Dynamics of Serial Multibody Systems Using the Decoupled Natural Orthogonal Complement Matrices

S. K. Saha[1]
Department of Mechanical Engineering,
I.I.T., Delhi, Hauz Khas,
New Delhi 110 016, India
e-mail: saha@mech.iitd.ernet.in

Constrained dynamic equations of motion of serial multibody systems consisting of rigid bodies in a serial kinematic chain are derived in this paper. First, the Newton-Euler equations of motion of the decoupled rigid bodies of the system at hand are written. Then, with the aid of the decoupled natural orthogonal complement (DeNOC) matrices associated with the velocity constraints of the connecting bodies, the Euler-Lagrange independent equations of motion are derived. The De NOC is essentially the decoupled form of the natural orthogonal complement (NOC) matrix, introduced elsewhere. Whereas the use of the latter provides recursive order n–n being the degrees-of-freedom of the system at hand—inverse dynamics and order n^3 forward dynamics algorithms, respectively, the former leads to recursive *order n algorithms for both the cases. The order n algorithms are desirable not only for their computational efficiency but also for their numerical stability, particularly, in forward dynamics and simulation, where the system's accelerations are solved from the dynamic equations of motion and subsequently integrated numerically. The algorithms are illustrated with a three-link three-degrees-of-freedom planar manipulator and a six-degrees-of-freedom Stanford arm.*

1 Introduction

The conventional approach to obtain the dynamic model, i.e., equations of motion, of a mechanical system, consisting of rigid bodies coupled by kinematic pairs or joints, is to use either Newton-Euler (NE) or Euler-Lagrange (EL) equations. While the NE equations are obtained from the free-body diagrams, the EL equations resulted from the kinetic and potential energy of the system. The former is not suitable for motion simulation, as it finds the internal forces and torques that do not affect the motion of the system. Alternatively, EL equations give independent set of equations that are good for motion simulation, however, require complex calculations for the partial derivatives. With the advent of digital computation, a series of new methods in the study of dynamic modeling of mechanical systems have been developed. Huston and Passerello (1974) introduced the first computer-oriented method to reduce the dimension of the unconstrained dynamical equations, namely, Newton-Euler equations of motion, by eliminating the constraint forces. What was basically used to reduce the dimension is an *orthogonal complement* of the velocity constraints of the mechanical system under study. An orthogonal complement is defined as the matrix whose columns span the null-space of the matrix of the velocity constraints, and, hence, the premultiplication of the transpose of it with the unconstrained dynamic equations of motion vanishes the constrained moments and forces. The said orthogonal complement is not unique. In some approaches, an orthogonal complement is found using numerical schemes which are of an intensive nature, requiring, for example, singular-value decomposition or eigenvalue computations (Wehage and Haug, 1982; Kamman and Huston, 1984). In Angeles and Lee (1988), Angeles and Ma (1988), and Saha and Angeles (1991), however, the complement is obtained naturally from the velocity constraint expressions without any complex computation. Therefore, the complement is called *natural orthogonal complement* (NOC). The NOC is the matrix that relates the angular and linear velocities of the rigid bodies in the system to the associated joint rates, as in Eq. (16).

The purpose of deriving a dynamic model here is to obtain inverse and forward dynamics algorithms required in control and simulation of the mechanical systems under study, respectively. While almost all proposed inverse dynamics algorithms for serial-chain mechanical systems, as shown in Fig. 1, are recursive of order n, $\mathcal{O}(n)$—n being the degrees-of-freedom of the system at hand—complexity, e.g., Hollerbach (1980), Luh, Walker and Paul (1980), Kane and Levinson (1983), Angeles et al. (1989), Sciavicco and Siciliano (1996), most of the traditional forward dynamics algorithms are of $\mathcal{O}(n^3)$ (Walker and Orin, 1982; Kane and Levinson, 1983; Angeles and Ma, 1988). Note here that in inverse dynamics the explicit derivation of the equations of motion in terms of the generalized coordinates is not required, e.g., in Sciavicco and Siciliano (1996), where the joint torques are calculated from the trajectory of the end-effector. Alternatively, explicit expressions for the matrices, particularly, the generalized inertia matrix associated with the equations of motion must be evaluated for forward dynamics, e.g., in Angeles and Ma (1988) and others, in which the joint accelerations are solved from the equations of motion. Conventionally, a numerical approach is taken to solve for the joint accelerations. For example, in Angeles and Ma (1988), (i) first, the elements of the inertia matrix are evaluated; (ii) then, the numerical decomposition of the matrix, namely, the Cholesky decomposition (Stewart, 1973), is performed; and (iii) finally, the joint accelerations are solved by backward and forward substitutions. Since the complexity of the Cholesky decomposition is of order n^3, the forward dynamics algorithm also requires $\mathcal{O}(n^3)$ computations. There is, however, an alternative approach to obtain a forward dynamics algorithm recursively, whose computational complexity is in the order n, i.e., $\mathcal{O}(n)$, e.g., Armstrong (1979),

[1] Presently (May 1999–Jan. 2000) at the Institute B of Mechanics, University of Stuttgart, Stuttgart, Germany as a Humbold Fellow.

Contributed by the Applied Mechanics Division of The American Society of Mechanical Engineers for publication in the ASME Journal of Applied Mechanics.

Discussion on the paper should be addressed to the Technical Editor, Professor Lewis T. Wheeler, Department of Mechanical Engineering, University of Houston, Houston, TX 77204-4792, and will be accepted until four months after final publication of the paper itself in the ASME Journal of Applied Mechanics.

Manuscript received by the ASME Applied Mechanics Division, July 14, 1998; final revision, June 17, 1999. Associate Technical Editor: N. C. Perkins.

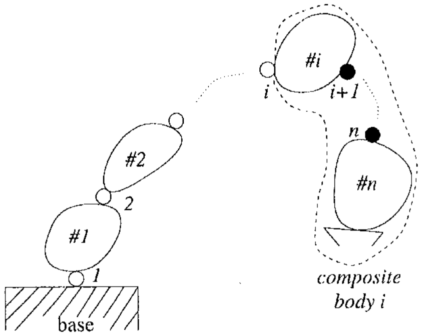

Fig. 1 An *n*-body serial manipulator

Featherstone (1983), Rodriguez (1987), Schiehlen (1991), Rodriguez and Kreutz-Delgado (1992), and Saha (1997). There is also a parallel $\mathcal{O}(\log N)$ algorithm proposed in Fijany et al. (1995) which requires a special computer hardware, namely, a parallel architecture. Because of the special hardware requirement the parallel algorithms will not be discussed further.

The algorithm by Armstrong (1979) was the first of its kind where the recursive solution to the equations of motion of an n-link manipulator mounted on a spacecraft is obtained. Featherstone (1983) has introduced the concept of *articulated body inertia* (ABI) which has been efficiently evaluated by McMillan and Orin (1995). The ABI is derived by extending the expressions for the equations of motion of a single body to a system of coupled bodies. Rodriguez (1987), on the other hand, presented a novel robot inverse and forward dynamics algorithms based on Kalman filtering and smoothing technique arising in the state estimation theory. The methodology has been used to solve several other dynamics problems as well, namely, by Rodriguez, Jain, and Kreutz-Delgado (1991), and Jain and Rodriguez (1995). Schiehlen (1991) and Saha (1997), however, have independently used the linear algebra approach to arrive at a recursive forward dynamics algorithm. It is interesting to note that, compared to the $\mathcal{O}(n^3)$ schemes, e.g., Angeles and Ma (1988), computational complexity of all the $\mathcal{O}(n)$ schemes, wherever known, are efficient when $n \geq 12$, as is evident from Table 2. The $\mathcal{O}(n)$ algorithms, however, calculate the joint accelerations that are smooth functions of time (Ascher et al., 1997). As a result, numerical integration is faster and, hence, the total time in simulation may be less while comparing with the $\mathcal{O}(n^3)$ scheme. Besides, $\mathcal{O}(n)$ algorithms provide physical interpretations of the parameters they calculate and the intermediate steps they follow. Thus, the recursive forward dynamics is becoming more and more popular.

In this paper, the NOC (Angeles and Lee, 1988) is taken one step further to *uniformly* develop a set of independent dynamic equations of motion from which both recursive inverse and forward dynamics algorithms can be written. Here, the NOC is expressed as a product of two matrices, one is a block triangular and the other one is a block diagonal. Hence, the term *decoupled* added to it. Unlike the NOC, its modified form, i.e., the DeNOC, allows one to write expressions of the elements of the matrices and vectors associated with the dynamic equations of motion in *analytical, recursive* form, as shown in Section 4. These characteristics were not so obvious with the earlier form of the NOC, e.g., in Saha and Angles (1991). Moreover, many of the physical interpretations obtained in this paper have corresponding interpretations in Featherstone (1983), Rodriguez (1987), Rodriguez and Kreutz-Delgado (1992), and others, as indicated after Eqs. (14), (17), and (50). The present approach provides an alternate approach to the development of a recursive forward dynamics algorithm, compared to others, e.g., Rodriguez (1987). The present approach is built upon basic mechanics and linear algebra theory. Hence, it is easy to comprehend, even by a fresh mechanical engineer without any knowledge of control theory. Moreover, there is equivalence of many steps and interpreted variables obtained from Gaussian elimination of the inertia matrix, as shown in Appendix B, and Kalman filtering (Rodriguez, 1987). As a result, it may be commented that they are equivalent under certain conditions. The real proof is, however, behind the scope of this paper.

The contributions of this paper are listed as follows.

1 A systematic *uniform* development of the dynamic model of a serial-chain mechanical system. Here the word, uniform, implies that *no* different sets of equations of motion are required to obtain inverse and forward dynamics algorithms, as in Sciavicco and Siciliano (1996) and others, where Newton-Euler equations of the motion of the uncoupled bodies are used to obtain a recursive inverse dynamics algorithm and the joint accelerations (forward dynamics) are computed from the independent set of equations.

2 Recursive expressions not only for the generalized inertia matrix (Saha, 1997), but also for the generalized matrix of convective inertia terms, and the generalized forces/torques, as shown in Section 4.

3 As a consequence of the above items, not only a forward dynamics algorithm (Saha, 1997) but also an inverse dynamics algorithm is possible from the same set of dynamics equations of motion.

4 Implementation of the above algorithms to a planar three-degrees-of-freedom manipulator and a six-degrees-of-freedom Stanford arm.

5 Physical interpretation of different terms obtained during the dynamic modeling, and correlation of those with the similar terms appeared in the literature, e.g., Featherstone (1983) and Rodriguez (1987) and Rodriguez and Kreutz-Delgado (1992).

The paper is organized as follows: Section 2 introduces some definitions required to derive the dynamic model given in Section 3. Section 4 shows how to derive the analytical expressions for the matrices and vectors associated with the equations of motion, whereas Section 5 gives the applications, namely, inverse and forward dynamics algorithms. The results of the algorithms with respect to two examples are provided in Section 6.

2 Some Definitions

Referring to Fig. 1, it is assumed that a serial multibody system, namely, the serial manipulator, has a fixed "base" and n moving rigid bodies, numbered #1, . . . , #n, coupled by n one-degree-of-freedom kinematic pairs or joints, say, a prismatic or a revolute, which are indicated in Fig. 1 as 1, . . . , n. The ith joint couples the ith body with that of the $(i-1)$st. Now, referring to the motion of the ith body, Fig. 2, the following terms are defined:

$\boldsymbol{t}_i$ *and* $\boldsymbol{n}_i$*:* the six-dimensional vectors of *twist* and *wrench* of the ith link, i.e.,

$$\mathbf{t}_i \equiv \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix} \quad \text{and} \quad \mathbf{w}_i \equiv \begin{bmatrix} \mathbf{n}_i \\ \mathbf{f}_i \end{bmatrix} \tag{1}$$

where $\boldsymbol{\omega}_i$ and $\mathbf{v}_i$ are the three-dimensional vectors of angular velocity and the linear velocity of the mass center of the ith link, C_i, as shown in Fig. 2, respectively, whereas $\mathbf{n}_i$ and $\mathbf{f}_i$ are the three-dimensional vectors of the moment about C_i and the force at C_i, respectively.

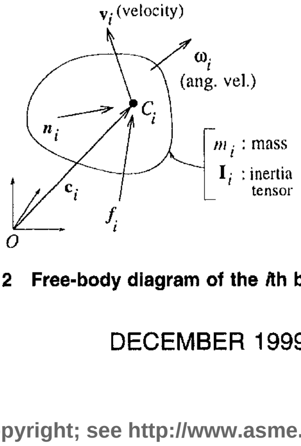

Fig. 2 Free-body diagram of the *i*th body

$\mathbf{\Omega}_i$ *and* $\mathbf{M}_i$: the 6 × 6 matrices of *angular velocity* and *mass* of the *i*th body, respectively, namely,

$$\mathbf{\Omega}_i \equiv \begin{bmatrix} \boldsymbol{\omega}_i \times \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \boldsymbol{\omega}_i \times \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{M}_i \equiv \begin{bmatrix} \mathbf{I}_i & \mathbf{O} \\ \mathbf{O} & m_i\mathbf{1} \end{bmatrix} \tag{2}$$

where $\boldsymbol{\omega}_i \times \mathbf{1}$ is the 3 × 3 cross-product tensor associated with vector $\boldsymbol{\omega}_i$, which when operates on any three-dimensional Cartesian vector, $\mathbf{x}$, results in a cross-product vector, i.e., $(\boldsymbol{\omega}_i \times \mathbf{1})\mathbf{x} \equiv \boldsymbol{\omega}_i \times \mathbf{x}$. Moreover, $\mathbf{I}_i$ and m_i are the 3 × 3 inertia tensor about the mass center, C_i, and the mass of the *i*th body, respectively. Furthermore, $\mathbf{1}$ and $\mathbf{O}$ are the 3 × 3 identity matrix and the matrix of zeros, respectively. Henceforth, the dimensions of $\mathbf{1}$ and $\mathbf{O}$ should be understood as of dimensions that are compatible with the dimensions of the matrix expressions where they appear.

t and w: the 6*n*-dimensional vectors of generalized twist and wrench, respectively, i.e.,

$$\mathbf{t} \equiv [\mathbf{t}_1^T, \ldots, \mathbf{t}_n^T]^T \quad \text{and} \quad \mathbf{w} \equiv [\mathbf{w}_1^T, \ldots, \mathbf{w}_n^T]^T \tag{3}$$

where $\mathbf{t}_i$ and $\mathbf{w}_i$, for $i = 1, \ldots, n$, are defined in Eq. (1).

$\mathbf{\Omega}$ *and M:* the 6*n* × 6*n* matrices of generalized angular velocity and mass, respectively, namely,

$$\mathbf{\Omega} \equiv \text{diag}\,[\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_n] \quad \text{and} \quad \mathbf{M} \equiv \text{diag}\,[\mathbf{M}_1, \ldots, \mathbf{M}_n] \tag{4}$$

where $\mathbf{\Omega}_i$ and $\mathbf{M}_i$, for $i = 1, \ldots, n$, are given in Eq. (2).

$\dot{\boldsymbol{\theta}}$: the *n*-dimensional independent vector of joint rates, i.e.,

$$\dot{\boldsymbol{\theta}} \equiv [\dot{\theta}_1, \ldots, \dot{\theta}_n]^T \tag{5}$$

where θ_i is the displacement of the *i*th joint, as shown in Fig. 3.

3 Dynamic Modeling Using the DeNOC

For a serial-chain mechanical system shown in Fig. 1, the steps to obtain the dynamic equations of motion using the DeNOC matrices are given below:

1 Derivation of the Unconstrained Newton-Euler (NE) Equations of Motion:

(a) From the free-body diagram of the *i*th rigid body, Fig. 2, write the Newton-Euler equations of motion as

$$\mathbf{I}_i\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \mathbf{I}_i\boldsymbol{\omega}_i = \mathbf{n}_i \tag{6a}$$

$$m_i\dot{\mathbf{v}}_i = \mathbf{f}_i \tag{6b}$$

where $\mathbf{n}_i$ and $\mathbf{f}_i$ are the resultant moment about and force applied at the mass center, C_i, respectively. Using the definitions given in Eqs. (1) and (2), the above six scalar equations can be put in a compact form as

$$\mathbf{M}_i\dot{\mathbf{t}}_i + \dot{\mathbf{M}}_i\mathbf{t}_i = \mathbf{w}_i \tag{7}$$

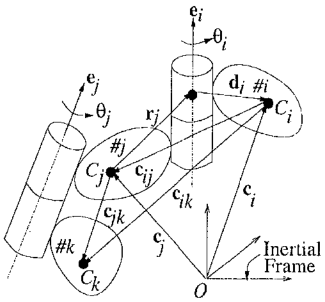

Fig. 3 A coupled system

where the six-dimensional *twist-rate* vector, $\dot{\mathbf{t}}_i$, and the 6 × 6 *mass-rate* matrix, $\dot{\mathbf{M}}_i$, are, respectively, the time derivative of $\mathbf{t}_i$ and $\mathbf{M}_i$, which are defined as

$$\dot{\mathbf{t}}_i \equiv \begin{bmatrix} \dot{\boldsymbol{\omega}}_i \\ \dot{\mathbf{v}}_i \end{bmatrix} \quad \text{and} \quad \dot{\mathbf{M}}_i \equiv \begin{bmatrix} \boldsymbol{\omega}_i \times \mathbf{I}_i & \mathbf{O} \\ O & \mathbf{O} \end{bmatrix}. \tag{8}$$

(b) Write Eq. (7) for $i = 1, \ldots, n$: This gives 6*n* uncoupled scalar equations for the whole system, which are put in the following form:

$$\mathbf{M}\dot{\mathbf{t}} + \dot{\mathbf{M}}\mathbf{t} = \mathbf{w} \tag{9}$$

where $\dot{\mathbf{t}}$ and $\dot{\mathbf{M}}$ are the time derivative of the generalized twist, $\mathbf{t}$, and generalized mass, $\mathbf{M}$, respectively, i.e.,

$$\dot{\mathbf{t}} \equiv [\dot{\mathbf{t}}_1^T, \ldots, \dot{\mathbf{t}}_n^T]^T \quad \text{and} \quad \dot{\mathbf{M}} \equiv \text{diag}\,[\dot{\mathbf{M}}_1, \ldots, \dot{\mathbf{M}}_n] \tag{10}$$

in which $\dot{\mathbf{t}}_i$ and $\dot{\mathbf{M}}_i$ appear in Eq. (8).

2 Derivation of Kinematic Constraints:

(a) In this step, motion constraints due to the kinematic pairs are derived. For example, velocity constraints between two links, #*i* and #*j*, coupled by the *i*th revolute joint, as shown in Fig. 3, can be expressed as

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_j + \dot{\theta}_i\mathbf{e}_i \tag{11a}$$

$$\mathbf{v}_i = \mathbf{v}_j + \boldsymbol{\omega}_j \times \mathbf{r}_j + \boldsymbol{\omega}_i \times \mathbf{d}_i. \tag{11b}$$

The above six scalar equations are written in a compact form as

$$\mathbf{t}_i = \mathbf{B}_{ij}\mathbf{t}_j + \mathbf{p}_i\dot{\theta}_i \tag{12}$$

where the 6 × 6 matrix, $\mathbf{B}_{ij}$, and the six-dimensional vector, $\mathbf{p}_i$, are as follows:

$$\mathbf{B}_{ij} \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{c}_{ij} \times \mathbf{1} & \mathbf{1} \end{bmatrix} \quad \text{and} \quad \mathbf{p}_i \equiv \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_i \times \mathbf{d}_i \end{bmatrix} \tag{13}$$

$\mathbf{c}_{ij} \times \mathbf{1}$ being the cross-product tensor associated with vector $\mathbf{c}_{ij}$, defined similar to $\boldsymbol{\omega}_i \times \mathbf{1}$ of Eq. (2), i.e., $(\mathbf{c}_{ij} \times \mathbf{1})\mathbf{x} \equiv \mathbf{c}_{ij} \times \mathbf{x}$, for any arbitrary three-dimensional Cartesian vector $\mathbf{x}$. The vector, $\mathbf{c}_{ij}$, as shown in Fig. 3, is given by, $\mathbf{c}_{ij} \equiv -\mathbf{d}_i - \mathbf{r}_j$. It is interesting to note here that the matrix, $\mathbf{B}_{ij}$, and the vector, $\mathbf{p}_i$, have the following interpretations:

- For two rigidly connected moving bodies, #*i* and #*j*, $\mathbf{B}_{ij}$, propagates the twist of #*j* to #*i*. Hence, matrix $\mathbf{B}_{ij}$ is termed here as the *twist propagation matrix,* which has the following properties:

$$\mathbf{B}_{ij}\mathbf{B}_{jk} = \mathbf{B}_{ik}, \quad \mathbf{B}_{ii} = \mathbf{1}, \quad \text{and} \quad \mathbf{B}_{ij}^{-1} = \mathbf{B}_{ji}. \tag{14}$$

  Matrix $\mathbf{B}_{ij}$ is nothing but the *state transition* matrix of Rodriguez (1987).
- Vector, $\mathbf{p}_i$, on the other hand, takes into account the motion of the *i*th joint. Hence, $\mathbf{p}_i$ is termed as the *joint-rate propagation vector,* which is dependent on the type of joint. For example, the expression of $\mathbf{p}_i$ in Eq. (13) is for a revolute joint shown in Fig. 3, whereas for a prismatic joint vector $\mathbf{p}_i$ is given by

$$\mathbf{p}_i \equiv \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i \end{bmatrix}: \quad \text{for a prismatic joint} \tag{15}$$

  where $\mathbf{e}_i$ is the unit vector parallel to the axis of linear motion. Correspondingly, $\dot{\theta}_i$ of Eq. (12) would mean the linear joint rate. Other joints are not treated here because any other joint, e.g., spherical or screw, can be treated as combination of revolute or revolute and prismatic pairs, respectively.

(b) Write Eq. (12), for $i = 1, \ldots, n$, as

$$\mathbf{t} = \mathbf{N}\dot{\boldsymbol{\theta}}, \quad \text{where} \quad \mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_d. \tag{16}$$

In Eq. (16), $\mathbf{N}$ is the $6n \times n$ NOC matrix, introduced in Angeles and Lee (1988), whereas its decoupled form, $\mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_d$, is termed as the *decoupled* NOC (DeNOC) matrices (Saha, 1997). The $6n \times 6n$ lower block triangular matrix, $\mathbf{N}_l$, and the $6n \times n$ block diagonal matrix, $\mathbf{N}_d$, are given by

$$\mathbf{N}_l \equiv \begin{bmatrix} \mathbf{1} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{B}_{21} & \mathbf{1} & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{B}_{n1} & \mathbf{B}_{n2} & \cdots & \mathbf{1} \end{bmatrix} \quad \text{and}$$

$$\mathbf{N}_d \equiv \begin{bmatrix} \mathbf{p}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{p}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{p}_n \end{bmatrix} \tag{17}$$

where the first two properties of the twist propagation matrix, $\mathbf{B}_{ij}$, given in Eq. (14) are used. Moreover, $\mathbf{O}$ and $\mathbf{0}$ are the 6×6 matrix and the six-dimensional vector of zeros, respectively. Matrices $\mathbf{N}_l$ and $\mathbf{N}_d$ are the *spatial operators* of Rodriguez and Kreutz-Delgado (1992), which are "anticausal" and "memory-less" nonrecursive, respectively.

3 Incorporation of the Kinematic Constraints Into the NE Equations of Motion: Premultiply $\mathbf{N}^T$ with the decoupled NE equations of motions, Eq. (9), i.e.,

$$\mathbf{N}^T(\mathbf{M}\dot{\mathbf{t}} + \dot{\mathbf{M}}\mathbf{t}) = \mathbf{N}^T(\mathbf{w}^E + \mathbf{w}^I) \tag{18}$$

where $\mathbf{w}$ is substituted as $\mathbf{w} \equiv \mathbf{w}^E + \mathbf{w}^I$—$\mathbf{w}^E$ and $\mathbf{w}^I$ are the generalized external and internal constraint wrenches, respectively. Since constraint moments and forces do not perform any work, $\mathbf{t}^T\mathbf{w}^I$ vanishes, i.e., $\mathbf{t}^T\mathbf{w}^I = \dot{\boldsymbol{\theta}}^T\mathbf{N}^T\mathbf{w}^I = \mathbf{0}$, which implies that, for independent $\dot{\boldsymbol{\theta}}$, $\mathbf{N}^T\mathbf{w}^I = \mathbf{0}$. Hence, Eq. (18) is rewritten as

$$\mathbf{N}^T(\mathbf{M}\dot{\mathbf{t}} + \dot{\mathbf{M}}\mathbf{t}) = \mathbf{N}^T\mathbf{w}^E. \tag{19}$$

Note that in order to eliminate the constraint wrench from the decoupled NE equations, Eq. (9), as done in Eq. (19), it is important to write the generalized twist, $\mathbf{t}$, as a linear transformation of the *independent* generalized speed, as in Eq. (16).

4 Constrained Euler-Lagrange (EL) Dynamic Equations of Motion: Substitute the expression of $\mathbf{t}$, Eq. (16), and its time derivative, i.e., $\dot{\mathbf{t}} = \mathbf{N}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{N}}\dot{\boldsymbol{\theta}}$, into Eq. (19). The result is the following constrained dynamic equations of motion:

$$\mathbf{I}\ddot{\boldsymbol{\theta}} + \mathbf{C}\dot{\boldsymbol{\theta}} = \boldsymbol{\tau} \tag{20}$$

where

$\mathbf{I} \equiv \mathbf{N}^T\mathbf{M}\mathbf{N} \equiv \mathbf{N}_d^T\tilde{\mathbf{M}}\mathbf{N}_d$: the $n \times n$ generalized inertia matrix (*GIM*);

$\mathbf{C} \equiv \mathbf{N}^T(\mathbf{M}\dot{\mathbf{N}} + \dot{\mathbf{M}}\mathbf{N}) \equiv \mathbf{N}_d^T\tilde{\mathbf{M}}'\mathbf{N}_d$: the $n \times n$ matrix of convective inertia terms;

$\boldsymbol{\tau} \equiv \mathbf{N}^T\mathbf{w}^E \equiv \mathbf{N}_d^T\tilde{\mathbf{w}}^E$: the n-dimensional vector of generalized forces due to driving torques and those resulting from gravity and dissipation, etc.

The expressions for $\tilde{\mathbf{M}}$, $\tilde{\mathbf{M}}'$, and $\tilde{\mathbf{w}}^E$ are associated with the composite body of the system under study and derived in the next section, Section 4. Note that Eq. (20) is essentially the Euler-Lagrange equations of motion that has been arrived from the uncoupled Newton-Euler equations, e.g., Eq. (9). Moreover, the use of the NOC, $\mathbf{N}$, as introduced in Angeles and Lee (1988), does not allow further simplification of associated expressions of Eq. (20), i.e., matrices $\mathbf{I}$ and $\mathbf{C}$, and vector $\boldsymbol{\tau}$. Hence, a computer algorithm is required at this stage for dynamic analysis. Alternatively, the DeNOC matrices given in Eq. (16) allows one to write the analytical *recursive* expressions for each element of the matrices, $\mathbf{I}$, $\mathbf{C}$, and $\boldsymbol{\tau}$, as shown in the following section. These expressions lead to *recursive* inverse and forward dynamics algorithms, given in Sections 5.1 and 5.2, respectively, whose computational complexities are of order n, i.e., $\mathcal{O}(n)$, and provide many physical interpretations, e.g., after Eqs. (14), (17), (23*b*), and (50).

4 Analytical Expressions for I, C, and $\boldsymbol{\tau}$

The analytical expressions for the GIM, $\mathbf{I}$, the generalized matrix of convective inertia terms, $\mathbf{C}$, and the generalized vector of joint forces/torques, as given after Eq. (20), are derived next. These expressions provide the basis for *uniform* development of the *recursive* inverse and forward dynamics algorithms. That is, unlike other approaches, e.g., Sciavicco and Siciliano (1996) and others, where free-body diagrams and NE equations are used to develop inverse dynamics algorithm and the EL equations are used for forward dynamics, here only one set of equations, namely, Eqs. (20), (24), (29), and (30), are used for both the algorithms.

4.1 Derivation of the GIM, I. Substituting the expression for the NOC in terms of the DeNOC matrices, i.e., $\mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_d$, as in Eq. (17), into that of matrix $\mathbf{I} \equiv \mathbf{N}^T\mathbf{M}\mathbf{N}$, as given after Eq. (20), one obtains

$$\mathbf{I} \equiv \mathbf{N}_d^T\tilde{\mathbf{M}}\mathbf{N}_d, \quad \text{where} \quad \tilde{\mathbf{M}} \equiv \mathbf{N}_l^T\mathbf{M}\mathbf{N}_l \tag{21}$$

in which the $6n \times 6n$ symmetric matrix, $\tilde{\mathbf{M}}$, is given by

$$\tilde{\mathbf{M}} \equiv \begin{bmatrix} \tilde{\mathbf{M}}_1 & \mathbf{B}_{21}^T\tilde{\mathbf{M}}_2 & \cdots & \mathbf{B}_{n1}^T\tilde{\mathbf{M}}_n \\ \tilde{\mathbf{M}}_2\mathbf{B}_{21} & \tilde{\mathbf{M}}_2 & \cdots & \mathbf{B}_{n2}^T\tilde{\mathbf{M}}_n \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{M}}_n\mathbf{B}_{n1} & \tilde{\mathbf{M}}_n\mathbf{B}_{n2} & \cdots & \tilde{\mathbf{M}}_n \end{bmatrix} \tag{22}$$

where the 6×6 matrix, $\tilde{\mathbf{M}}_i$, for $i = 1, \ldots, n$, is defined as

$$\tilde{\mathbf{M}}_i \equiv \mathbf{M}_i + \sum_{k=i+1}^{n} \mathbf{B}_{ki}^T\mathbf{M}_k\mathbf{B}_{ki}. \tag{23a}$$

Note the summation over $k = i + 1, \ldots, n$ in Eq. (23*a*). An algorithm computing $\tilde{\mathbf{M}}_i$, for $i = 1, \ldots, n$, from Eq. (23*a*) will certainly require order n^2 computations. However, a close look to Eq. (23*a*) with the first two properties of $\mathbf{B}_{ij}$, Eq. (14), in mind reveals that the summation expression can be evaluated recursively, i.e., $\sum_{k=i+1}^{n} \mathbf{B}_{ki}^T\mathbf{M}_k\mathbf{B}_{ki} \equiv \mathbf{B}_{i+1,i}^T\tilde{\mathbf{M}}_{i+1}\mathbf{B}_{i+1,i}$, for $i = n, \ldots, 1$, with $\tilde{\mathbf{M}}_{n+1} \equiv \mathbf{O}$, because there is no $(n + 1)$st link in the system. Hence, Eq. (23*a*) has the following recursive relation:

$$\tilde{\mathbf{M}}_i = \mathbf{M}_i + \mathbf{B}_{i+1,i}^T\tilde{\mathbf{M}}_{i+1}\mathbf{B}_{i+1,i} \quad \text{where} \quad \tilde{\mathbf{M}}_n \equiv \mathbf{M}_n. \tag{23b}$$

The 6×6 matrix, $\tilde{\mathbf{M}}_i$, is the mass matrix of *composite body, i*, i.e., $\tilde{\mathbf{M}}_i$ represents the mass and inertia properties of the system comprising of $(n - i + 1)$ rigidly connected bodies, namely, bodies #$i, \ldots,$ #n, as indicated in Fig. 1 by dotted line. The expressions for the elements of the GIM, $\mathbf{I}$ of Eq. (20), which is symmetric positive definite, are then given by

$$i_{ij} = i_{ji} = \mathbf{p}_i^T\tilde{\mathbf{M}}_i\mathbf{B}_{ij}\mathbf{p}_j \tag{24}$$

for $i = 1, \ldots, n$; $j = 1, \ldots, i$.

Table 1 Computational complexities in inverse dynamics

Algorithm	M	A	$n = 6$	
Hollerbach (1980)	$412n - 277$	$320n - 201$	$2195M$	$1719A$
Luh et al. (1980)	$150n - 48$	$131n + 48$	$852M$	$834A$
Walker and Orin (1982)	$137n - 22$	$101n - 11$	$800M$	$595A$
Proposed	$120n - 44$	$97n - 55$	$676M$	$527A$
Khalil et al. (1986)	$105n - 92$	$94n - 86$	$538M$	$478A$
Angeles et al. (1989)	$105n - 109$	$90n - 105$	$521M$	$435A$
Balafoutis et al. (1988)	$93n - 69$	$81n - 65$	$489M$	$421A$

M: Multiplication/Division; A: Addition/Subtraction

4.2 Derivation of C Matrix. Using the NOC and its time derivative in terms of the DeNOC matrices, i.e., $\mathbf{N} \equiv \mathbf{N}_l\mathbf{N}_d$, and $\dot{\mathbf{N}} \equiv \dot{\mathbf{N}}_l\mathbf{N}_d + \mathbf{N}_l\dot{\mathbf{N}}_d$, respectively, the generalized matrix of convective inertia terms, $\mathbf{C}[\equiv\mathbf{N}^T(\mathbf{M}\dot{\mathbf{N}} + \dot{\mathbf{M}}\mathbf{N})]$, as defined after Eq. (20), is obtained as

$$\mathbf{C} \equiv \mathbf{N}_d^T\tilde{\mathbf{M}}'\mathbf{N}_d, \quad \text{where} \quad \tilde{\mathbf{M}}' \equiv \mathbf{N}_l^T\mathbf{M}\dot{\mathbf{N}}_l + \tilde{\mathbf{M}}\mathbf{\Omega} + \dot{\tilde{\mathbf{M}}} \tag{25}$$

in which the following identities are used:

$$\dot{\mathbf{N}}_d \equiv \mathbf{\Omega}\mathbf{N}_d \quad \text{and} \quad \dot{\tilde{\mathbf{M}}} = \mathbf{N}_l^T\dot{\mathbf{M}}\mathbf{N}_l \tag{26}$$

where the $6n \times 6n$ matrix, $\mathbf{\Omega}$, is defined in Eq. (4). In Eq. (25), the multiplication of two block diagonal matrices, $\mathbf{M}$ and $\mathbf{\Omega}$, can be easily performed, whereas the expressions for the matrices, $\mathbf{N}_l^T\mathbf{M}\dot{\mathbf{N}}_l$ and $\dot{\tilde{\mathbf{M}}}$, are shown below:

$$\mathbf{N}_l^T\mathbf{M}\dot{\mathbf{N}}_l \equiv \begin{bmatrix} \mathbf{B}_{21}^T\tilde{\mathbf{H}}_{21} & \cdots & \mathbf{B}_{n1}^T\tilde{\mathbf{H}}_{n,n-1} & \mathbf{O} \\ \tilde{\mathbf{H}}_{21} & \cdots & \mathbf{B}_{n2}^T\tilde{\mathbf{H}}_{n,n-1} & \mathbf{O} \\ \vdots & \ddots & \vdots & \vdots \\ \tilde{\mathbf{H}}_{n-1,1} & \cdots & \mathbf{B}_{n2}^T\tilde{\mathbf{H}}_{n,n-1} & \mathbf{O} \end{bmatrix} \quad \text{and}$$

$$\dot{\tilde{\mathbf{M}}} \equiv \begin{bmatrix} \dot{\tilde{\mathbf{M}}}\dot{\tilde{\mathbf{M}}}_1 & \dot{\tilde{\mathbf{M}}}_2 & \cdots & \dot{\tilde{\mathbf{M}}}_n \\ \dot{\tilde{\mathbf{M}}}_2 & \dot{\tilde{\mathbf{M}}}_2 & \cdots & \dot{\tilde{\mathbf{M}}}_n \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\tilde{\mathbf{M}}}_n & \dot{\tilde{\mathbf{M}}}_n & \cdots & \dot{\tilde{\mathbf{M}}}_n \end{bmatrix}, \tag{27}$$

the 6×6 matrices, $\dot{\tilde{\mathbf{M}}}_i$ and $\tilde{\mathbf{H}}_{ij}$, for $i = 1, \ldots, n$; $j = 1, \ldots, i - 1$, being obtained as

$$\dot{\tilde{\mathbf{M}}}_i = \dot{\mathbf{M}}_i + \dot{\tilde{\mathbf{M}}}_{i+1} \quad \text{and} \quad \tilde{\mathbf{H}}_{ij} \equiv \tilde{\mathbf{M}}_i\dot{\mathbf{B}}_{ij} + \mathbf{B}_{i+1,i}^T\tilde{\mathbf{H}}_{i+1,i} \tag{28}$$

in which $\tilde{\mathbf{M}}_{n+1} = \tilde{\mathbf{H}}_{n+1,n} = \mathbf{O}$. The individual element of the $n \times n$ matrix, $\mathbf{C}$, i.e., c_{ij}, for $i, j = 1, \ldots, n$ is given as

$$c_{ij} = \begin{cases} \mathbf{p}_i^T(\mathbf{B}_{ji}^T\tilde{\mathbf{M}}_j\mathbf{\Omega}_j + \mathbf{B}_{j+1,i}^T\tilde{\mathbf{H}}_{j+1,j} + \dot{\tilde{\mathbf{M}}}_j)\mathbf{p}_j & \text{if} \quad i \le j \\ \mathbf{p}_i^T(\tilde{\mathbf{M}}_i\mathbf{B}_{ij}\mathbf{\Omega}_j + \tilde{\mathbf{H}}_{ij} + \dot{\tilde{\mathbf{M}}}_i)\mathbf{p}_j & \text{otherwise.} \end{cases} \tag{29}$$

4.3 Derivation of $\boldsymbol{\tau}$ Vector. The generalized vector of forces and torques, $\boldsymbol{\tau}$, can be obtained from its expression after Eq. (20), i.e., for, $i = 1, \ldots, n$,

$$\tau_i = \mathbf{p}_i^T\tilde{\mathbf{w}}_i^E, \quad \text{where} \quad \tilde{\mathbf{w}}_i^E = \mathbf{w}_i^E + \mathbf{B}_{i+1,i}^T\tilde{\mathbf{w}}_{i+1}^E \tag{30}$$

in which $\tilde{\mathbf{w}}_i^E$ is the six-dimensional vector component of the $6n$-dimensional vector, $\tilde{\mathbf{w}}^E$, namely,

$$\tilde{\mathbf{w}}^E \equiv [(\tilde{\mathbf{w}}_1^E)^T, \ldots, (\tilde{\mathbf{w}}_n^E)^T]^T. \tag{31}$$

Note in Eq. (30) that in the absence of any external forces and moments on the end-effector, $\tilde{\mathbf{w}}_{n+1}^E = \mathbf{0}$.

5 Applications

A class of serial multibody systems are industrial manipulators. Suitable dynamic models of these manipulators are required mainly for their control and simulation purposes. Control involves *inverse dynamics*, whereas simulation requires *forward dynamics*. These algorithms are presented in the following subsections.

5.1 Inverse Dynamics Algorithm. The inverse dynamics problem is defined as, *given the motion of a manipulator, calculate the actuator forces/torques required to maintain the given motion.* Based on the dynamic equations of motion, Eq. (20) and the expressions for the elements of the associated matrices, **I**, **C**, and $\boldsymbol{\tau}$, Eqs. (24), (29), and (30), respectively, a recursive inverse dynamics algorithm is presented here for the following input: For $i = 1, \ldots, n$,

1 constant Denavit-Hartenberg (DH) parameters (Denavit and Hartenberg, 1955)—the DH parameters are also defined in Appendix A—of the system under study, i.e., a_i, b_i, and α_i, for a revolute pair, and a_i, b_i, and θ_i, for a prismatic pair.

2 time history of the *variable* DH parameter, i.e., θ_i, for a revolute pair, and b_i, for a prismatic joint, and their first and second time derivatives, i.e., $\dot{\theta}_i$, $\ddot{\theta}_i$, and $\dot{b}_i$, $\ddot{b}_i$.

3 mass of each body, m_i.

4 vector denoting the distance of the $(i + 1)$st joint from the ith mass center, C_i, in the $(i + 1)$st frame.

5 inertia tensor of the ith link about its mass center, C_i, in the $(i + 1)$st frame.

The inverse dynamics algorithm then calculates the joint forces/torques, i.e., τ_i, for $i = 1, \ldots, n$, in two recursive steps, as shown below:

Forward Recursion

$$\begin{aligned} &\boldsymbol{\nu}_1 = \mathbf{p}_1\dot{\theta}_1; && \boldsymbol{\xi}_1 = \mathbf{p}_1\ddot{\theta}_1 + \mathbf{\Omega}_1\mathbf{p}_1\dot{\theta}_1 \\ &\boldsymbol{\nu}_2 = \mathbf{p}_2\dot{\theta}_2 + \boldsymbol{\nu}_1; && \boldsymbol{\xi}_2 = \mathbf{p}_2\ddot{\theta}_2 + \mathbf{\Omega}_2\mathbf{p}_2\dot{\theta}_2 + \mathbf{B}_{21}\boldsymbol{\xi}_1 + \dot{\mathbf{B}}_{21}\boldsymbol{\nu}_1 \\ &\vdots && \vdots \\ &\boldsymbol{\nu}_n = \mathbf{p}_n\dot{\theta}_n + \boldsymbol{\nu}_{n-1}; && \boldsymbol{\xi}_n = \mathbf{p}_n\ddot{\theta}_n + \mathbf{\Omega}_n\mathbf{p}_n\dot{\theta}_n + \mathbf{B}_{n,n-1}\boldsymbol{\xi}_{n-1} \\ & && \qquad + \dot{\mathbf{B}}_{n,n-1}\boldsymbol{\nu}_{n-1} \end{aligned}$$

Backward Recursion

$$\begin{aligned} \gamma_n &= \mathbf{M}_n\boldsymbol{\xi}_n + \dot{\mathbf{M}}_n\boldsymbol{\nu}_n; & \tau_n &= \mathbf{p}_n^T\boldsymbol{\gamma}_n \\ \gamma_{n-1} &= \mathbf{M}_{n-1}\boldsymbol{\xi}_{n-1} + \dot{\mathbf{M}}_{n-1}\boldsymbol{\nu}_{n-1} + \mathbf{B}_{n,n-1}^T\boldsymbol{\gamma}_n; & \tau_{n-1} &= \mathbf{p}_{n-1}^T\boldsymbol{\gamma}_{n-1} \\ &\vdots & &\vdots \\ \gamma_1 &= \mathbf{M}_1\boldsymbol{\xi}_1 + \dot{\mathbf{M}}_1\boldsymbol{\nu}_1 + \mathbf{B}_{21}^T\boldsymbol{\gamma}_2; & \tau_1 &= \mathbf{p}_1^T\boldsymbol{\gamma}_1 \end{aligned}$$

where $\boldsymbol{\nu}_i$, $\boldsymbol{\xi}_i$, and $\boldsymbol{\gamma}_i$ are the six-dimensional vectors. Note from the expression for τ_i above and that in Eq. (30), $\tilde{\mathbf{w}}_i^E \equiv \gamma_i$, while $\tilde{\mathbf{w}}_{n+1}^E = \mathbf{0}$. This implies that, without any external forces/moments on the end-effector, only inertia forces, γ_i, contribute to $\mathbf{w}_i^E$, and, hence, the required joint forces/torques are due to system motion only.

For an all revolute-joint serial manipulator, the computational complexity of the above algorithm is compared in Table 1 with

Table 2 Computational complexities in forward dynamics

Algorithm	M	A	$n = 6$	$n = 12$
Proposed	$201n - 335$	$193n - 361$	$871M$ $797A$	$2077M$ $1955A$
Featherstone (1983)	$199n - 198$	$174n - 173$	$996M$ $871A$	$2190M$ $1915A$
Walker and Orin (1982)	$\frac{1}{6}n^3 + 11\frac{1}{2}n^2 + 38\frac{1}{3}n - 47$	$\frac{1}{6}n^3 + 7n^2 + 38\frac{5}{6}n - 46$	$633M$ $480A$	$2357M$ $1716A$

M: Multiplication/Division; A: Addition/Subtraction

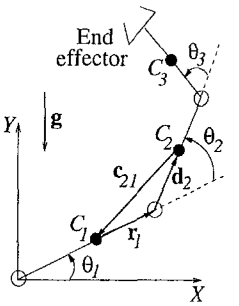

Fig. 4 A three-degrees-of-freedom manipulator

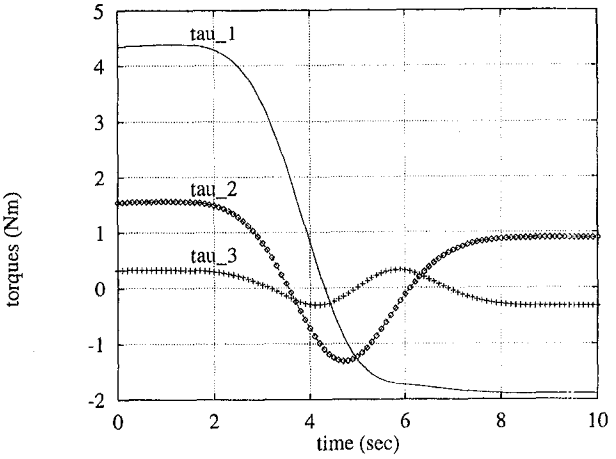

Fig. 5 Torques for three-degrees-of-freedom manipulator

some other existing algorithms. Even though the algorithm is not the best but it is very simple, as evident from the two-step algorithm where six-dimensional vectors are treated similar to those of three dimensional vectors. That is, if one knows $\boldsymbol{\nu}_1$ finding out $\boldsymbol{\nu}_2$ is very similar to the evaluation of the linear velocity of body 2 from the known linear velocity of body 1. A computer software called RIDIM (Recursive Inverse Dynamics of Industrial Manipulators) is developed in C++, which has the following features:

- RIDIM can handle manipulators with both revolute and prismatic pairs or joints;
- gravity is taken into account by providing negative acceleration due to the gravity, denoted by **g**, to the first body, #1 (Kane and Levinson, 1983), i.e., an additional term is added to $\boldsymbol{\xi}_1$ as

$$\boldsymbol{\xi}_1 = \mathbf{p}_1\ddot{\theta}_1 + \boldsymbol{\Omega}_1\mathbf{p}_1\dot{\theta}_1 + \boldsymbol{\rho} \quad \text{where} \quad \boldsymbol{\rho} \equiv [\mathbf{0}^T, -\mathbf{g}^T]^T. \qquad (32)$$

The rest of the algorithm remains same.

The software is illustrated with two examples, as in Section 6.

5.2 Forward Dynamics Algorithm. Forward dynamics is defined as, *given the joint forces/torques, find the joint accelerations.* This step is required in simulation, where the joint accelerations, $\ddot{\boldsymbol{\theta}}$, are solved from the dynamic equations of motion, Eq. (20), and integrated numerically to obtain the joint velocities and positions, $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$, respectively, for a given set of initial conditions.

The required input for the forward dynamics algorithm are as follows: For $i = 1, \ldots, n$,

- inverse dynamics items, 1, and 3–5, as in Section 5.1.
- initial value for the *variable* DH parameter, i.e., θ_i, for a revolute pair, and b_i, for a prismatic pair, and their first time derivative, i.e., $\dot{\theta}_i$ and $\dot{b}_i$.
- .time history of the input joint forces/torques, i.e., τ_i.
- each component of the six-dimensional vector, $\boldsymbol{\phi} \equiv \boldsymbol{\tau} - \mathbf{C}\dot{\boldsymbol{\theta}}$, i.e., ϕ_i, which is to be computed from any recursive inverse dynamics scheme, e.g., the one given in Section 5.1, while $\ddot{\boldsymbol{\theta}} = \mathbf{0}$ (Walker and Orin, 1982).

The forward dynamics algorithm, as proposed in Saha (1997), is presented here for the sake of completeness of this paper. The algorithm is based on the $\mathbf{UDU}^T$ decomposition of the generalized inertia matrix, $\mathbf{I}$ of Eq. (20), i.e., $\mathbf{I} \equiv \mathbf{UDU}^T$, as outlined in Appendix B. Substituting for $\mathbf{I}$, and, $\boldsymbol{\phi} \equiv \boldsymbol{\tau} - \mathbf{C}\dot{\boldsymbol{\theta}}$, in Eq. (20), one obtains

$$\mathbf{UDU}^T\ddot{\boldsymbol{\theta}} = \boldsymbol{\phi}. \qquad (33)$$

A three-step recursive scheme is then used to calculate the joint accelerations from the above equations, i.e.,

1 Solution for $\hat{\boldsymbol{\tau}}$: The solution, $\hat{\boldsymbol{\tau}} = \mathbf{U}^{-1}\boldsymbol{\phi}$, is evaluated as

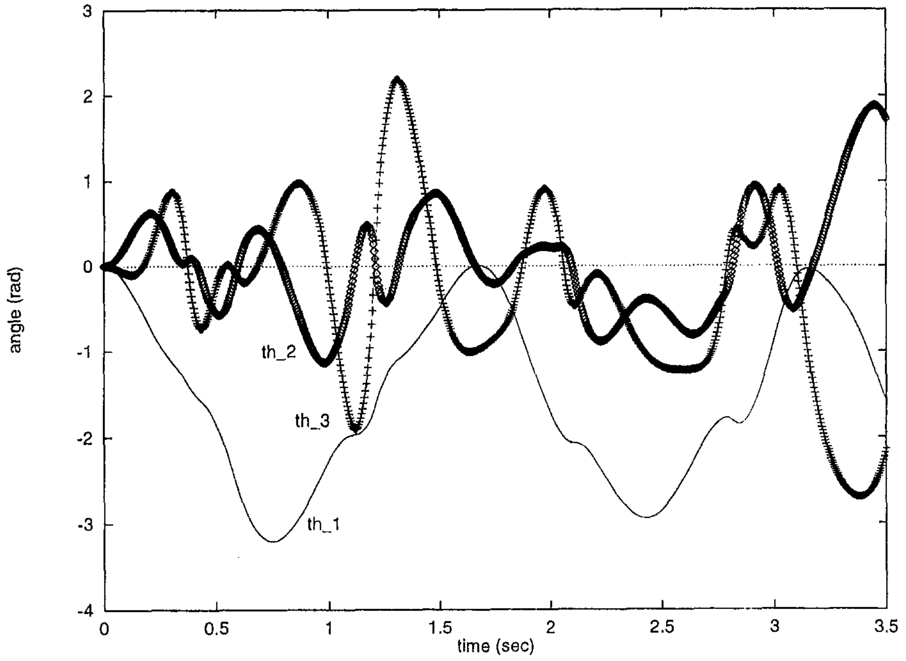

Fig. 6 Simulated joint angles of the planar manipulator

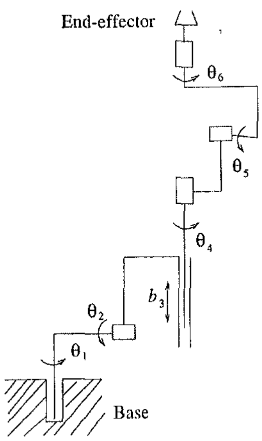

Fig. 7 The six-degrees-of-freedom Stanford arm

$$\hat{\tau}_i = \phi_i - \mathbf{p}_i^T \boldsymbol{\eta}_{i,i+1} \tag{34}$$

where $\hat{\boldsymbol{\tau}}_n \equiv \phi_n$, and the six-dimensional vector, $\boldsymbol{\eta}_{i,i+1}$, is obtained as

$$\boldsymbol{\eta}_{i,i+1} \equiv \mathbf{B}_{i+1,i}^T \boldsymbol{\eta}_{i+1} \quad \text{and} \quad \boldsymbol{\eta}_{i+1} \equiv \boldsymbol{\psi}_{i+1} \hat{\tau}_{i-1} + \boldsymbol{\eta}_{i+1,i+2} \tag{35}$$

in which $\boldsymbol{\eta}_{n,n+1} = \mathbf{0}$. The new variable, $\boldsymbol{\psi}_{i+1}$, is the six-dimensional vector which is defined in Appendix B.

2 Solution for $\tilde{\boldsymbol{\tau}}$: The solution of the equation, $\mathbf{D}\tilde{\boldsymbol{\tau}} = \hat{\boldsymbol{\tau}}$, involves the inverse of the diagonal matrix, **D** of Eq. (54), which is simple, namely, $\mathbf{D}^{-1}$ has only nonzero diagonal elements that are the reciprocal of the corresponding diagonal elements of **D**. Thus, vector $\tilde{\boldsymbol{\tau}}$ is obtained as follows: For $i = 1, \ldots, n$,

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

The scalar, $\hat{m}_i$, is defined in Eq. (47).

3 Solution for $\ddot{\boldsymbol{\theta}}$: In this step, $\ddot{\boldsymbol{\theta}} = \mathbf{U}^{-T}\tilde{\boldsymbol{\tau}}$, is calculated, for $i = 2, \ldots, n$

$$\ddot{\theta}_i = \tilde{\tau}_i - \boldsymbol{\psi}_i^T \boldsymbol{\mu}_{i,i-1} \tag{36}$$

where $\ddot{\theta}_1 \equiv \tilde{\tau}_1$, and the six-dimensional vector, $\boldsymbol{\mu}_{i,i-1}$, is obtained from

$$\boldsymbol{\mu}_{i,i-1} \equiv \mathbf{B}_{i,i-1} \boldsymbol{\mu}_{i-1} \quad \text{and} \quad \boldsymbol{\mu}_{i-1} \equiv \mathbf{p}_{i-1} \ddot{\theta}_{i-1} + \boldsymbol{\mu}_{i-1,i-2} \tag{37}$$

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

The complexity of the proposed forward dynamics algorithm is compared in Table 2 with some other algorithms. The complexity of the proposed algorithm is very similar to those of Featherstone (1983).

Based on the above algorithm, a simulation software, RFDSIM (Recursive Forward Dynamics and Simulation of Industrial Manipulators) is also developed in C++ in which the numerical integration scheme is based on the Runge-Kutta fifth/sixth-order method (Press et al., 1997).

6 Examples

For the illustration of the inverse and forward dynamics algorithms developed in Sections 5.1 and 5.2, respectively, two manipulators, namely, a three-degrees-of-freedom planar manipulator and a six-degrees-of-freedom Stanford arm, as shown in Figs. 4 and 7, respectively, are analyzed using RIDIM and RFDSIM software.

6.1 A Three-Degrees-of-Freedom Planar Manipulator. The manipulator under study is assumed to move in the X–Y plane, whereas the gravity is working in the negative Y direction, as indicated in Fig. 4. Let **i** and **j** be the two unit vectors parallel to the axes, X and Y, respectively, and $\mathbf{k} \equiv \mathbf{i} \times \mathbf{j}$. The expressions for the three joint torques, namely, τ_1, τ_2, and τ_3, are evaluated explicitly from the inverse dynamics algorithm given in Section 5.1, which exactly match with those reported in Angeles (1997). Moreover, RIDIM is used to find the joint torques for the following normal joint angle trajectories and their corresponding 1st and 2nd time derivatives:

$$\theta_i = \frac{1}{2}\left[\frac{2\pi}{T}t - \sin\left(\frac{2\pi}{T}t\right)\right] \tag{38}$$

with $T = 10.0$ sec. The joint torques obtained from RIDIM are plotted in Fig. 5, where "tau_1" "tau_2" and "tau_3" denote the joint torques, τ_1, τ_2, and τ_3, respectively. The plots are also verified with the plots obtained from the explicit expressions.

In order to test RFDSIM, free fall simulation, i.e., motion due to gravity only, without any external joint torque, of the manipulator is carried out with the initial conditions as $\theta_i(0) = \dot{\theta}_i(0) = 0$, for $i = 1, 2, 3$. The variations of the simulated joint angles are shown in Figs. 6, where "th_1," "th_2," and "th_3" represent θ_1, θ_2, and θ_3, respectively. Note from Fig. 4 that due to gravity the first joint angle, θ_1, will increase initially in the negative direction, as evident from Fig. 6. Moreover, the system under gravity behaves as a triple pendulum. These is evident from all the joint angle variations, namely, θ_1, θ_2, and θ_3 of Fig. 6.

6.2 A Stanford Arm. As a second example, a more general six-axis six-degrees-of-freedom manipulator, namely, the Stanford arm, as shown in Fig. 7, is taken whose Denavit-Hartenberg (DH) parameters, along with the mass and inertia properties are given in Table 3. The normal joint trajectory for the six joints are taken from Cyril (1988) as

$$\theta_i = \frac{1}{2}\left[\frac{2\pi}{T}t - \sin\left(\frac{2\pi}{T}t\right)\right] \quad \text{for} \quad i \neq 3 \tag{39a}$$

$$b_3 = \frac{1}{2}\left[\frac{2\pi}{T}t - \sin\left(\frac{2\pi}{T}t\right)\right] \tag{39b}$$

with $T = 10.0$ sec, $\theta_i(0) = 0$, for $i \neq 2, 3$, $\theta_2(0) = 90$ deg, $b_3(0) = 0$, and $\theta_i(T) = 60$ deg, for $i \neq 3$ and $b_3(T) = 0.1$ m. The joint torques are calculated from RIDIM and plotted in Figs.

Table 3 DH parameters, and mass and inertia properties of Stanford arm

Link	a_i	b_i	α_i	θ_i	m_i	r_x	r_y	r_z	I_{xx}	I_{yy}	I_{zz}
	(m)	(m)	(deg)	(deg)	(kg)		(m)			(kg-m^2)	
1	0	.2	−90	θ_1 (0)	9	0	−.1	0	.01	.02	.01
2	0	.1	−90	θ_2 (90)	6	0	0	0	.05	.06	.01
3	0	b_3(0)	0	0	4	0	0	0	.4	.4	.01
4	0	.6	90	θ_4 (0)	1	0	−.1	0	.001	.001	.0005
5	0	0	−90	θ_5 (0)	.6	0	0	0	.0005	.0005	.0002
6	0	0	0	θ_6 (0)	.5	0	0	0	.003	.001	.002

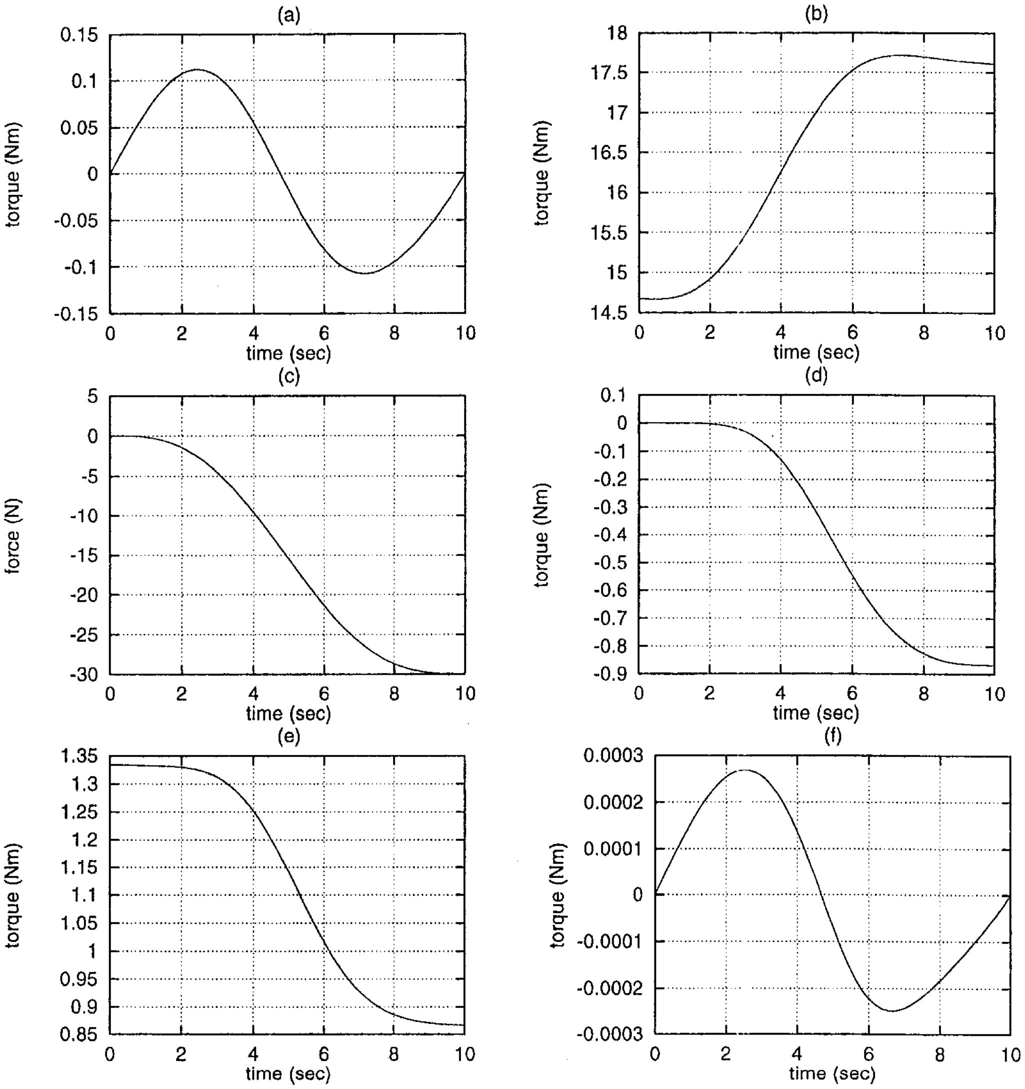

Fig. 8 Torques and force for the six-degrees-of-freedom Stanford arm at (*a*) Joint 1; (*b*) Joint 2; (*c*) Joint 3; (*d*) Joint 4; (*e*) Joint 5; (*f*) Joint 6

8(*a*)–(*f*), which exactly match with those reported in Cyril (1988) for the same arm.

Interpretations similar to the planar manipulator, as given in Section 6.1, can be provided for the Stanford arm also. For example, the motion of joint 3 should sharply increase once the joint 2 turns more than 90 deg, which are evident from Figs. 9(*c*) and (*b*), respectively.

7 Conclusions

A dynamic modeling methodology for serial multi-body mechanical systems is presented. The modeling is based on the Newton-Euler equations of motion and the DeNOC matrices associated with the kinematic constraints, namely, Eq. (16). The dynamic modeling using the DeNOC has the following features:

- physical interpretations of many terms, e.g., the mass matrix of a *composite body*, as defined after Eq. (23*b*), and others, as given after Eqs. (14), (17), and (50).
- unlike the original definition of the NOC (Angeles and Lee, 1988), the DeNOC provides recursive analytical expressions for the elements of the associated matrices, i.e., **I**, **C**, and $\boldsymbol{\tau}$, Eqs. (24), (29), and (30), respectively.
- due to the recursive expressions of the elements of matrices, **I**, **C**, and $\boldsymbol{\tau}$, recursive $\mathcal{O}(n)$ algorithms for inverse and forward dynamics are possible for a general n-axis industrial manipulator, as presented in Section 5.1 and 5.2.

The computational complexities for both the inverse and forward dynamics algorithms are computed and compared in Tables 1 and 2, respectively. The algorithms are not the best in terms of the complexities, however, provide simple algorithms, particularly, the inverse dynamics algorithm presented in Section 5.1. They are also developed uniformly from the same set of equations, namely, Eqs. (20), (24), (29), and (30), which is simple to implement. Two software, namely, RIDIM and RFDSIM are developed in C++, which are tested with respect to two manipulators, namely, a three-degrees-of-freedom planar manipulator and a six-degrees-of-freedom Stanford arm. The results from RIDIM are compared with those available in the literature for the same systems, namely, with Angeles (1997) for the three-degrees-of-freedom manipulator and with Cyril (1988) for the six-degrees-of-freedom Stanford arm, whereas the results of RFDSIM are interpreted intuitively.

Acknowledgment

A part of the research work reported in this paper was possible under the Department of Science and Technology, Government of India, Grant No. HR/OY/E-02/96. The author sincerely acknowledges the support.

References

Angeles, J., 1997, *Fundamentals of Robotic Mechanical Systems*, Springer-Verlag, New York.

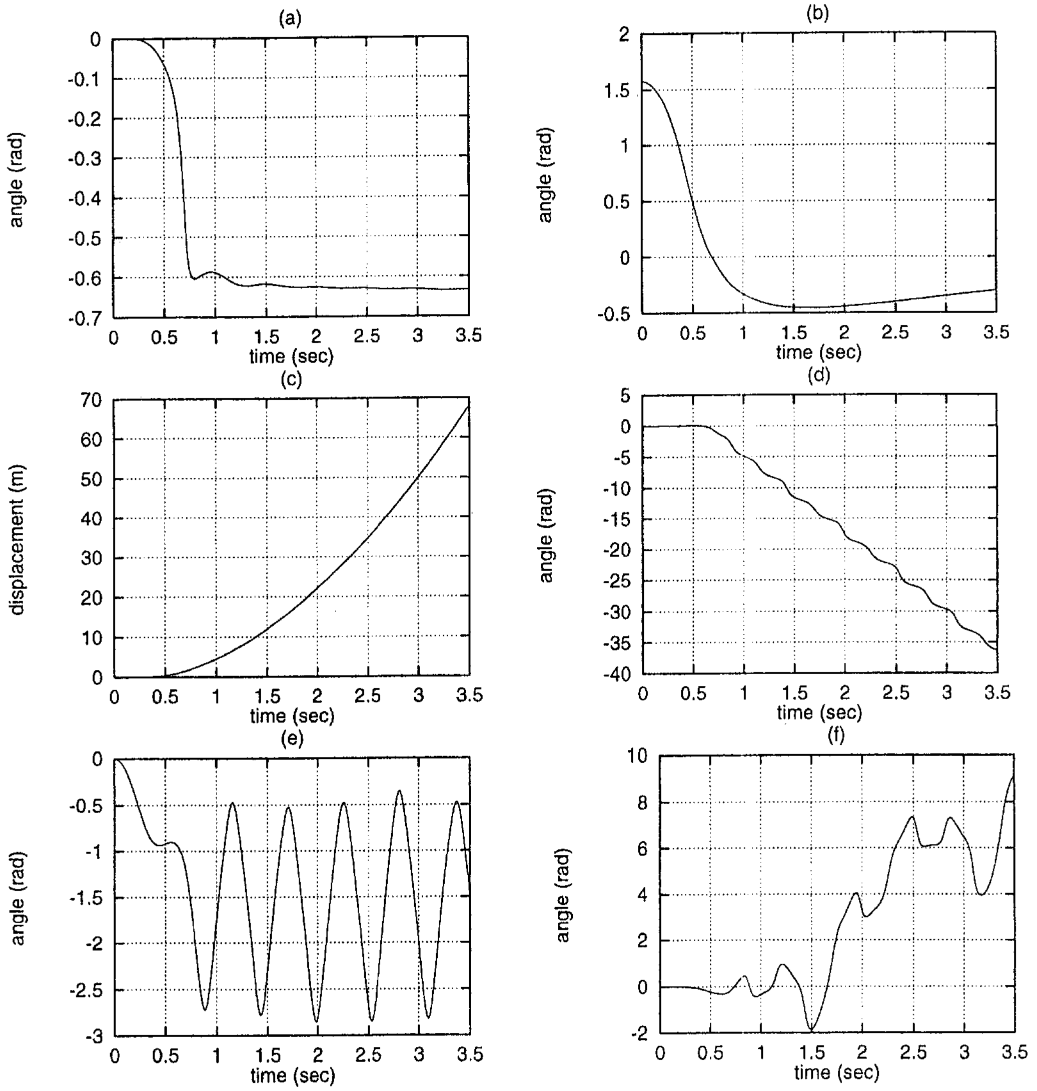

Fig. 9 Simulated joint angles and displacement for the six-degrees-of-freedom Stanford arm at (*a*) Joint 1; (*b*) Joint 2; (*c*) Joint 3; (*d*) Joint 4; (*e*) Joint 5; (*f*) Joint 6

Angeles, J., and Lee, S., 1988, "The Formulation of Dynamical Equations of Holonomic Mechanical Systems Using a Natural Orthogonal Complement," ASME JOURNAL OF APPLIED MECHANICS, Vol. 55, pp. 243–244.

Angeles, J., and Ma, O., 1988, "Dynamic Simulation of n-axis Serial Robotic Manipulators Using a Natural Orthogonal Complement," *Int. J. of Rob. Res.*, Vol. 7, No. 5, pp. 32–47.

Angeles, J., Ma, O., and Rojas, A., 1989, "An Algorithm for the Inverse Dynamics of n-Axis General Manipulator Using Kane's Formulation of Dynamical Equations," *Computers and Mathematics with Applications*, Vol. 17, No. 12, pp. 1545–1561.

Armstrong, W. W., 1979, "Recursive Solution to the Equations of Motion of an n-Link Manipulator," *Proc. of the 5th World Cong. on Th. of Mach. and Mech.*, Vol. 2, Montreal, Canada, ASME, New York, pp. 1343–1346.

Ascher, U. M., Pai, D. K., and Cloutier, B. P., 1997, "Forward Dynamics, Elimination Methods, and Formulation Stiffness in Robot Simulation," *Int. J. of Rob. Res.*, Vol. 16, No. 6, pp. 749–758.

Cyril, X., 1988, "Dynamics of Flexible-Link Manipulators," Ph.D. thesis, McGill University, Canada.

Denavit, J., and Hartenberg, R. S., 1955, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," ASME JOURNAL OF APPLIED MECHANICS, Vol. 77, pp. 215–221.

Featherstone, R., 1983, "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *Int. J. of Rob. Res.*, Vol. 2, No. 1, pp. 13–30.

Fijany, A., Sharf, I., and D'Eleuterio, M. T. D., 1995, "Parallel $\mathcal{O}(\log N)$ Algorithms for Computation of Manipulator Forward Dynamics," *IEEE Trans. on R&A*, Vol. 11, No. 3.

Hollerbach, J. M., 1980, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Trans. on Sys., Man, and Cybernetics*, Vol. SMC-10, pp. 730–736.

Huston, H., and Passerello, C. E., 1974, "On Constraint Equations—A New Approach," ASME JOURNAL OF APPLIED MECHANICS, Vol. 41, pp. 1130–1131.

Jain, A., and Rodriguez, G., 1995, "Diagonalized Lagrangian Robot Dynamics," *IEEE Trans. on R&A*, Vol. 11, No. 4, pp. 571–584.

Kamman, J. W., and Huston, R. L., 1984, "Dynamics of Constrained Multibody Systems," ASME JOURNAL OF APPLIED MECHANICS, Vol. 51, pp. 899–903.

Kane, T. R., and Levinson, D. A., 1983, "The Use of Kane's Dynamical Equations for Robotics," *Int. J. of Rob. Res.*, Vol. 2, No. 3, pp. 3–21.

Lilly, K. W., and Orin, D. E., 1990, "Efficient $\mathcal{O}(n)$ Computation of the Operational Space Inertia Matrix," *Proc. of the IEEE Conf. on R&A*, Vol. 2, Cincinnati, OH, May 13–18, pp. 1014–1019.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., 1980, "On-Line Computational Scheme for Mechanical Manipulators," ASME *Journal of Dynamic Systems, Measurement and Control*, Vol. 102, pp. 69–76.

McMillan, S., and Orin, D. E., 1995, "Efficient Computation of Articulated-Body Inertias Using Successive Axial Screws," *IEEE Trans. on R&A*, Vol. 11, No. 4, pp. 606–611.

Press, W. H., Teukolsky, S. A., Vellerling, W. T., and Flannery, B. P., 1997, *Numerical Recipes in C*, 2nd Ed., Cambridge University Press, New Delhi.

Rodriguez, G., 1987, "Kalman Filtering, Smoothing, and Recursive Robot Arm Forward and Inverse Dynamics," *IEEE Trans. on R&A*, Vol. RA-3, No. 6, pp. 624–639.

Rodriguez, G., Jain, A., and Kreutz-Delgado, K., 1991, "A Spatial Operator Algebra for Manipulation Modeling and Control," *Int. J. of Rob. Res.*, Vol. 10, No. 4, pp. 371–381.

Rodriguez, G., and Kreutz-Delgado, K., 1992, "Spatial Operator Factorization and Inversion of the Manipulator Mass Matrix," *IEEE Trans. on R&A*, Vol. 8, No. 1, pp. 65–76.

Saha, S. K., and Angeles, J., 1991, "Dynamics of Nonholonomic Mechanical Systems Using a Natural Orthogonal Complement," ASME JOURNAL OF APPLIED MECHANICS, Vol. 58, pp. 238–243.

Saha, S. K., 1997, "A Decomposition of the Manipulator Inertia Matrix," *IEEE Trans. on R&A*, Vol. 13, No. 2, pp. 301–304.

Schiehlen, W., 1991, "Computational Aspects in Multibody System Dynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 90, No. 1–3, pp. 569–582.

Sciavicco, L., and Siciliano, B., 1996, *Modeling and Control of Robot Manipulators*, McGraw-Hill, New York.

Stewart, G. E., 1973, *Introduction to Matrix Computations*, Academic Press, San Diego, CA.

Walker, M. W., and Orin, D. E., 1982, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," ASME *Journal of Dynamic Systems, Measurement and Control*, Vol. 104, pp. 205–211.

Wehage, R. A., and Haug, E. J., 1982, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," ASME *Journal of Mechanical Design*, Vol. 104, pp. 247–255.

APPENDIX A

Denavit-Hartenberg Nomenclature

In order to describe the Denavit-Hartenberg (DH) nomenclatures (Denavit and Hartenberg, 1955) that is followed here, note that the manipulator under study consists of $n + 1$ links, namely, a "base" and n links, denoted by #1, . . . , #n, coupled by n pairs numbered as $1, \ldots, n$, respectively. As shown in Fig. 1, ith pair couples the $(i - 1)$st and the ith links. Moreover, a coordinate system X_i, Y_i, Z_i is attached to the $(i - 1)$st link.

Then, for the first n frames, DH parameters are defined according to the following rules: Referring to Fig. 10,

1 Z_i is the axis of the ith pair. Its positive direction can be chosen arbitrarily.

2 X_i is defined as the common perpendicular to Z_{i-1} and Z_i, directed from the former to the latter. The origin of the ith frame, O_i, is the point where X_i intersects Z_i. If these two axes intersect, the positive direction of X_i is chosen arbitrary. The origin, O_i, is coincides with the origin of the $(i - 1)$st frame, O_{i-1}.

3 the distance between Z_i and Z_{i+1} is defined as a_i, which is *non-negative.*

4 the Z_i coordinate of the intersection of the X_{i+1} axis with Z_i is defined as b_i, which thus can be either positive or negative. For a prismatic joint, b_i is variable.

5 the angle between Z_i and Z_{i+1} is defined as α_i, and is measured about the positive direction of X_{i+1}, and

6 the angle between X_i and X_{i+1} is defined as θ_i, and is measured about the positive direction of Z_i. For a revolute joint, θ_i is variable.

Since no $(n + 1)$st link exists the above definitions do not apply to the $(n + 1)$st frame and it can be chosen at will.

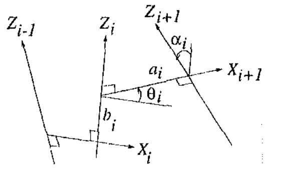

Fig. 10 Denavit-Hartenberg nomenclature

APPENDIX B

Decomposition of the GIM, I

In order to obtain the $\mathbf{UDU}^T$ decomposition of the GIM, $\mathbf{I}$ of Eq. (20)—the elements of which are given by Eq. (24)—the rules of the Gaussian elimination (GE) (Stewart, 1973) are applied on these elements for the Reverse Gaussian Elimination (RGE), as proposed in Saha (1997). The scheme is reproduced here for the completeness of the paper.

B.1 RGE of the GIM, I. Conventionally, the GE (Stewart, 1973) begins from the first column of the matrix under interest. In the proposed elimination, it is assumed that the GE of matrix $\mathbf{I}$, Eq. (20), whose elements are given by Eq. (24), starts from the nth column. This is to obtain recursive relations starting from the nth body. Thus, the RGE name is used. In RGE, after the annihilation of the first $(n - 1)$ elements of the nth column, the modified inertia matrix, denoted by $\mathbf{L}_n$, is expressed as

$$\mathbf{L}_n \equiv \begin{bmatrix} i^{(n)}_{11} & & \text{sym} & 0 \\ \vdots & \ddots & & 0 \\ i^{(n)}_{n-1,1} & \cdots & i^{(n)}_{n-1,n-1} & 0 \\ i_{n1} & \cdots & i_{n,n-1} & i_{nn} \end{bmatrix} \tag{40}$$

where i_{nn} is the *pivot* (Stewart, 1973) and $i^{(n)}_{ij}$ are the modified elements of $\mathbf{I}$, whereas "sym" denotes the symmetric elements of the $(n - 1) \times (n - 1)$ matrix, resulting from the deletion of the nth row and column of matrix $\mathbf{L}_n$. Equation (40) is realized by premultiplying matrix $\mathbf{I}$ with the *elementary upper triangular matrix* (EUTM) of order n and index n, as done in GE with *elementary lower triangular matrix* (ELTM) (Stewart, 1973). An EUTM of order n and index k, denoted by $\mathbf{E}_k$, is defined as

$$\mathbf{E}_k \equiv \mathbf{1} - \boldsymbol{\alpha}_k \boldsymbol{\lambda}_k^T \tag{41}$$

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

$$\boldsymbol{\alpha}_k \equiv [\alpha_{1k}, \ldots, \alpha_{k-1,k}, 0, \ldots, 0]^T \tag{42}$$

$$\boldsymbol{\lambda}_k \equiv [0, \ldots, 0, 1, \ldots, 0]^T. \tag{43}$$

From Eqs. (42) and (43), the EUTM, $\mathbf{E}_k$, Eq. (41), has the following structure:

$$\mathbf{E}_k \equiv \begin{bmatrix} 1 & 0 & \cdots & -\alpha_{1k} & \cdots & 0 \\ & \ddots & \vdots & \vdots & \vdots & \vdots \\ & & 1 & -\alpha_{k-1,k} & \cdots & 0 \\ & & & 1 & \cdots & 0 \\ & & 0's & & \ddots & \vdots \\ & & & & & 1 \end{bmatrix} \tag{44}$$

where "$0's$" imply zeros. Moreover, in the RGE, the modified matrix after the annihilation of the first k–1 elements of the kth column is expressed as

$$\mathbf{L}_k = \mathbf{E}_k \mathbf{L}_{k+1} \tag{45}$$

Note here that, if $k = n$ and $\mathbf{L}_{n+1} \equiv \mathbf{I}$, and the matrix, $\mathbf{L}_n$ of Eq. (40), follows from Eq. (45). Furthermore, the elements of $\mathbf{E}_k$ and $\mathbf{L}_k$, α_{ik} and $i^{(k)}_{ij}$, respectively, are computed from the following scheme:

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

$$\alpha_{ik} = \mathbf{p}_i^T \boldsymbol{\psi}_{ik} \quad \text{and} \quad i^{(k)}_{ij} = \mathbf{p}_i^T \hat{\mathbf{M}}_{ik} \mathbf{B}_{ij} \mathbf{p}_j \tag{46}$$

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

In Eq. (46), matrix $\mathbf{B}_{ij}$, and vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ are defined in Eq. (13), whereas the six-dimensional vector $\boldsymbol{\psi}_{ik}$ and the terms associated to it are given as

$$\hat{\boldsymbol{\psi}}_k \equiv \hat{\mathbf{M}}_k \mathbf{p}_k, \quad \hat{\boldsymbol{\psi}}_{ik} \equiv \mathbf{B}_{ki}^T \hat{\boldsymbol{\psi}}_k, \quad \hat{m}_k \equiv \mathbf{p}_k^T \hat{\boldsymbol{\psi}}_k, \tag{47}$$

$$\boldsymbol{\psi}_k \equiv \frac{\hat{\boldsymbol{\psi}}_k}{\hat{m}_k}, \quad \text{and} \quad \boldsymbol{\psi}_{ik} \equiv \frac{\hat{\boldsymbol{\psi}}_{ik}}{\hat{m}_k}. \tag{48}$$

In Eq. (47), $\hat{\mathbf{M}}_k \equiv \hat{\mathbf{M}}_{k,k+1}$, which is used to simplify the notation. The 6×6 matrix, $\hat{\mathbf{M}}_{ik}$ of Eq. (46) and $\hat{\mathbf{M}}_k$ ($\equiv \hat{\mathbf{M}}_{k,k+1}$) of Eq. (47), are evaluated from the following relation:

$$\hat{\mathbf{M}}_{ik} \equiv \tilde{\mathbf{M}}_i - \boldsymbol{\Psi}_{ik}, \quad \text{where} \quad \boldsymbol{\Psi}_{ik} \equiv \sum_{l=k}^{n} \hat{\boldsymbol{\psi}}_{il} \boldsymbol{\psi}_{il}^T \tag{49}$$

and $\tilde{\mathbf{M}}_i$ is given in Eq. (23*a*). Note that, the 6×6 matrix, $\boldsymbol{\Psi}_{ik}$, has a recursive relation, whose substitution, along with that of $\tilde{\mathbf{M}}_i$, Eq.

(23*b*), into the expression of $\hat{\mathbf{M}}_{ik}$, Eq. (49), leads to a recursive equation, i.e.,

$$\hat{\mathbf{M}}_{ik} = \mathbf{M}_i + \mathbf{B}^T_{i+1,i}\hat{\mathbf{M}}_{i+1,k}\mathbf{B}_{i+1,i} \quad \text{and} \quad \hat{\mathbf{M}}_{kk} = \hat{\mathbf{M}}_k - \hat{\boldsymbol{\psi}}_k\boldsymbol{\psi}^T_k \tag{50}$$

where $k = n, \ldots, 2$; $i = k - 1, \ldots, 1$. Moreover, for $i = k - 1$, $\hat{\mathbf{M}}_{i+1,k} \equiv \hat{\mathbf{M}}_{kk}$, and $\hat{\mathbf{M}}_n \equiv \mathbf{M}_n$.

Note that, in contrast to the definition of $\tilde{\mathbf{M}}_i$ associated to the *composite body, i*, Eq. (23*a*), the matrix, $\hat{\mathbf{M}}_i$, as in Eq. (50) for $k = i$, has the following interpretation: It represents the mass and inertia properties of the *articulated body i,* which is defined as the system consisting of bodies #*i*, . . . , #*n* that are coupled by joints $i + 1, \ldots, n$, i.e., replace the fixed joints of the *i*th composite body, as shown in Fig. 1 by the dotted line, by a kinematic pair, say, a revolute joint. Here, the matrix, $\hat{\mathbf{M}}_i$, incorporates the effects of joints into the rigid *composite system i*, resulting $\tilde{\mathbf{M}}_i$ of Eq. (23*a*). In fact, comparison of the expressions for $\tilde{\mathbf{M}}_i$ and $\hat{\mathbf{M}}_i$ ($\equiv\hat{\mathbf{M}}_{i,i+1}$), as in Eqs. (23*b*) and can be obtained from (49), respectively, show that the 6 × 6 matrix, $\boldsymbol{\Psi}_{i,i+1}$, causes this change. This could have prompted Lilly and Orin (1990) to name the matrix as the *articulation transformation* matrix. Matrix, $\hat{\mathbf{M}}_i$ is also termed as the *articulated-body inertia of link i* (Featherstone, 1983), and the *state estimation error covariance* (Rodriguez, 1987; Rodriguez and Kreutz-Delgado, 1992) that satisfies the discrete Riccati equations. Thus, it is commented that a correlation exists between the Gaussian elimination technique and the Kalman filtering approach, which can be exploited for the deeper understanding of the dynamics characteristics of a complex mechanical system. Finally, the scalar, $\hat{m}_i$, is interpreted as the moment of inertia of the *articulated body, i,* about the axis of rotation of the *i*th joint.

B.2 $\mathbf{UDU}^T$ Decomposition. The scheme to obtain the $\mathbf{UDU}^T$ are then given in the following steps:

1 The RGE of **I**, whose elements are given by Eq. (24), can be expressed as

$$\mathbf{EI} = \mathbf{L}_2, \quad \text{where} \quad \mathbf{E} \equiv \mathbf{E}_2 \ldots \mathbf{E}_n \tag{51}$$

in which the $n \times n$ matrix, $\mathbf{E}_k$, for $k = n, \ldots, 2$, is the EUTM and **E** and $\mathbf{L}_2$, respectively, are the $n \times n$ upper and lower triangular matrices.

2 An essential property of the EUTM, $\mathbf{E}_K$, similar to the ELTM (Stewart, 1973), is

$$\mathbf{E}^{-1}_k \equiv (\mathbf{1} - \boldsymbol{\alpha}_k\boldsymbol{\lambda}^T_k)^{-1} = \mathbf{1} + \boldsymbol{\alpha}_k\boldsymbol{\lambda}^T_k \tag{52}$$

where $\boldsymbol{\alpha}_k$ and $\boldsymbol{\lambda}_k$ are defined in Eqs. (42) and (43), respectively. Using Eq. (52), the GIM, **I**, is written from Eq. (51) as

$$\mathbf{I} = \mathbf{UL}_2, \quad \text{where} \quad \mathbf{U} \equiv \mathbf{E}^{-1}. \tag{53}$$

In Eq. (53), **U** and $\mathbf{L}_2$ are the $n \times n$ upper and lower triangular matrices, respectively. Moreover, from the inverse of the EUTM, Eq. (52), it is clear that the diagonal elements of **U** are unity and the above-diagonal elements are the components of the vector, $\boldsymbol{\alpha}_k$, for $k = 2, \ldots, n$, that are evaluated in Eq. (46).

3 Since the factorization, given by Eq. (53) is not unique (Stewart, 1973), a unique decomposition is obtained by normalizing the elements of $\mathbf{L}_2$ as

$$\mathbf{L}_2 = \mathbf{DL}, \quad \text{where} \quad \mathbf{D} \equiv \text{diag}\,[\hat{m}_1, \ldots, \hat{m}_n] \tag{54}$$

D being the $n \times n$ diagonal matrix, whose non zero elements are those of the matrix, $\mathbf{L}_2$, as calculated in Eq. (47). Hence, the diagonal elements of matrix **L** are unity.

4 Since, for the SPD matrix, **I**, $\mathbf{L} \equiv \mathbf{U}^T$ (Stewart, 1973), the desired decomposition of the manipulator GIM, **I** of Eq. (20), is given as

$$\mathbf{I} = \mathbf{UDU}^T \tag{55}$$

where the elements of the matrices, **U** and **D**, are evaluated using Eqs. (46)–(50).