (1995).

7) S.K. Saha, and J. Angeles, "Dynamics of nonholonomic mechanical systems using a natural orthogonal complement," *ASME J. of Appl. Mech.*, V. 58, March, pp. 238–243 (1991).

A. Appendix

A.1 Matrix $\mathbf{E}_k$

The $(n+6)\times(n+6)$ elementary upper triangular matrix (EUTM) of order $n+6$ and index k is defined as

$$\mathbf{E}_k \equiv \begin{bmatrix} 1 & 0 & \cdots & -\boldsymbol{\alpha}_{ek} & \cdots & 0 \\ & 1 & \cdots & -\alpha_{1k} & \cdots & 0 \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & -\alpha_{k-1,k} & \cdots & 0 \\ & & & 1 & \cdots & 0 \\ & & 0's & & \ddots & \vdots \\ & & & & & 1 \end{bmatrix} \qquad (22)$$

where "$0's$" imply zeros, and **1** and **0** are the 6×6 identity matrix and the 6-dimensional vector of zeros, respectively. Moreover, the 6-dimensional vector, $\boldsymbol{\alpha}_{ek}$, and the scalar, α_{ik}, are calculated from the following scheme:

- For $k = n, \cdots, 1$; Do $i = k-1, \cdots, 1$;

$$\boldsymbol{\alpha}_{ek} = \mathbf{B}_{ke}^T \hat{\boldsymbol{\psi}}_k, \qquad \alpha_{ik} = \mathbf{p}_i^T \hat{\boldsymbol{\psi}}_{ik} \qquad (23)$$

  end do i; end for k.

where $\mathbf{p}_i$ is defined in eq.(10), and the 6-dimensional vectors, $\hat{\boldsymbol{\psi}}_k$ and $\hat{\boldsymbol{\psi}}_{ik}$, are obtained from the following relations:

$$\boldsymbol{\psi}_k \equiv \hat{\mathbf{M}}_{k,k+1}\mathbf{p}_k, \qquad \hat{m}_k \equiv \mathbf{p}_k^T \boldsymbol{\psi}_k \qquad (24)$$

$$\hat{\boldsymbol{\psi}}_k \equiv \frac{\boldsymbol{\psi}_k}{\hat{m}_k}, \quad \boldsymbol{\psi}_{ik} \equiv \mathbf{B}_{ki}^T \boldsymbol{\psi}_k, \quad \hat{\boldsymbol{\psi}}_{ik} \equiv \frac{\boldsymbol{\psi}_{ik}}{\hat{m}_k} \qquad (25)$$

in which the 6×6 matrix, $\hat{\mathrm{M}}_{k,k+1}$, can be evaluated recursively[6] as

$$\hat{\mathrm{M}}_{ik} = \mathrm{M}_i + \mathrm{B}^T_{i+1,i} \hat{\mathrm{M}}_{i+1,k} \mathrm{B}_{i+1,i} \qquad (26)$$

where, if $i = k-1$, $\hat{\mathrm{M}}_{i+1,k} \equiv \hat{\mathrm{M}}_{kk} = \hat{\mathrm{M}}_{k,k+1} - \hat{\boldsymbol{\psi}}_k \boldsymbol{\psi}^T_k$, and $\hat{\mathrm{M}}_{n,n+1} \equiv \mathrm{M}_n$.

A.2 Matrix L_k

In order to obtain the $(n+6) \times (n+6)$ matrix, L of eq.(12), from the proposed symbolic RGE, the following scheme is used.

$$\mathrm{L}_k = \mathrm{E}_k \mathrm{L}_{k+1} \qquad \text{for } k = n, \cdots, 1$$

where $\mathrm{L}_{n+1} \equiv \mathrm{I}$ and L_1 is the desired L, i.e. $\mathrm{L} \equiv \mathrm{L}_1$. As an example, the elements of the $(n+6) \times (n+6)$ matrix, L_k, are shown below for $k = n$,

$$\mathrm{L}_n \equiv \begin{bmatrix} \mathrm{I}_e^{(n)} & \mathrm{i}_1^{(n)} & \cdots & \mathrm{i}_{n-1}^{(n)} & \mathbf{0} \\ \mathrm{i}_1^{T(n)} & i_{11}^{(n)} & \cdots & i_{1,n-1}^{(n)} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ \mathrm{i}_{n-1}^{T\ (n)} & i_{n-1,1}^{(n)} & \cdots & i_{n-1,n-1}^{(n)} & 0 \\ \mathrm{i}_n^T & i_{n1} & \cdots & i_{n,n-1} & i_{nn} \end{bmatrix} \qquad (27)$$

where i_{nn} is the *pivot*[1], and $\mathrm{I}_e^{(n)}$, $\mathrm{i}_i^{T(n)}$, and $i_{ij}^{(n)} = i_{ji}^{(n)}$, denote the modified matrix from I_e, vector from i_i^T of I_{em}^T, and the scalar from i_{ij} of I_m, respectively. They are obtained from the following equations:

- For $k = n, \cdots, 1$; Do $i = k-1, \cdots, 1$; Do $j = i, \cdots, 1$

$$\mathrm{I}_e^{(k)} = \mathrm{B}^T_{0e} \hat{\mathrm{M}}_{ik} \mathrm{B}_{0e} \qquad (28)$$

$$\mathrm{i}_i^{T(k)} = \mathrm{p}_i^T \hat{\mathrm{M}}_{ik} \mathrm{B}_{ie} \qquad (29)$$

$$i_{ij}^{(k)} = \mathrm{p}_i^T \hat{\mathrm{M}}_{ik} \mathrm{B}_{ij} \mathrm{p}_j \qquad (30)$$

  end do j; end do i; end for k.