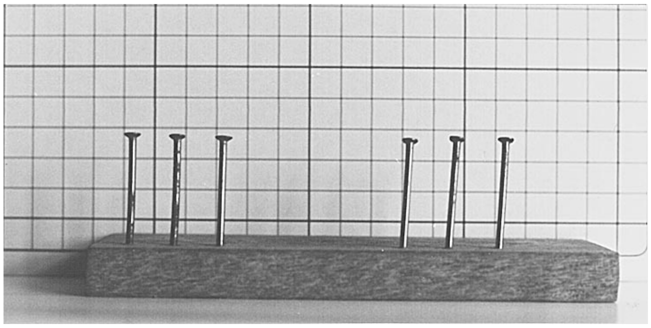

(a) (b)

Fig. 8. Experimental results of driving nails. (a) The proposed hitting using $t = t_{1d}$ and $\gamma = \tan^{-1} k$. (b) A conventional hitting with $t = t_1$ and $\gamma = 0$.

be driven slantwise by the friction force on the face of the nail head. First, the normal of the board is directed with $\boldsymbol{\Phi}_X = 90^\circ$ and $\boldsymbol{\Phi}_Z = 30^\circ$, and a nail is tapped perpendicularly on the board. The end-effector of the manipulator is positioned as described in Section IV, and the hammer is swung by θ_{s1d} of (33) so that the hitting time can be t_{1d}. When the hammer hits the head of a nail, the strain signal $v_s(t)$ was similar to the waveform in Fig. 5. The robot could drive three nails into the board perpendicularly as shown in Fig. 8(a). The perpendicularity of the three nails shows that the tangential velocity is negligible. When the hammer hits a nail at a conventional hitting time t_1, the three nails were driven into the board slantwise not perpendicularly, as shown in Fig. 8(b). The slant is caused by the tangential velocity at the hitting. Next, the board is rotated at $\boldsymbol{\Phi}_X = 60^\circ$ while keeping $\boldsymbol{\Phi}_Z = 30^\circ$ to verify the nailing in a nonvertical plane. The obtained strain signal $v_s(t)$ was similar to Fig. 7 and a nail could also be driven into the board perpendicularly.

The experiments prove that the flexible link hammer can hit an object flatwise without a tangential velocity at the proposed hitting time t_{1d}.

VI. Conclusion

Hitting by a flexible link hammer under gravity has been taken into consideration. The motion of the hammer subjected to gravity has been analyzed in order that the robot can hit an object from any direction in a 3-D space. This paper has derived the hitting conditions that the hammer can strike an object flatwise without a tangential velocity to its face. The conditions are realized by adjusting the swing-up angle of the link and positioning the base of the hammer. It is seen from experiments that the flexible link hammer can drive a nail perpendicularly into a board.

References

[1] W. J. Book, O. Maizza-Neto, and D. E. Whitney, "Feed-back control of two beam, two joint systems with distributed flexibility," *Trans. ASME J. Dyn. Syst. Meas., Contr.*, vol. 97, no. 4, pp. 424–431, 1975.

[2] S. Futami, N. Kyura, and S. Hara, "Vibration absorption control of industrial robots by acceleration feed-back," *IEEE Trans. Indust. Electron.*, vol. 30, pp. 299–305, 1983.

[3] P. J. Nathan and S. N. Singh, "Sliding mode control and elastic mode stabilization of a robotic arm with flexible links," *Trans. ASME, J. Dyn. Syst. Meas., Contr.*, vol. 113, pp. 669–676, Dec., 1991.

[4] Y. Sakawa and Z. H. Lou, "Modeling and control of coupled bending and torsional vibrations of flexible beams," *IEEE Trans. Automat. Contr.*, vol. 34, pp. 970–977, 1989.

[5] T. Fukuda and A. Arakawa, "Control of flexible robotic arms," *Trans. JSME*, vol. 53, no. 488, pp. 954–961, 1987.

[6] W. J. Book and D. Kwon, "Contact control for advanced applications of light weight arms," *J. Intell. Robot. Syst.*, vol. 6, pp. 121–137, 1992.

[7] B. V. Chapnik, G. R. Heppler, and J. D. Aplevich, "Modeling impact on a one-link flexible robotic arm," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 479–488, 1991.

[8] D. M. Rovner and R. H. Cannon, "Experiments toward on-line identification and control of a very flexible one-link manipulator," *Int. J. Robot. Res.*, vol. 6, no. 4, Winter 1987.

[9] D. Wang and M. Vidyasagar, "Modeling a class of multilink manipulators with the last link flexible," *IEEE Trans. Robot. Automat.*, vol. 8, Feb. 1992.

[10] ——, "Control of a class of manipulators with a single flexible link—Part I: Feedback linearization," *Trans. ASME, J. Dynamic Syst., Measure., Contr.*, vol. 113, pp. 655–661, Dec. 1991.

[11] T. Izumi and Y. Hitaka, "Control of impact for a hammering robot using a flexible link," in *Proc. IMACS/SICE Int. Symp. Robot., Mechatron. Manufact. Syst.*, pp. 1347–1352, 1992.

[12] Y. Sakawa and F. Matsuno, "Modeling and control of a flexible manipulator with a parallel drive mechanism," *Int. J. Contr.*, vol. 44, no. 2, pp. 299–313, 1986.

[13] F. Matsuno, S. Fukushima, Y. Ohsawa, M. Kiyohara, and Y. Sakawa, "Feedback control of a flexible manipulator with a parallel drive mechanism," *Int. J. Robot. Res.*, vol. 6, no. 4, pp. 76–84, 1987.

[14] F. Bellezza, L. Lanari, and G. Ulivi, "Exact modeling of the flexible slewing link," *IEEE Int. Conf. Robot. Automat.*, vol. 2, pp. 734–739, 1990.

[15] T. Kida, M. Ikeda, and I. Yamaguchi, "Optimal regulator with low-pass property and its application to LSS control," *Trans. Soc. Instrum. Contr. Eng.*, vol. 25, no. 4, pp. 448–454, 1989.

[16] F. Y. Wang and G. Guan, "Influences of rotatory inertia, shear and loading on vibrations of flexible manipulators," *J. Sound Vibration*, vol. 171, pp. 433–452, 1994.

A Decomposition of the Manipulator Inertia Matrix

Subir Kumar Saha

***Abstract*— A decomposition of the manipulator inertia matrix is essential, for example, in forward dynamics, where the joint accelerations are solved from the dynamical equations of motion. To do this, unlike a numerical algorithm, an analytical approach is suggested in this paper. The approach is based on the symbolic Gaussian elimination of the inertia matrix that reveal recursive relations among the elements of the resulting matrices. As a result, the decomposition can be done with the complexity of order n, $\mathcal{O}(n)$—n being the degrees of freedom of the manipulator—, as opposed to an $\mathcal{O}(n^3)$ scheme, required in the numerical approach. In turn, $\mathcal{O}(n)$ inverse and forward dynamics algorithms can be developed. As an illustration, an $\mathcal{O}(n)$ forward dynamics algorithm is presented.**

***Index Terms*— Articulated-body inertia, Kalman filtering, reverse Gaussian elimination (RGE), serial manipulator, symbolic decomposition.**

I. Introduction

The inertia matrix of a robotic manipulator or the generalized inertia matrix (GIM) arises from the robot's dynamic equations of motion. The decomposition of the GIM is required, for example,

Manuscript received February 21, 1995; revised August 30, 1995. This paper was recommended for publication by Associate Editor V. Kumar and Editor S. E. Salcudean upon evaluation of the reviewers' comments.

The author is with Department of Mechanical Engineering, Indian Institute of Technology, Delhi, Hauz Khas, New Delhi 110 016, India.

Publisher Item Identifier S 1042-296X(97)01052-5.

to perform forward dynamics, which is necessary in simulation. In forward dynamics, the joint accelerations are solved from the dynamic equations of motion. To do this, a numerical algorithm, first, calculates the numerical values of the elements of the GIM. Then, its decomposition is performed, for example, using the Cholesky decomposition [1], before the joint accelerations are calculated by forward and backward substitutions. Hence, for an n degrees of freedom fixed-base serial manipulator, complexity of order n^3, denoted by $\mathcal{O}(n^3)$, is inevitable. Besides, no insight is possible due to pure numerical approach.

Therefore, a different look into the problem was sought, which resulted in a forward dynamics scheme based on the *articulated-body inertia* [2]. This allows one to calculate the joint accelerations with $\mathcal{O}(n)$ computations. The concept bears a resemblance to the approach reported in [3] whose complexity is also of $\mathcal{O}(n)$, but is less efficient than that of [2]. The approach in [2], compared to an efficient $\mathcal{O}(n^3)$ algorithm, e.g., [4], is not suitable for the forward dynamics of manipulators with $n < 12$. This is due to very small coefficient of n^3, namely, $1/6$. On the contrary, the solutions of the joint accelerations using the $\mathcal{O}(n)$ algorithms are smoother [5]. Hence, their numerical integrations, as required in simulation to find out the joint velocities and positions, converge faster. As a result, in simulation, $\mathcal{O}(n)$ algorithms perform better than the $\mathcal{O}(n^3)$ scheme.

Later, in [6] and his other papers, many $\mathcal{O}(n)$ simulation algorithms for different robotic systems are provided without any complexity count. They are based on Kalman filtering and smoothing techniques, arising in the state estimation theory. The approach gives a deeper insight to the manipulator dynamics, which was possible due to the establishment of the equivalency of the *discrete-time state space systems* and the reported *spatially recursive state space model*, in which the *distance* between two successive joints plays the role of *time interval* of the discrete-time models.

In this paper, Gaussian elimination (GE) [1] is performed symbolically to obtain the desired decomposition. That is, instead of evaluating the values of the elements of the generalized inertia matrix (GIM), they are, first, written as *analytical expressions*. This is done by writing the dynamic equations of motion based on the *natural orthogonal complement* (NOC) [7], and its decoupled form [8]. Next, the rules of the GE are applied to obtain the analytical expressions for the elements of the decomposed matrices, which can be evaluated recursively. Note that the detection of the recursive relations in the GE is not so obvious in a numerical approach like [4]. Moreover, using the present approach, $\mathcal{O}(n)$ inverse and forward dynamics algorithms can be developed. In addition, the analytical expressions provide with many physical interpretations. Some of them explain the concepts like the *articulated-body inertia* [2] and the *state estimation error covariance* [6].

This paper is organized as follows: Section II derives the GIM, whereas its decomposition is shown in Section III. As an application of the decomposition, a forward dynamics algorithm is developed in Section IV, whose computational complexity is also reported. Finally, the conclusions are given in Section V.

II. Generalized Inertia Matrix

The $n \times n$ generalized inertia matrix (GIM), $\mathbf{I}$, associated with the dynamic equations of motion of an n-link n-DOF serial manipulator, as shown in Fig. 1, is derived in [7], which is given as

$$\mathbf{I} \equiv \mathbf{T}^T\mathbf{M}\mathbf{T}, \quad \text{where} \quad \mathbf{M} \equiv \mathrm{diag}(\mathbf{M}_1, \cdots, \mathbf{M}_n) \tag{1}$$

and $\mathbf{T}$ is the $6n \times n$ *natural orthogonal complement* (NOC) matrix [7], whereas the 6×6 matrix elements, $\mathbf{M}_i$, for $i = 1, \cdots, n$, of the $6n \times 6n$ generalized mass matrix, $\mathbf{M}$ of (1), are the *extended mass* of the ith link with respect to its mass center C_i, Fig. 2, i.e.,

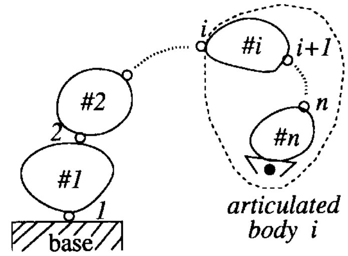

Fig. 1. An n-link n-DOF manipulator.

$$\mathbf{M}_i \equiv \begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i\mathbf{1} \end{bmatrix} \tag{2}$$

in which $\mathbf{I}_i$ and m_i are the 3×3 inertia tensor about C_i and the mass of the ith link, respectively.

Alternatively, the GIM, $\mathbf{I}$, is derived in [8] as

$$\mathbf{I} \equiv \mathbf{T}_d^T\tilde{\mathbf{M}}\mathbf{T}_d, \quad \text{where} \quad \tilde{\mathbf{M}} \equiv \mathbf{T}_l^T\mathbf{M}\mathbf{T}_l \tag{3}$$

and $\mathbf{T}_d \equiv \mathrm{diag}(\mathbf{p}_1, \cdots, \mathbf{p}_n)$. Note from the comparison of (1) and (3), $\mathbf{T} = \mathbf{T}_l\mathbf{T}_d$ [8], which actually allows one to write the analytical expressions for the elements of the GIM, (8). Moreover, the symmetric $6n \times 6n$ matrix, $\tilde{\mathbf{M}}$, is derived as

$$\tilde{\mathbf{M}} \equiv \begin{bmatrix} \tilde{\mathbf{M}}_1 & \mathbf{B}_{21}^T\tilde{\mathbf{M}}_2 & \cdots & \mathbf{B}_{n1}^T\tilde{\mathbf{M}}_n \\ \tilde{\mathbf{M}}_2\mathbf{B}_{21} & \tilde{\mathbf{M}}_2 & \cdots & \mathbf{B}_{n2}^T\tilde{\mathbf{M}}_n \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{M}}_n\mathbf{B}_{n1} & \tilde{\mathbf{M}}_n\mathbf{B}_{n2} & \cdots & \tilde{\mathbf{M}}_n \end{bmatrix} \tag{4}$$

where the 6×6 matrices, $\mathbf{B}_{ij}$, along with the 6-D vectors, $\mathbf{p}_i$, for $i, j = 1, \cdots, n$, are associated with the relation between the *twist* of the ith link, $\mathbf{t}_i \equiv [\boldsymbol{\omega}_i^T, \dot{\mathbf{c}}_i^T]^T$—$\boldsymbol{\omega}_i$ and $\dot{\mathbf{c}}_i$ being the 3-D vectors of angular velocity and linear velocity of the mass center of the ith link, C_i, respectively,—and that of the jth link, $\mathbf{t}_j$, i.e.,

$$\mathbf{t}_i = \mathbf{B}_{ij}\mathbf{t}_j + \mathbf{p}_i\dot{\theta}_i \tag{5}$$

where $\dot{\theta}_i$ is the joint rate of the ith revolute joint, as shown in Fig. 2, and the 6×6 matrix, $\mathbf{B}_{ij}$, and the 6-D vector, $\mathbf{p}_i$, are defined by

$$\mathbf{B}_{ij} \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{C}_{ij} & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{p}_i \equiv \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_i \times \mathbf{d}_i \end{bmatrix} \tag{6}$$

$\mathbf{1}$ and $\mathbf{O}$ being the 3×3 identity and zero matrices, respectively, which, henceforth, should be understood as of being dimensions compatible with the size of the matrix in which they appear. Also, $\mathbf{C}_{ij}$ is the 3×3 cross-product tensor associated with the vector, $\mathbf{c}_{ij} \equiv \mathbf{c}_j - \mathbf{c}_i$ of Fig. 2, which is defined as, $\mathbf{C}_{ij}\mathbf{x} \equiv \mathbf{c}_{ij} \times \mathbf{x}$, for any arbitrary 3-D vector, $\mathbf{x}$, whereas the 3-D vectors, $\mathbf{e}_i$ and $\mathbf{d}_i$, are the unit vector parallel to the axis of the ith revolute joint and a position vector, respectively. For other type of joints, e.g., prismatic, vector $\mathbf{p}_i$ should be derived accordingly. Furtheromre, the 6×6 matrices, $\tilde{\mathbf{M}}_i$, for $i = 1, \cdots, n$, as in (4), are the mass matrix of the ith composite body that consists of rigidly connected links, $\#i, \cdots, \#n$, with respect to C_i. Matrix $\tilde{\mathbf{M}}_i$ can be obtained recursively as

$$\tilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \tilde{\mathbf{M}}_{i,i+1} \quad \text{where} \quad \tilde{\mathbf{M}}_{i,i+1} \equiv \mathbf{B}_{i+1,i}^T\tilde{\mathbf{M}}_{i+1}\mathbf{B}_{i+1,i} \tag{7}$$

and $\tilde{\mathbf{M}}_{n+1} \equiv \mathbf{O}$ since no $(n+1)$st link exists, i.e., $\tilde{\mathbf{M}}_n \equiv \mathbf{M}_n$.

Now, using (6) and the the expression for $\mathbf{T}_d$, as after (3), the elements of the symmetric manipulator inertia matrix or the GIM, $\mathbf{I}, i_{ij}$, are written as

$$i_{ij} \equiv \mathbf{p}_i^T\tilde{\mathbf{M}}_i\mathbf{B}_{ij}\mathbf{p}_j \tag{8}$$

where $i = 1, \cdots, j; j = 1, \cdots, n$. Equation (8) is also derived in [9] that is based on the *Structurally Recursive Method*. It is worthwhile

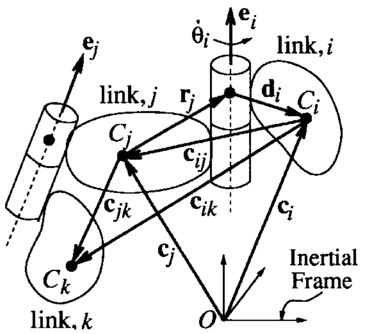

Fig. 2. A system of coupled links.

to mention here that, without the analytical expression for i_{ij}, (8), the symbolic Gaussian elimination, as done in Section III, is not possible.

III. Decomposition of the GIM

In order to find the desired decomposition, Gaussian elimination (GE) [1] of the GIM is performed that never fails, since the matrix, $\mathbf{I}$ of (3), whose elemets are given in (8), is symmetric positive definite (SPD). Note, however, that a modified GE, as described in the Appendix, and defined as the *reverse* Gaussian elimination (RGE), will be used. The reason is to obtain recursive relations that start from the nth link, similar to (4). The steps to decompose the GIM are described below:

1) Based on (23) and (24), the RGE of $\mathbf{I}$ is performed, while $k = n, \cdots, 2$, i.e.,

$$\mathbf{EI} = \mathbf{L}_2 \quad \text{where} \quad \mathbf{E} \equiv \mathbf{E}_2 \cdots \mathbf{E}_n \tag{9}$$

$\mathbf{E}$ and $\mathbf{L}_2$, respectively, being the $n \times n$ upper and lower triangular matrices.

2) An essential property of the elementary upper triangular matrix (EUTM), as defined in (20), is stated here as

$$\mathbf{E}_k^{-1} \equiv \left(\mathbf{1} - \boldsymbol{\alpha}_k \boldsymbol{\lambda}_k^T\right)^{-1} = \mathbf{1} + \boldsymbol{\alpha}_k \boldsymbol{\lambda}_k^T \tag{10}$$

where $\boldsymbol{\alpha}_k$ and $\boldsymbol{\lambda}_k$ are defined in (21) and (22), respectively. Using (10), the GIM, $\mathbf{I}$ of (9), is given as

$$\mathbf{I} = \mathbf{UL}_2, \quad \text{where} \quad \mathbf{U} \equiv \mathbf{E}^{-1}. \tag{11}$$

In (11), $\mathbf{U}$ and $\mathbf{L}_2$ are the $n \times n$ upper and lower triangular matrices, respectively. Moreover, from the inverse of the EUTM, (10), it is clear that the diagonal elements of $\mathbf{U}$ are unity and the above-diagonal elements are the components of the vector, $\boldsymbol{\alpha}_k$, for $k = 2, \cdots, n$, that are evaluated in (24).

3) Since the factorization given by (11) is not unique [1], a unique decomposition is obtained by normalizing the elements of $\mathbf{L}_2$ as

$$\mathbf{L}_2 = \mathbf{DL} \quad \text{where} \quad \mathbf{D} \equiv \text{diag}(\hat{m}_1, \cdots, \hat{m}_n) \tag{12}$$

$\mathbf{D}$ being the $n \times n$ diagonal matrix, whose non zero elements are those of the matrix, $\mathbf{L}_2$, as in (25). Hence, the diagonal elements of matrix $\mathbf{L}$ are unity.

4) Note, for the SPD matrix, $\mathbf{I}, \mathbf{L} \equiv \mathbf{U}^T$ [1]. Thus, the desired decomposition of the manipulator GIM is

$$\mathbf{I} = \mathbf{UDU}^T \tag{13}$$

where the elements of the matrices, $\mathbf{U}$ and $\mathbf{D}$, are evaluated from (24) to (27).

Note here that the matrix, $\hat{\mathbf{M}}_{i,i+1}$, associated with the RGE, as in (26) and others, has the following interpretation: in contrast to the definition of $\tilde{\mathbf{M}}_i$ of *composite body*, (7), matrix $\hat{\mathbf{M}}_{i,i+1}$ is the extended mass of *articulated body* i, which is defined as the system consisting of links $\#i, \cdots, \#n$ that are coupled by joints $i+1, \cdots, n$, as illustrated in Fig. 1. That is, $\hat{\mathbf{M}}_{i,i+1}$ is the result of the incorporation of the joints into the rigid ith *composite system* whose extended mass is $\tilde{\mathbf{M}}_i$. Moreover, matrix $\hat{\mathbf{M}}_{i,i+1}$ is the *articulated-body inertia of link* i, as derived in [2], and the *state estimation error covariance* [6] that satisfies the discrete Riccati equations. Furthermore, the scalar, $\hat{m}_i$, is the moment of inertia of the ith *articulated body*.

TABLE I
Comparison of Computational Complexities

Algorithm	DOF	M	A
Proposed	n	$201n - 335$	$193n - 361$
[2]	n	$199n - 198$	$174n - 173$
[4]	n	$\frac{1}{6}n^3 + 11\frac{1}{2}n^2$ $+38\frac{1}{3}n - 47$	$\frac{1}{6}n^3 + 7n^2$ $+38\frac{5}{6}n - 46$

[2]: Algorithm of [2]; [4]: Algorithm of [4].

IV. Forward Dynamics Algorithm

As an illustration, a forward dynamics algorithm is presented whose order of complexity is $n, \mathcal{O}(n)$. Assume that the equations of motion for the n-DOF manipulator, as shown in Fig. 1, are

$$\mathbf{I}\ddot{\boldsymbol{\theta}} = \boldsymbol{\tau} \tag{14}$$

where $\mathbf{I}$ is the GIM whose elements are given by (8), and $\ddot{\boldsymbol{\theta}} \equiv [\ddot{\theta}_1, \cdots, \ddot{\theta}_n]^T$ is the n-dimensional vector of joint accelerations. Moreover, $\boldsymbol{\tau}$ denotes the n-dimensional vector of torques or forces due to known external moments and forces, and those, arising from the gravity, centrifugal and Coriolis accelerations. Vector $\boldsymbol{\tau}$ is assumed to be computed efficiently from an $\mathcal{O}(n)$ inverse dynamics algorithm, e.g., [4], while $\ddot{\boldsymbol{\theta}} = \mathbf{0}$. Thus, in order to render an $\mathcal{O}(n)$ forward dynamics algorithm, it is necessary to solve $\ddot{\boldsymbol{\theta}}$ from (14) with $\mathcal{O}(n)$ computations. This can be done using the proposed decomposition, i.e., by recursively solving the following three sets of equations, which are obtained from the substitution of (13) into (14), namely

$$\mathbf{U}\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}, \quad \mathbf{D}\tilde{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}, \quad \text{and} \quad \mathbf{U}^T\ddot{\boldsymbol{\theta}} = \tilde{\boldsymbol{\tau}}. \tag{15}$$

The recursive scheme is as follows.

1) *Solution for* $\hat{\boldsymbol{\tau}}$: The components of the vector, $\hat{\boldsymbol{\tau}} = \mathbf{U}^{-1}\boldsymbol{\tau}, \hat{\tau}_i$, for $i = n-1, \cdots, 1$, is evaluated as

$$\hat{\tau}_i = \tau_i - \mathbf{p}_i^T \tilde{\boldsymbol{\eta}}_{i,i+1} \tag{16}$$

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

$$\tilde{\boldsymbol{\eta}}_{i,i+1} \equiv \mathbf{B}_{i+1,i}^T \tilde{\boldsymbol{\eta}}_{i+1} \quad \text{and} \quad \tilde{\boldsymbol{\eta}}_{i+1} \equiv \hat{\boldsymbol{\psi}}_{i+1}\hat{\tau}_{i+1} + \tilde{\boldsymbol{\eta}}_{i+1,i+2} \tag{17}$$

in which $\tilde{\boldsymbol{\eta}}_{n,n+1} = \mathbf{0}$, and $\hat{\boldsymbol{\psi}}_{i+1}$ is given in (25).

2) *Solution for* $\tilde{\boldsymbol{\tau}}$: The solution of the equation, $\mathbf{D}\tilde{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}$, is simple since $\mathbf{D}$ is a diagonal matrix. The solution, $\tilde{\tau}_i$, for $i = 1, \cdots, n$, is given by

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

3) *Solution for* $\ddot{\boldsymbol{\theta}}$: In this step, $\ddot{\boldsymbol{\theta}} \equiv \mathbf{U}^{-T}\tilde{\boldsymbol{\tau}}$, is calculated as $\mathbf{E}^T\tilde{\boldsymbol{\tau}}$, i.e., for $i = 2, \cdots, n$

$$\ddot{\theta}_i = \tilde{\tau}_i - \hat{\boldsymbol{\psi}}_i^T \tilde{\boldsymbol{\mu}}_{i,i-1} \tag{18}$$

where $\ddot{\theta}_1 \equiv \tilde{\tau}_1, \hat{\boldsymbol{\psi}}_i$ is defined in (25), and the 6-D vector, $\tilde{\boldsymbol{\mu}}_{i,i-1}$, is obtained from

$$\tilde{\boldsymbol{\mu}}_{i,i-1} \equiv \mathbf{B}_{i,i-1}\tilde{\boldsymbol{\mu}}_{i-1} \quad \text{and} \quad \tilde{\boldsymbol{\mu}}_{i-1} \equiv \mathbf{p}_{i-1}\ddot{\theta}_{i-1} + \tilde{\boldsymbol{\mu}}_{i-1,i-2} \tag{19}$$

in which $\tilde{\boldsymbol{\mu}}_{10} = \mathbf{0}$.

The computational complexity of the proposed algorithm is computed in terms of divisions or multiplications (M) and subtractions or additions (A), and compared with those of [2], [4] in Table I.

The comparison shows that the complexity of the proposed scheme is very similar to that of [2].

V. Conclusion

Based on the well-known method of decomposing a matrix using Gaussian elimination (GE), a *symbolic* decomposition of the manipulator matrix, i.e., the generalized inertia matrix (GIM) of a serial manipulator is proposed. The approach has the following features:

i) Instead of finding the values of the elements of the GIM, as done in a numerical approach, e.g., in [4], each element is found here as an analytical expression, (8).

ii) Since the natural orthogonal complement (NOC) matrix of [7] does not allow the symbolic representation of the GIM, the representation of the NOC as two decoupled matrices, as indicated after (3), is used.

iii) The rules of the GE are applied to the symbolic representation of the elements of the GIM. As a consequence, the recursive relations between the elements of the decomposed matrices, $\mathbf{U}$ and $\mathbf{D}$, (24)–(27), are recognizable. This is not possible in the numerical decomposition, e.g., in [4].

iv) An $\mathcal{O}(n)$ simulation algorithm is developed in Section IV.

v) Each step of the proposed decomposition is provided with corresponding physical interpretations, as after (6) and (13).

Finally, it is pointed out that even if there are similarities in the results and interpretations of this paper to those in [2], [6], the present approach is different, and its originalities are stated in the features, i)–iii) above.

Appendix: Reverse Gaussian Elimination

Conventionally, the Gaussian elimination (GE) of a matrix [1] starts with the annihilation of the last $(n-1)$ elements of the first column, wheras in the proposed reverse Gaussian elimination (RGE) annihilation starts with the last column, i.e., make first $(n-1)$ elements of the last column zero.

To perform the RGE, the *elementary upper triangular matrix* (EUTM) [8] is introduced that is similar to the *elementary lower triangular matrix* (ELTM) [1] in the GE. An EUTM of order n and index k, denoted by $\mathbf{E}_k$, is defined as

$$\mathbf{E}_k \equiv \mathbf{1} - \boldsymbol{\alpha}_k \boldsymbol{\lambda}_k^T \tag{20}$$

where $\mathbf{1}$ is the $n \times n$ identity matrix and the n-dimensional vectors, $\boldsymbol{\alpha}_k$ and $\boldsymbol{\lambda}_k$, are defined by

$$\boldsymbol{\alpha}_k \equiv [\alpha_{1k}, \cdots, \alpha_{k-1,k},\ 0, \cdots, 0]^T \tag{21}$$

$$\boldsymbol{\lambda}_k \equiv [0, \cdots, 0,\ 1, \cdots, 0]^T. \tag{22}$$

Now, the RGE of the GIM, $\mathbf{I}$, whose elements are given by (8), is obtained as

$$\mathbf{L}_k = \mathbf{E}_k \mathbf{L}_{k+1} \tag{23}$$

where, $k = n, \cdots, 2$, and $\mathbf{L}_{n+1} \equiv \mathbf{I}$. The elements of $\mathbf{E}_k$ and $\mathbf{L}_k$, α_{ik} and $i_{ij}^{(k)}$, respectively, are then computed from the following scheme:

- For $k = n, \cdots, 2$; Do $i = k-1, \cdots, 1$; Do $j = i, \cdots, 1$

$$\alpha_{ik} = \mathbf{p}_i^T \boldsymbol{\psi}_{ik} \quad \text{and} \quad i_{ij}^{(k)} = \mathbf{p}_i^T \hat{\mathbf{M}}_{ik} \mathbf{B}_{ij} \mathbf{p}_j \tag{24}$$

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

In (24), matrix $\mathbf{B}_{ij}$, and vectors $\mathbf{p}_i$ or $\mathbf{p}_j$ are defined in (6), whereas the 6-D vector $\hat{\boldsymbol{\psi}}_{ik}$ and the associated terms are given below

$$\hat{\boldsymbol{\psi}}_k \equiv \hat{\mathbf{M}}_{k,k+1}\mathbf{p}_k, \quad \hat{\boldsymbol{\psi}}_{ik} \equiv \mathbf{B}_{ki}^T \hat{\boldsymbol{\psi}}_k, \quad \hat{m}_k \equiv \mathbf{p}_k^T \hat{\boldsymbol{\psi}}_k, \tag{25}$$

$$\boldsymbol{\psi}_k \equiv \frac{\hat{\boldsymbol{\psi}}_k}{\hat{m}_k}, \quad \text{and} \quad \boldsymbol{\psi}_{ik} \equiv \frac{\hat{\boldsymbol{\psi}}_{ik}}{\hat{m}_k}. \tag{26}$$

The 6 × 6 matrix, $\hat{\mathbf{M}}_{ik}$ of (24) or $\hat{\mathbf{M}}_{k,k+1}$ of (25), being evaluated recursively for, $k = n, \cdots, 2; i = k, \cdots, 1$, as

$$\hat{\mathbf{M}}_{ik} = \begin{cases} \hat{\mathbf{M}}_{k,k+1} - \boldsymbol{\psi}_k \hat{\boldsymbol{\psi}}_k^T & \text{if } i = k \\ \mathbf{M}_i + \mathbf{B}_{i+1,i}^T \hat{\mathbf{M}}_{i+1,k} \mathbf{B}_{i+1,i} & \text{otherwise} \end{cases} \tag{27}$$

in which $\hat{\mathbf{M}}_{n,n+1} \equiv \mathbf{M}_n$, while $k = n$.

References

[1] G. E. Stewart, *Introduction to Matrix Computations*. New York: Academic, 1973.

[2] R. Featherstone, “The calculation of robot dynamics using articulated-body inertias,” *Int. J. Robot. Res.*, vol. 2, no. 1, pp. 13–30, 1983.

[3] W. W. Armstrong, “Recursive solution to the equations of motion of an n-link manipulator,” in *Proc. 5th World Cong. Theory Mach. Mech.*, ASME, Montreal, Canada, 1979, vol. 2, pp. 1343–1346.

[4] M. W. Walker and D. E. Orin, “Efficient dynamic computer simulation of robotic mechanisms,” *ASME J. Dyn. Sys., Measure., Contr.*, vol. 104, pp. 205–211, Sept. 1982.

[5] B. P. Cloutier, D. K. Pai, and U. M. Ascher, “The formulation stiffness of forward dynamics algorithms and implications for robot simulation,” in *Proc. IEEE Conf. Robot. Automat.*, Nagoya, Japan, May 21–27, 1995, vol. 3, pp. 2816–2822.

[6] G. Rodriguez, “Kalman filtering, smoothing, and recursive robot arm forward and inverse dynamics,” *IEEE Trans. Robot. Automat.*, vol. RA-3, pp. 624–639, 1987.

[7] J. Angeles and S. Lee, “The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement,” *ASME J. Appl. Mech.*, vol. 55, pp. 243–244, Mar. 1988.

[8] S. K. Saha, “The $\mathbf{UDU}^T$ decomposition of manipulator inertia matrix,” in *Proc. IEEE Conf. Robot. Automat.*, Nagoya, Japan, May 21–27, 1995, vol. 3, pp. 2829–2834.

[9] K. W. Lilly and D. E. Orin, “Alternate formulation for the manipulator inertia matrix,” *Int. J. Robot. Res.*, vol. 10, no. 1, pp. 64–74, 1991.