International Conference on Multi Body Dynamics 2011
Vijayawada, India. pp. 351–365

Dynamics Algorithms for Multibody Systems

S. V. Shah, S. K. Saha and J. K. Dutt

Department of Mechanical Engineering, Indian Institute of Technology Delhi New Delhi 110 016, India
surilvshah@gmail.com; saha@mech.iitd.ac.in; jkdutt@mech.iitd.ac.in

Abstract. This paper presents dynamics algorithms for multibody systems using the concept of Decoupled Natural Orthogonal Complement (DeNOC) matrices. For this, a multibody system is treated here as an open-loop serial-or tree-types. In fact, it is shown how the knowledge for a serial system can be extended to a tree-type system. The resulting equations are used to obtain recursive inverse and forward dynamics algorithms. The methodology is illustrated using a tree-type four Degree-Of-Freedom (DOF) robotic gripper and 100-DOF serial-type rope system. It will be shown how efficiency of the proposed algorithm will benefit as the DOF of a system becomes very large.

Keywords.

1. Introduction

Studies on any multibody system is attempted to perform either force or motion analyses. Kinematic relationships between constituent links or bodies of a multibody system are foundation for aforesaid analyses. With increase in multibody applications, the systems to be studied have become more complex and contain large number of bodies. Moreover, many multibody systems comprise of smaller subsystems. Hence, simply breaking down a system to body level does not provide macroscopic purview of several kinematic and dynamics properties, thereby, increase the complexity of analyses in many instances.

In this work, dynamics is proposed using the concept of Decoupled Natural Orthogonal Complement (DeNOC) matrices, which when combined with the uncoupled Newton-Euler equations of motion gives rise to the minimal-order constrained dynamic equations of motion of the system at hand by eliminating the constraint forces. This decoupled form of the DeNOC matrices brings the benefits like analytical recursive expressions for the elements of the vectors and matrices appearing in the dynamic equations of motion, $O(n)$ recursive dynamics, etc., as reported for the serial-type systems by Saha (1999a).

Several other approaches were also proposed in the literature by Featherstone (1987), Bae and Haug (1988), Rodriguez *et al.* (1992), and Rosenthal (1991) to

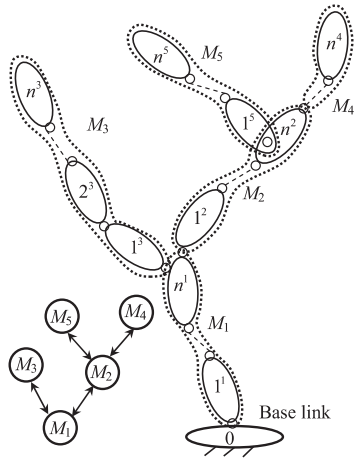

Figure 1. A tree-type system and its modules.

obtained recursive dynamics algorithms. In comparison to those approaches the DeNOC based approach considered the system as a whole, i.e., NE equations of motion of all the bodies in a system are grouped together. Next, simple matrix operations using the DeNOC matrices, derived from the velocity constraints of serially connected bodies or kinematic modules, are performed to draw up minimal-order equations of motion. As a consequence, a global approach is possible in contrast to the body-centric approach by Featherstone (1983). Moreover, the use of DeNOC matrices allows one to uniformly develop both recursive inverse and forward dynamics algorithms, and analytical inversion of the Generalized Inertia Matrix (Saha, 1999b). The analytical inversion was not possible using any of the other recursive $O(n)$ dynamic formulations.

In the proposed approach, a multiple system is considered to be an open-loop serial- or tree-type. For a tree-type system, it is considered to have a set of many kinematic modules as shown in figure 1, where each module contains serially connected rigid links and emerges from the last link of the parent module. The links in the module M_i are denoted as $1^i, \ldots, k^i, \ldots, n^i$, where superscript i is the module number. Total number of modules, number of links in the module, and the total number of links in all the modules, are designated using letters, s, n^i, and n, respectively. It is interesting to note here that a tree-type system with no branch degenerate to a serial-type system. Hence, the methodology to be presented within the paper will be with respect to a tree-type system only.

The paper is organized as follows: Section 2 presents dynamics modeling. Recursive algorithms are presented in section 3. Computational efficiency is presented in section 4. Finally, conclusions are given in section 5.

2. Dynamic Modeling

The DeNOC-based methodology for dynamic modeling begins with the Newton-Euler (NE) equations of motion. For the i^{th} module, say M_i of

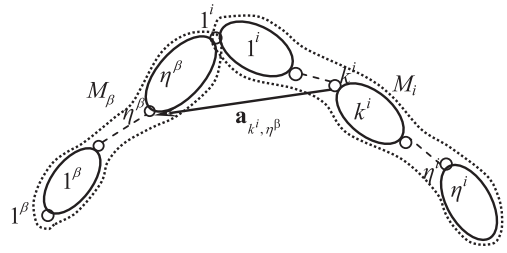

Figure 2. Module M_i and its parent M_β.

figure 2, the NE equations of motion may be expressed in a compact form as

$$\bar{\boldsymbol{M}}_i\dot{\bar{\boldsymbol{t}}}_i + \bar{\boldsymbol{\Omega}}_i\bar{\boldsymbol{M}}_i\bar{\boldsymbol{E}}_i\bar{\boldsymbol{t}}_i = \bar{\boldsymbol{w}}_i \tag{1}$$

where the matrices $\bar{\boldsymbol{M}}_i, \bar{\boldsymbol{\Omega}}_i$, and $\bar{\boldsymbol{E}}_i$, are of sizes $6n^i \times 6n^i$ each, and $\bar{\boldsymbol{w}}_i$ is the $6n^i$-dimensional vector of module-wrench associated with module M_i. Matrices, $\bar{\boldsymbol{M}}_i, \bar{\boldsymbol{\Omega}}_i, \bar{\boldsymbol{E}}_i$, and vector $\bar{\boldsymbol{w}}_i$ are defined as

$$\begin{aligned}
\bar{\boldsymbol{M}}_i &\equiv \text{diag}\,[\boldsymbol{M}_1 \ \cdots \ \boldsymbol{M}_k \ \cdots \ \boldsymbol{M}_n]^i, \\
\bar{\boldsymbol{\Omega}}_i &\equiv \text{diag}\,[\boldsymbol{\Omega}_1 \ \cdots \ \boldsymbol{\Omega}_k \ \cdots \ \boldsymbol{\Omega}_n]^i, \qquad \text{and} \quad \bar{\boldsymbol{w}}_i \equiv \begin{bmatrix} \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w_k} \\ \vdots \\ \boldsymbol{w}_n \end{bmatrix}^i \\
\bar{\boldsymbol{E}}_i &\equiv \text{diag}\,[\boldsymbol{E}_1 \ \cdots \ \boldsymbol{E}_k \ \cdots \ \boldsymbol{E}_n]^i,
\end{aligned} \tag{2}$$

where $\boldsymbol{\Omega}_k$, $\boldsymbol{M}_k$ and $\boldsymbol{E}_k$, are the 6×6 angular velocity, mass and coupling matrices, respectively, and $\boldsymbol{w}_k$ is the 6-dimnesional vector of wrench for the k^{th} link of module M_i as obtained in Shah and Saha (2009). Combining equation (1) for $i = 1, \ldots, s$, the uncoupled NE equations of motion for the s-coupled modules are then put together as

$$\boldsymbol{M}\dot{\boldsymbol{t}} + \boldsymbol{\Omega}\boldsymbol{M}\boldsymbol{E}\boldsymbol{t} = \boldsymbol{w} \tag{3}$$

In equation (3), $\boldsymbol{M}$, $\boldsymbol{\Omega}$, and $\boldsymbol{E}$ are the $6n \times 6n$ generalized matrices of mass, angular velocity, and coupling, respectively, and $\boldsymbol{w}$ is the $6n$-dimensional generalized vector of wrenches, which are given as follows:

$$\begin{aligned}
\boldsymbol{M} &\equiv \text{diag}\,[\bar{\boldsymbol{M}}_1 \ \cdots \ \bar{\boldsymbol{M}}_i \ \cdots \ \bar{\boldsymbol{M}}_s], \\
\boldsymbol{\Omega} &\equiv \text{diag}\,[\bar{\boldsymbol{\Omega}}_1 \ \cdots \ \bar{\boldsymbol{\Omega}}_i \ \cdots \ \bar{\boldsymbol{\Omega}}_s], \qquad \text{and} \quad \boldsymbol{w} \equiv \begin{bmatrix} \bar{\boldsymbol{w}}_1 \\ \vdots \\ \bar{\boldsymbol{w}}_s \end{bmatrix} \\
\boldsymbol{E} &\equiv \text{diag}\,[\bar{\boldsymbol{E}}_1 \ \cdots \ \bar{\boldsymbol{E}}_i \ \cdots \ \bar{\boldsymbol{E}}_s],
\end{aligned} \tag{4}$$

2.1 Derivation of the DeNOC matrices

In order to obtain constrained equations of motion the Decoupled Natural Orthogonal Complement (DeNOC) matrices are derived next in this subsection

using the velocity constraints. As shown in figure 2, n^{β} is the parent link of the module M_i. It may be noted that the symbol 'β' is used to signify a parent. For the i^{th} module M_i, the $6n^i$-dimensional vector module-twist, $\bar{\boldsymbol{t}}_i$, is obtained from the module-twist of its parent module $M_\beta, \bar{\boldsymbol{t}}_\beta$, and the n^i-dimensional joint-rate, $\dot{\bar{\boldsymbol{\theta}}}_i$, i.e.,

$$\bar{\boldsymbol{t}}_i = \bar{\boldsymbol{A}}_{i,\beta}\bar{\boldsymbol{t}}_\beta + \bar{\boldsymbol{N}}_i\dot{\bar{\boldsymbol{\theta}}} \tag{5}$$

where $\bar{\boldsymbol{t}}_i$ and $\dot{\bar{\boldsymbol{\theta}}}_i$ are defined as follows:

$$\bar{\boldsymbol{t}}_i \equiv \begin{bmatrix} \boldsymbol{t}_1 \\ \vdots \\ \boldsymbol{t}_k \\ \vdots \\ \boldsymbol{t}_n \end{bmatrix}^i \quad \text{and} \quad \dot{\bar{\boldsymbol{\theta}}}_i \equiv \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_k \\ \vdots \\ \dot{\theta}_n \end{bmatrix}^i \tag{6}$$

Note that the superscript i outside the bracket represents the i^{th} module. The term $\boldsymbol{t}_k$ is associated with the angular velocity and linear velocity of the origin of the k^{th} link, i.e., $\boldsymbol{t}_k \equiv [\boldsymbol{\omega}_k^T \;\; \dot{\boldsymbol{o}}_k^T]^T$. Moreover, $\bar{\boldsymbol{A}}_{i,\beta}$, and $\bar{\boldsymbol{N}}_i$ are the $6n^i \times 6n^\beta$ module-twist propagation and $6n^i \times n^i$ module-joint-rate propagation matrices, respectively, which are given by

$$\bar{\boldsymbol{A}}_{i,\beta} \equiv \begin{bmatrix} \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{A}_{1^i,n^\beta} \\ \vdots & & \vdots & \vdots \\ \boldsymbol{O} & & \boldsymbol{O} & \boldsymbol{A}_{k^i,n^\beta} \\ \vdots & & \vdots & \vdots \\ \boldsymbol{O} & \cdots & \boldsymbol{O} & \boldsymbol{A}_{n^i,n^\beta} \end{bmatrix}, \quad \text{and}$$

$$\bar{\boldsymbol{N}}_i \equiv \begin{bmatrix} \boldsymbol{p}_1 & & \boldsymbol{O} & & \cdots & \boldsymbol{O} \\ & \ddots & & & & \\ \boldsymbol{A}_{2,1}\boldsymbol{p}_1 & & \boldsymbol{p}_{k-1} & & \ddots & \vdots \\ & \ddots & & & & \\ \vdots & & \boldsymbol{A}_{k,k-1}\boldsymbol{p}_{k-1} & & \ddots & \boldsymbol{O} \\ & & & \ddots & & \\ \boldsymbol{A}_{n,1}\boldsymbol{p}_1 & & \cdots & & \boldsymbol{A}_{n,n-1}\boldsymbol{p}_{n-1} & \boldsymbol{p}_n \end{bmatrix}^i \tag{7}$$

In equation (7) '$\boldsymbol{O}$' and '$\boldsymbol{0}$' denote null matrix and null vector of compatible dimensions, whereas $\bar{\boldsymbol{A}}_{i,\beta}$ is associated with the twist-propagation of the parent module

(β^{th}) to the child module (i^{th}). Its block elements $\boldsymbol{A}_{k,k-1}$ and $\boldsymbol{p}_k$ in $\bar{\boldsymbol{N}}_i$ are the 6×6 twist-propagation matrix and the 6-dimensional joint-motion propagation vector, respectively. The definitions of the generalized twist and generalized joint rate vectors of the multibody system at hand are then introduced as

$$\boldsymbol{t} \equiv \begin{bmatrix} \bar{\boldsymbol{t}}_1 \\ \vdots \\ \bar{\boldsymbol{t}}_i \\ \vdots \\ \bar{\boldsymbol{t}}_s \end{bmatrix} \quad \text{and} \quad \dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\bar{\boldsymbol{\theta}}}_1 \\ \vdots \\ \dot{\bar{\boldsymbol{\theta}}}_i \\ \vdots \\ \dot{\bar{\boldsymbol{\theta}}}_s \end{bmatrix} \tag{8}$$

Now, substitution of equation (5) it into equation (8) yields

$$\boldsymbol{t} = \bar{\boldsymbol{N}}_l \bar{\boldsymbol{N}}_d \dot{\boldsymbol{\theta}} \tag{9}$$

where the $6n \times 6n$ block lower triangular matrix $\bar{\boldsymbol{N}}_l$ and the $6n \times n$ block diagonal matrix $\bar{\boldsymbol{N}}_d$ are given by

$$\bar{\boldsymbol{N}}_l \equiv \begin{bmatrix} \bar{\mathbf{1}}_1 & & & \boldsymbol{O}'s \\ \boldsymbol{A}_{2,1} & \bar{\mathbf{1}}_2 & & \\ \vdots & \ddots & \ddots & \\ \bar{\boldsymbol{A}}_{s,1} & \ddots & \bar{\boldsymbol{A}}_{s,s-1} & \bar{\mathbf{1}}_s \end{bmatrix}; \quad \bar{\boldsymbol{A}}_{j,i} \equiv \boldsymbol{O}, \text{ if } M_j \notin \gamma_i \tag{10}$$

$$\bar{\boldsymbol{N}}_d \equiv \text{diag}\,[\bar{\boldsymbol{N}}_1 \quad \cdots \quad \bar{\boldsymbol{N}}_i \quad \cdots \quad \bar{\boldsymbol{N}}_s]$$

where $\bar{\mathbf{1}}_i$ is the $6n^i \times 6n^i$ identity and matrix and γ_i stands for the set of kinematic modules originating from module M_i including M_i as shown within the dashed boundary of figure 3. The matrices $\bar{\boldsymbol{N}}_l$ and $\bar{\boldsymbol{N}}_d$ are the desired DeNOC matrices for the system under study written in terms of the module information rather than in terms of the link information (Saha, 1999a).

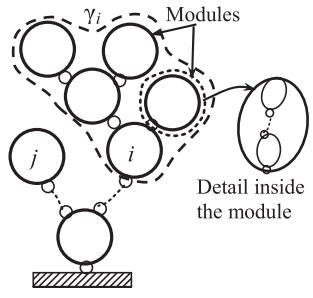

Figure 3. Definition of γ_i.

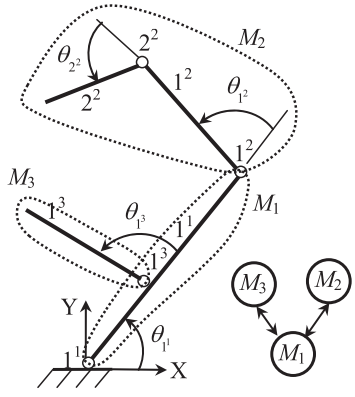

Figure 4. A gripper and its modularization.

2.2 Equations of motion

As vector of constraint forces and moments is orthogonal to the columns of velocity transformation matrix (Angeles an Ma, 1988) one may show that the pre-multiplication of equation (3) by $\bar{\boldsymbol{N}}_d^T \bar{\boldsymbol{N}}_l^T$ yields the minimal set of equations of motion, where the constraint wrenches are eliminated, i.e.,

$$\bar{\boldsymbol{N}}_d^T \bar{\boldsymbol{N}}_l^T (\boldsymbol{M}\dot{\boldsymbol{t}} + \boldsymbol{\Omega M E t}) = \bar{\boldsymbol{N}}_d^T \bar{\boldsymbol{N}}_l^T \boldsymbol{w}^E \tag{11}$$

where $\boldsymbol{w}^E$ is the generalized wrench due to external moments and forces. Substituting equations (9) and its time derivative into equation (11), and rearranging them, the following equations of motion is obtained

$$\boldsymbol{I}\ddot{\boldsymbol{\theta}} + \boldsymbol{C}\dot{\boldsymbol{\theta}} = \boldsymbol{\tau} \tag{12}$$

where the expressions of the $n \times n$ Generalized Inertia Matrix (GIM) $\boldsymbol{I}$, the $n \times n$ Matrix of Convective Inertia (MCI) $\boldsymbol{C}$, and the n-dimensional vector of the generalized external forces $\boldsymbol{\tau}$ are given as

$$\boldsymbol{I} \equiv \boldsymbol{N}^T \boldsymbol{M N},\ \boldsymbol{C} \equiv \boldsymbol{N}^T (\boldsymbol{M}\dot{\boldsymbol{N}} + \boldsymbol{\Omega M E N}), \boldsymbol{\tau} \equiv \boldsymbol{N}^T \boldsymbol{w}^E \tag{13}$$

where $\boldsymbol{N} \equiv \bar{\boldsymbol{N}}_l \bar{\boldsymbol{N}}_d$.

3. Recursive Dynamics

Based on the dynamic modeling presented in section 2, recursive inverse and forward dynamics algorithms can be developed. In order to comprehend the steps better, the algorithms are developed for a four-DOF tree-type gripper shown in figure 4. However, a general algorithm applicable to any serial or tree type open-loop system can be written. The gripper in figure 4 can be used for holding an object to be manipulated, say, by a robot hand. It has three kinematic modules, namely, M_1, M_2, and M_3. Module M_1 has one link, module M_2 has two links and

module M_3 has one link. Hence, the 24×24 matrix $\bar{\boldsymbol{N}}_l$ and 24×4 matrix $\bar{\boldsymbol{N}}_d$ are given by

$$\bar{\boldsymbol{N}}_l \equiv \begin{bmatrix} \bar{\mathbf{1}}_1 & & \boldsymbol{O}'s \\ \bar{\boldsymbol{A}}_{2,1} & \bar{\mathbf{1}}_2 & \\ \bar{\boldsymbol{A}}_{3,1} & \boldsymbol{O} & \bar{\mathbf{1}}_3 \end{bmatrix}, \quad \text{and} \quad \bar{\boldsymbol{N}}_d \equiv \begin{bmatrix} \bar{\boldsymbol{N}}_1 & & \boldsymbol{O}'s \\ & \bar{\boldsymbol{N}}_2 & \\ \boldsymbol{O}'s & & \bar{\boldsymbol{N}}_3 \end{bmatrix} \tag{14}$$

3.1 Inverse dynamics

In inverse dynamics one attempts to find the joint torques and forces for a given set of joint motions, i.e., joint positions, velocity and acceleration. Table 1 shows the steps which comprise of intra-and inter-modular computations. In step 1, i.e., forward recursion, module-twist $\bar{\boldsymbol{t}}_i$, module-twist-rate $\dot{\bar{\boldsymbol{t}}}_i$, and module inertia wrench $\bar{\boldsymbol{w}}_i^*$ are obtained using the information of the parent module, whereas in step 2, i.e., backward recursion, joint torques and forces are obtained based on equation (12). It may be noted here that the inter-modular steps are nothing but the compact representations of the intra-modular steps and hence, they are equivalent.

Numerical values for the joint torques for a given set of input motions are then obtained using a MATLAB (2010) program developed for this purpose. The trajectory for each joint is taken as per equation (15), whereas, the initial and final joint angles are denoted as $\theta(0)$ and $\theta(T)$, respectively, are given in table 2.

$$\theta = \theta(0) + \frac{\theta(T) - \theta(0)}{T}\left[t - \frac{T}{2\pi}\sin\left(\frac{2\pi}{T}t\right)\right] \tag{15}$$

The joint trajectory in equation (15) is so chosen that the initial and final joint rates and accelerations for all joints are zeros. This ensures smooth motion of the joints. The lengths and masses of the links are taken as

$$l_{1^1} = 0.1\,\text{m}, \quad l_{1^2} = l_{2^2} = l_{1^3} = 0.05\,\text{m}, \quad m_{1^1} = 0.4\,\text{Kg}, \quad \text{and}$$
$$m_{1^2} = m_{2^2} = m_{1^3} = 0.2\,\text{Kg}.$$

Figure 5 shows the joint torques required at joints 1^1 and 1^2 over a time interval of $T = 1$ second to follow the prescribed trajectories given by equation (15). In order to validate the results a CAD model of the gripper mechanism was developed in ADAMS (2005) software and used for computation joint torques. The results are shown in figure 5, which show close match with those obtained using the proposed inverse dynamics algorithm. Hence the numerical results are validated. As far as the computational complexity is concerned, the proposed algorithm implemented in MATLAB environment took only 0.025 sec compared to 1.95 sec by ADAMS software when implemented in Intel T2300@1.66 GHz computing system.

3.2 Forward dynamics

The forward dynamics algorithm attempts to find the joint acceleration form the equation of motion of equation (12), while joint torques and forces are known.

Table 1. Inverse dynamics.

Step 1. Forward recursion			
Module M_i	Inter-modular	Link k^i	Intra-modular
$i=1$	$\bar{\boldsymbol{t}}_1 = \bar{\boldsymbol{N}}_1\dot{\bar{\boldsymbol{\theta}}}_1;\ \dot{\bar{\boldsymbol{t}}}_1 = \dot{\bar{\boldsymbol{N}}}_1\dot{\bar{\boldsymbol{\theta}}}_1 + \bar{\boldsymbol{N}}_1\ddot{\bar{\boldsymbol{\theta}}}_1$ $\bar{\boldsymbol{w}}_1^* \equiv \bar{\boldsymbol{M}}_1\dot{\bar{\boldsymbol{t}}}_1 + \bar{\boldsymbol{\Omega}}_1\bar{\boldsymbol{M}}_1\bar{\boldsymbol{E}}_1\bar{\boldsymbol{t}}_1$	$k=1^1$	$\boldsymbol{t}_{1^1} = \boldsymbol{p}_{1^1}\dot{\theta}_{1^1};\ \dot{\boldsymbol{t}}_{1^1} = \boldsymbol{\Omega}_{1^1}\boldsymbol{p}_{1^1}\dot{\theta}_{1^1} + \boldsymbol{p}_{1^1}\ddot{\theta}_{1^1}$ $\boldsymbol{w}_{1^1}^* \equiv \boldsymbol{M}_{1^1}\dot{\boldsymbol{t}}_{1^1} + \boldsymbol{\Omega}_{1^1}\boldsymbol{M}_{1^1}\boldsymbol{E}_{1^1}\boldsymbol{t}_{1^1}$
$i=2$	$\bar{\boldsymbol{t}}_2 = \bar{\boldsymbol{A}}_{2,1}\bar{\boldsymbol{t}}_1 + \bar{\boldsymbol{N}}_2\dot{\bar{\boldsymbol{\theta}}}_2$ $\dot{\bar{\boldsymbol{t}}}_2 = \dot{\bar{\boldsymbol{A}}}_{2,1}\bar{\boldsymbol{t}}_1 + \bar{\boldsymbol{A}}_{2,1}\dot{\bar{\boldsymbol{A}}}_{2,1}\dot{\bar{\boldsymbol{t}}}_1 + \dot{\bar{\boldsymbol{N}}}_2\dot{\bar{\boldsymbol{\theta}}}_2 + \bar{\boldsymbol{N}}_2\ddot{\bar{\boldsymbol{\theta}}}_2$ $\bar{\boldsymbol{w}}_2^* \equiv \bar{\boldsymbol{M}}_2\dot{\bar{\boldsymbol{t}}}_2 + \bar{\boldsymbol{\Omega}}_2\bar{\boldsymbol{M}}_2\bar{\boldsymbol{E}}_2\bar{\boldsymbol{t}}_2$	$k=1^2$	$\boldsymbol{t}_{1^2} = \boldsymbol{A}_{1^2,1^1}\boldsymbol{t}_{1^1} + \boldsymbol{p}_{1^2}\dot{\theta}_{1^2};\ \dot{\boldsymbol{t}}_{1^2} = \boldsymbol{A}_{1^2,1^1}\dot{\boldsymbol{t}}_{1^1} + \dot{\boldsymbol{A}}_{1^2,1^1}\boldsymbol{t}_{1^1} + \boldsymbol{\Omega}_{1^2}\boldsymbol{p}_{1^2}\dot{\boldsymbol{\theta}}_{1^2} + \boldsymbol{p}_{1^2}\ddot{\theta}_{1^2}$ $\boldsymbol{w}_{1^2}^* \equiv \boldsymbol{M}_{1^2}\dot{\boldsymbol{t}}_{1^2} + \boldsymbol{\Omega}_{1^2}\boldsymbol{M}_{1^2}\boldsymbol{E}_{1^2}\boldsymbol{t}_{1^2}$
		$k=2^2$	$\boldsymbol{t}_{2^2} = \boldsymbol{A}_{2^2,1^2}\boldsymbol{t}_{1^2} + \boldsymbol{p}_{2^2}\dot{\theta}_{2^2};\ \dot{\boldsymbol{t}}_{2^2} = \boldsymbol{A}_{2^2,1^2}\dot{\boldsymbol{t}}_{1^2} + \dot{\boldsymbol{A}}_{2^2,1^2}\boldsymbol{t}_{1^2} + \boldsymbol{\Omega}_{2^2}\boldsymbol{p}_{2^2}\dot{\boldsymbol{\theta}}_{2^2} + \boldsymbol{p}_{2^2}\ddot{\theta}_{2^2}$ $\boldsymbol{w}_{2^2}^* \equiv \boldsymbol{M}_{2^2}\dot{\boldsymbol{t}}_{2^2} + \boldsymbol{\Omega}_{2^2}\boldsymbol{M}_{2^2}\boldsymbol{E}_{2^2}\boldsymbol{t}_{2^2}$
$i=3$	$\bar{\boldsymbol{t}}_3 = \bar{\boldsymbol{A}}_{3,1}\bar{\boldsymbol{t}}_1 + \bar{\boldsymbol{N}}_3\dot{\bar{\boldsymbol{\theta}}}_3$ $\dot{\bar{\boldsymbol{t}}}_3 = \dot{\bar{\boldsymbol{A}}}_{3,1}\bar{\boldsymbol{t}}_1 + \bar{\boldsymbol{A}}_{3,1}\dot{\bar{\boldsymbol{t}}}_1 + \dot{\bar{\boldsymbol{N}}}_3\dot{\bar{\boldsymbol{\theta}}}_3 + \bar{\boldsymbol{N}}_3\ddot{\bar{\boldsymbol{\theta}}}_3$ $\bar{\boldsymbol{w}}_3^* \equiv \bar{\boldsymbol{M}}_3\dot{\bar{\boldsymbol{t}}}_3 + \bar{\boldsymbol{\Omega}}_3\bar{\boldsymbol{M}}_3\bar{\boldsymbol{E}}_3\bar{\boldsymbol{t}}_3$	$k=1^3$	$\boldsymbol{t}_{1^3} = \boldsymbol{A}_{1^3,1^1}\boldsymbol{t}_{1^1} + \boldsymbol{p}_{1^3}\dot{\theta}_{1^3}$ $\dot{\boldsymbol{t}}_{1^3} = \boldsymbol{A}_{1^3,1^1}\dot{\boldsymbol{t}}_{1^1} + \dot{\boldsymbol{A}}_{1^3,1^1}\boldsymbol{t}_{1^1} + \boldsymbol{\Omega}_{1^3}\boldsymbol{p}_{1^3}\dot{\theta}_{1^3} + \boldsymbol{p}_{1^3}\ddot{\theta}_{1^3}$ $\boldsymbol{w}_{1^3}^* \equiv \boldsymbol{M}_{1^3}\dot{\boldsymbol{t}}_{1^3} + \boldsymbol{\Omega}_{1^3}\boldsymbol{M}_{1^3}\boldsymbol{E}_{1^3}\boldsymbol{t}_{1^3}$
Step 2. Backward recursion			
$i=3$	$\bar{\boldsymbol{\tau}}_3 = \bar{\boldsymbol{N}}_3^T\tilde{\bar{\boldsymbol{w}}}_3^*;\ \tilde{\bar{\boldsymbol{w}}}_3^* = \bar{\boldsymbol{w}}_3^*$	$k=1^3$	$\tau_{1^3} = \boldsymbol{p}_{1^3}^T\tilde{\boldsymbol{w}}_{1^3}^*;\ \tilde{\boldsymbol{w}}_{1^3}^* = \boldsymbol{w}_{1^3}^*$
$i=2$	$\bar{\boldsymbol{\tau}}_2 = \bar{\boldsymbol{N}}_2^T\tilde{\bar{\boldsymbol{w}}}_2^*;\ \tilde{\bar{\boldsymbol{w}}}_2^* = \bar{\boldsymbol{w}}_2^*$	$k=2^2$	$\tau_{2^2} = \boldsymbol{p}_{2^2}^T\tilde{\boldsymbol{w}}_{2^2}^*;\ \tilde{\boldsymbol{w}}_{2^2} = \boldsymbol{w}_{2^2}^*$
		$k=1^2$	$\tau_{1^2} = \boldsymbol{p}_{1^2}^T\tilde{\boldsymbol{w}}_{1^2}^*;\ \tilde{\boldsymbol{w}}_{1^2}^* = \boldsymbol{w}_{1^2}^* + \boldsymbol{A}_{2^2 1^2}\tilde{\boldsymbol{w}}_{2^2}^*$
$i=1$	$\bar{\boldsymbol{\tau}}_1 = \bar{\boldsymbol{N}}_1^T\tilde{\bar{\boldsymbol{w}}}_1^*;$ $\tilde{\bar{\boldsymbol{w}}}_1^* = \bar{\boldsymbol{w}}_1^* + \bar{\boldsymbol{A}}_{2,1}\tilde{\bar{\boldsymbol{w}}}_2^* + \bar{\boldsymbol{A}}_{3,1}\tilde{\bar{\boldsymbol{w}}}_3^*$	$k=1^1$	$\tau_{1^1} = \boldsymbol{p}_{1^1}^T\tilde{\boldsymbol{w}}_{1^1}^*;\ \tilde{\boldsymbol{w}}_{1^1}^* = \boldsymbol{w}_{1^1}^* + \boldsymbol{A}_{1^2 1^1}\tilde{\boldsymbol{w}}_{1^2}^* + \boldsymbol{A}_{1^3,1^1}\tilde{w}_{1^3}^*$

Table 2. Joint motions.

Joint	$\theta(0)$	$\theta(T)$
1^1	0	60°
1^2	0	80°
2^2	0	80°
1^3	90°	120°

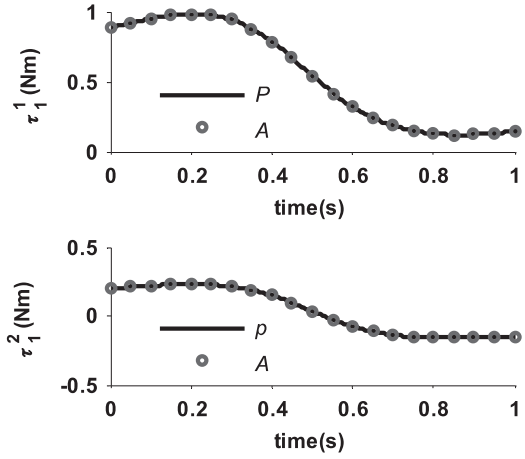

Figure 5. Torques at joints 1^1 and 1^2 (P-Proposed, A-Adams).

Numerical integration of these joint accelerations will then allow one to find the system configuration at every time instance. The above forward dynamics computations followed by numerical integration are refereed in literature as simulation. Unlike a recursive inverse dynamics algorithm presented above, a recursive forward dynamics algorithm is rather complex. Inverse dynamics calculations require only matrix-vector operations as shown in the previous sub-section. In the case of forward dynamics, solution of the joint accelerations from equation (12) is required, necessitating, an order (n^3) numerical calculation. For large n, such algorithms should be avoided due to poor computational efficiency and problems associated with numerical instability (Mohan and Saha, 2007). Hence, recursive order (n) forward dynamics algorithm is preferred. Here one such algorithm is presented, which is illustrated with 4-DOF gripper shown in figure 4. Steps are given below:

1) The GIM of the 3-module gripper system is obtained using equation (13) as

$$\begin{aligned} \boldsymbol{I} &\equiv \bar{\boldsymbol{N}}_d^T(\bar{\boldsymbol{N}}_l^T \boldsymbol{M} \bar{\boldsymbol{N}}_l)\bar{\boldsymbol{N}}_d \\ &\equiv \begin{bmatrix} \bar{\boldsymbol{N}}_1^T \tilde{\bar{\boldsymbol{M}}}_1 \bar{\boldsymbol{N}}_1 & & \text{sym} \\ \bar{\boldsymbol{N}}_2^T \tilde{\bar{\boldsymbol{M}}}_2 \bar{\boldsymbol{A}}_{2,1} \bar{\boldsymbol{N}}_1 & \bar{\boldsymbol{N}}_2^T \tilde{\bar{\boldsymbol{M}}}_2 \bar{\boldsymbol{N}}_2 & \\ \bar{\boldsymbol{N}}_3^T \tilde{\bar{\boldsymbol{M}}}_3 \bar{\boldsymbol{A}}_{3,1} \bar{\boldsymbol{N}}_1 & \boldsymbol{O} & \bar{\boldsymbol{N}}_3^T \tilde{\bar{\boldsymbol{M}}}_3 \bar{\boldsymbol{N}}_3 \end{bmatrix} \end{aligned} \tag{16}$$

where $\boldsymbol{I}$ is the 4×4 matrix. In equation (16), the block elements $\bar{\boldsymbol{N}}_i$ and $\bar{\boldsymbol{A}}_{j,i}$ are obtained in equation (7). Moreover, the mass matrix of the composite module $\tilde{\bar{\boldsymbol{M}}}_i$, for $i = 3, 2, 1$, is obtained recursively as

$$\tilde{\bar{\boldsymbol{M}}}_3 = \bar{\boldsymbol{M}}_3, \tilde{\bar{\boldsymbol{M}}}_2 = \bar{\boldsymbol{M}}_2, \tilde{\bar{\boldsymbol{M}}}_1 = \bar{\boldsymbol{M}}_1 + \sum_{j=3,2} \bar{\boldsymbol{A}}_{j,1}^T \tilde{\bar{\boldsymbol{M}}}_j \bar{\boldsymbol{A}}_{j,1} \tag{17}$$

2) Following the Block Reverse Gaussian Elimination (BRGE) similar to the RGE of (Saha, 1997) for serial-type system, the GIM in equation (16) is decomposed as $\boldsymbol{I} = \bar{\boldsymbol{U}} \bar{\boldsymbol{D}} \bar{\boldsymbol{U}}^T$, where the 4×4 block upper triangular matrix $\bar{\boldsymbol{U}}$ and block diagonal matrix $\bar{\boldsymbol{D}}$ have following representations:

$$\bar{\boldsymbol{U}} \equiv \begin{bmatrix} \bar{\mathbf{1}}_1 & \bar{\boldsymbol{N}}_1^T \bar{\boldsymbol{A}}_{2,1}^T \bar{\boldsymbol{\Psi}}_2 & \bar{\boldsymbol{N}}_1^T \bar{\boldsymbol{A}}_{3,1}^T \bar{\boldsymbol{\Psi}}_3 \\ & \bar{\mathbf{1}}_2 & \boldsymbol{O} \\ \bar{\boldsymbol{O}}'s & & \bar{\mathbf{1}}_3 \end{bmatrix}, \quad \text{and}$$

$$\bar{\boldsymbol{D}} \equiv \begin{bmatrix} \bar{\boldsymbol{N}}_1^T \hat{\bar{\boldsymbol{\Psi}}}_1 & & \boldsymbol{O}'s \\ & \bar{\boldsymbol{N}}_2^T \hat{\bar{\boldsymbol{\Psi}}}_2 & \\ \boldsymbol{O}'s & & \bar{\boldsymbol{N}}_3^T \hat{\bar{\boldsymbol{\Psi}}}_3 \end{bmatrix} \tag{18}$$

block elements $\bar{\boldsymbol{\Psi}}_i$, and $\hat{\bar{\boldsymbol{\Psi}}}_i$, for $i = 1, 2$, and 3, being expressed as

For $i = 1, 2, 3$

$$\hat{\bar{\boldsymbol{\Psi}}}_i = \hat{\bar{\boldsymbol{M}}}_i \bar{\boldsymbol{N}}_i \quad \text{and} \quad \bar{\boldsymbol{\Psi}}_i = \hat{\bar{\boldsymbol{\Psi}}}_i \hat{\bar{\boldsymbol{I}}}_i^{-1} \quad \text{where} \quad \hat{\bar{\boldsymbol{I}}}_i = \bar{\boldsymbol{N}}_i^T \hat{\bar{\boldsymbol{\Psi}}}_i \tag{19}$$

In equation (19), the mass matrix of articulated module $\hat{\bar{\boldsymbol{M}}}_i$ similar to the articulated body inertia of a serial-type system, for $i = 3, 2, 1$, is obtained recursively as

$$\hat{\bar{\boldsymbol{M}}}_3 = \bar{\boldsymbol{M}}_3, \hat{\bar{\boldsymbol{M}}}_2 = \bar{\boldsymbol{M}}_2, \hat{\bar{\boldsymbol{M}}}_1 = \bar{\boldsymbol{M}}_1 + \sum_{j=3,2} \bar{\boldsymbol{A}}_{j,1}^T \bar{\boldsymbol{\Phi}}_j \hat{\bar{\boldsymbol{M}}}_j \bar{\boldsymbol{A}}_{j,1} \tag{20}$$

where the $6\eta^i \times 6\eta^i$ matrix $\bar{\boldsymbol{\phi}}_j$ is obtained as

$$\bar{\boldsymbol{\phi}}_j = \mathbf{1} - \bar{\boldsymbol{\Psi}}_j \boldsymbol{N}_j^T, \quad \text{for} \quad j = 2, 3 \tag{21}$$

3) Based on the $\bar{\boldsymbol{U}} \bar{\boldsymbol{D}} \bar{\boldsymbol{U}}^T$ decomposition of the GIM, the constrained equations of motion, equation (12), are rewritten as follows:

$$\bar{\boldsymbol{U}} \bar{\boldsymbol{D}} \bar{\boldsymbol{U}}^T \ddot{\boldsymbol{\theta}} = \boldsymbol{\varphi}, \quad \text{where} \quad \boldsymbol{\varphi} = \boldsymbol{\tau} - \boldsymbol{C}\dot{\boldsymbol{\theta}} \tag{22}$$

The joint accelerations are then solved using three sets of linear algebraic equations, namely

$$1)\ \bar{\boldsymbol{U}} \hat{\bar{\boldsymbol{\varphi}}} = \boldsymbol{\varphi}, \quad \text{where} \quad \hat{\bar{\boldsymbol{\varphi}}} = \bar{\boldsymbol{D}} \bar{\boldsymbol{U}}^T \ddot{\boldsymbol{\theta}}$$

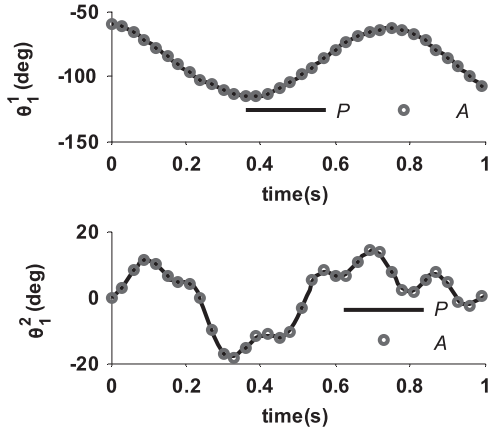

Figure 6. Simulated joint angles 1^1 and 1^2 (P-Proposed, A-Adams).

$$\begin{aligned} &2)\ \bar{\boldsymbol{D}}\tilde{\ddot{\boldsymbol{\varphi}}} = \hat{\ddot{\boldsymbol{\varphi}}}, \quad \text{where} \quad \tilde{\ddot{\boldsymbol{\varphi}}} = \bar{\boldsymbol{U}}^T \ddot{\boldsymbol{\theta}} \\ &3)\ \bar{\boldsymbol{U}}^T \ddot{\boldsymbol{\theta}} = \tilde{\ddot{\boldsymbol{\varphi}}} \end{aligned} \tag{23}$$

The Recursive forward dynamics algorithm to obtain the joint acceleration using equation (23) is presented in table 3. In order to validate the proposed recursive forward dynamics algorithm, a MATLAB code is written. Numerical results are obtained for the fall of the gripper under gravity. The accelerations were numerically integrated twice using fixed step ordinary differential equation solver, namely, "ode 5" of MATLAB for the following initial joint angles and rates:

$$\theta_{1^1} = -60^\circ, \theta_{2^1} = \theta_{2^2} = \theta_{1^3} = 0, \quad \text{and} \quad \dot{\theta}_{1^1} = \dot{\theta}_{1^2} = \dot{\theta}_{2^2} = \dot{\theta}_{1^3} = 0.$$

The step size was taken 0.01 sec. Figure 7 shows the plots for the simulated joint angles θ_{1^1} and θ_{1^2}.

The results were also compared with those obtained from ADAMS software for time duration of 1 second and step size of 0.01 sec. As per the computational complexity is concerned, it was found that the MATLAB program based on forward dynamics algorithm took only 0.42 sec compared to 2.1 sec by ADAMS. Hence, the customized proposed algorithm performed better than a commercial algorithm, which was expected.

4. Computational Efficiency

Recursive inverse and forward dynamics algorithms presented in the previous section have complexity of order (n) as shown in table 4. It is interesting to note that the computational complexity does not depend on the number of modules but on the number of links, n. In the case of inverse dynamics, the proposed algorithm performed as good as the fastest algorithm available in the literature, whereas the forward dynamics algorithm outperforms the algorithms available in the literature.

Table 3. Forward dynamics.

Step 1 and 2. $\bar{\boldsymbol{U}}\hat{\tilde{\boldsymbol{\varphi}}} = \boldsymbol{\varphi}$ and $\bar{\boldsymbol{D}}\tilde{\tilde{\boldsymbol{\varphi}}} = \hat{\tilde{\boldsymbol{\varphi}}}$			
Module M_i	Inter-modular	Link k^i	Inter-modular
$i = 3$	$\hat{\tilde{\boldsymbol{\varphi}}}_3 = \bar{\boldsymbol{\varphi}}_3; \tilde{\tilde{\boldsymbol{\varphi}}} = \hat{\tilde{\boldsymbol{I}}}_3^{-1}\hat{\tilde{\boldsymbol{\varphi}}}_3$	$k = 1^3$	$\hat{\varphi}_{1^3} = \varphi_{1^3}; \tilde{\varphi}_{1^3} = \hat{\varphi}_{1^3}/\hat{m}_{1^3}$
$i = 2$	$\hat{\tilde{\boldsymbol{\varphi}}}_2 = \bar{\boldsymbol{\varphi}}_2; \tilde{\tilde{\boldsymbol{\varphi}}} = \hat{\tilde{\boldsymbol{I}}}_2^{-1}\hat{\tilde{\boldsymbol{\varphi}}}_2$	$k = 2^2$	$\hat{\varphi}_{2^2} = \varphi_{2^2}; \tilde{\varphi}_{2^2} = \hat{\varphi}_{2^2}/\hat{m}_{2^2}$
		$k = 1^2$	$\hat{\varphi}_{1^2} = \varphi_{1^2} - \boldsymbol{P}_{1^2}^T\tilde{\boldsymbol{\eta}}_{1^2}; \tilde{\boldsymbol{\eta}}_{1^2} = \boldsymbol{A}_{2^2,1^2}^T\boldsymbol{\eta}_{2^2}; \boldsymbol{\eta}_{2^2} = \boldsymbol{\psi}_{2^2}\hat{\varphi}_{2^2}$ $\tilde{\varphi}_{1^2} = \hat{\varphi}_{1^2}/\hat{m}_{1^2}$
$i = 1$	$\hat{\tilde{\boldsymbol{\varphi}}}_1 = \bar{\boldsymbol{\varphi}}_1 - \bar{\boldsymbol{N}}_1^T\tilde{\bar{\boldsymbol{\eta}}}_1; \tilde{\bar{\boldsymbol{\eta}}}_1 = \bar{\boldsymbol{A}}_{2,1}^T\bar{\boldsymbol{\eta}}_2 + \bar{\boldsymbol{A}}_{3,1}^T\bar{\boldsymbol{\eta}}_3;$ $\bar{\boldsymbol{\eta}}_2 = \bar{\boldsymbol{\Psi}}_2\hat{\tilde{\boldsymbol{\varphi}}}_2; \bar{\boldsymbol{\eta}}_3 = \bar{\boldsymbol{\Psi}}_3\hat{\tilde{\boldsymbol{\varphi}}}_3$ $\tilde{\tilde{\boldsymbol{\varphi}}}_1 = \hat{\tilde{\boldsymbol{I}}}_1^{-1}\hat{\tilde{\boldsymbol{\varphi}}}_1$	$k = 1^1$	$\hat{\varphi}_{1^1} = \varphi_{1^1} - \boldsymbol{P}_{1^1}^T\tilde{\boldsymbol{\eta}}_{1^1}; \tilde{\boldsymbol{\eta}}_{1^1} = \boldsymbol{A}_{1^2,1^1}^T\boldsymbol{\eta}_{1^2} + \boldsymbol{A}_{1^3,1^1}^T\boldsymbol{\eta}_{1^3};$ $\boldsymbol{\eta}_{1^2} = \boldsymbol{\psi}_{1^2}\hat{\varphi}_{1^2} + \tilde{\boldsymbol{\eta}}_{1^2}; \boldsymbol{\eta}_{1^3} = \boldsymbol{\psi}_{1^3}\hat{\varphi}_{1^3}$ $\tilde{\varphi}_{1^1} = \hat{\varphi}_{1^1}/\hat{m}_{1^1}$
Step 3. $\bar{\boldsymbol{U}}^T\ddot{\boldsymbol{\theta}} = \tilde{\tilde{\boldsymbol{\varphi}}}$			
$i = 1$	$\ddot{\tilde{\boldsymbol{\theta}}}_1 = \tilde{\tilde{\boldsymbol{\varphi}}}_1$	$k = 1^1$	$\ddot{\theta}_1^1 = \tilde{\varphi}_{1^1}$
$i = 2$	$\ddot{\tilde{\boldsymbol{\theta}}}_2 = \tilde{\tilde{\boldsymbol{\varphi}}}_2 - \bar{\boldsymbol{\Psi}}_2^T\tilde{\bar{\boldsymbol{\mu}}}_2; \tilde{\bar{\boldsymbol{\mu}}}_2 = \bar{\boldsymbol{A}}_{2,1}\bar{\boldsymbol{\mu}}_1; \bar{\boldsymbol{\mu}}_1 = \bar{\boldsymbol{N}}_1\ddot{\tilde{\boldsymbol{\theta}}}_1$	$k = 1^2$	$\ddot{\theta}_{1^2} = \tilde{\varphi}_{1^2} - \boldsymbol{\psi}_{1^2}^T\tilde{\boldsymbol{\mu}}_{1^2}; \tilde{\boldsymbol{\mu}}_{1^2} = \boldsymbol{A}_{1^2 1^1}\boldsymbol{\mu}_{1^1}; \boldsymbol{\mu}_{1^1} = \boldsymbol{P}_{1^1}\ddot{\theta}_{1^1}$
		$k = 2^2$	$\ddot{\theta}_{2^2} = \tilde{\varphi}_{2^2} - \boldsymbol{\psi}_{2^2}^T\tilde{\boldsymbol{\mu}}_{2^2}; \tilde{\boldsymbol{\mu}}_{2^2} = \boldsymbol{A}_{2^2 1^2}, \boldsymbol{\mu}_{1^2};$ $\boldsymbol{\mu}_{1^2} = \boldsymbol{P}_{1^2}\ddot{\theta}_{1^2} + \tilde{\boldsymbol{\mu}}_{1^2}$
$i = 3$	$\ddot{\tilde{\boldsymbol{\theta}}}_3 = \tilde{\tilde{\boldsymbol{\varphi}}}_3 - \bar{\boldsymbol{\Psi}}_3^T\tilde{\bar{\boldsymbol{\mu}}}_3; \tilde{\bar{\boldsymbol{\mu}}}_3 = \bar{\boldsymbol{A}}_{3,1}\bar{\boldsymbol{\mu}}_1; \bar{\boldsymbol{\mu}}_1 = \bar{\boldsymbol{N}}_1\ddot{\tilde{\boldsymbol{\theta}}}_1$	$k = 1^3$	$\ddot{\theta}_{1^3} = \tilde{\varphi}_{1^3} - \boldsymbol{\psi}_{1^3}^T\tilde{\boldsymbol{\mu}}_{1^3}; \tilde{\boldsymbol{\mu}}_{1^3} = \boldsymbol{A}_{1^3 1^1}\boldsymbol{\mu}_{1^1}; \boldsymbol{\mu}_{1^1} = \boldsymbol{P}_{1^1}\ddot{\theta}_{1^1}$

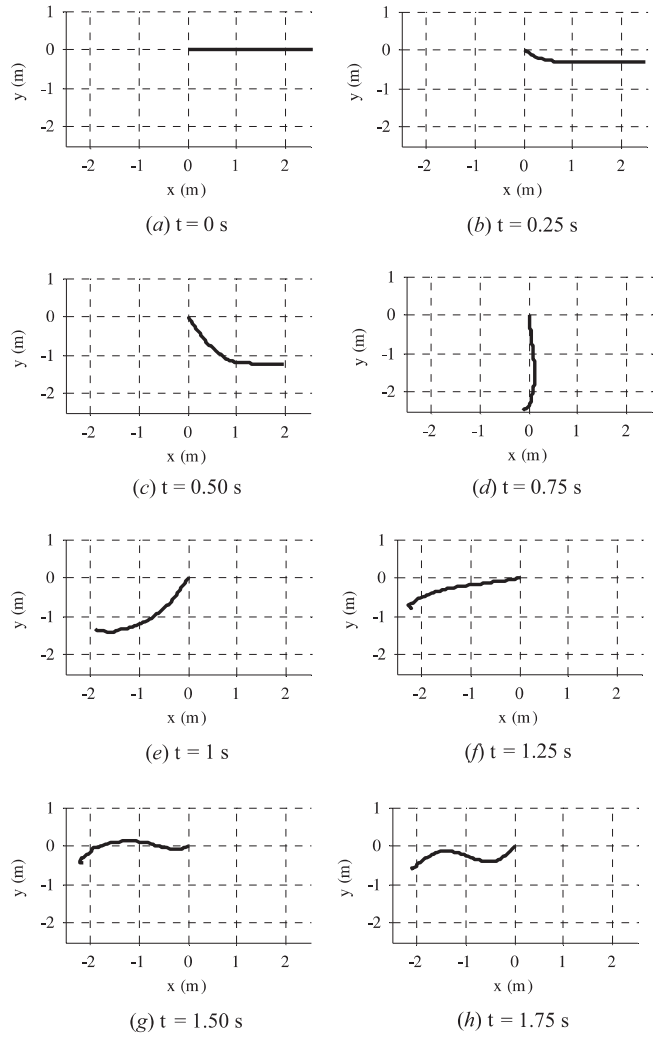

Figure 7. Simulation of 2.5 m rope.

It may be seen that for large n significant saving can be obtained. For that, highly flexible rope or hose (Fritzkowski and Kaminski, 2010) was simulated in which the system was considered to have many small rigid links.

The rope of 2.5 m length was considered to have 50 cylindrical elements. Each element is of 50 mm length, 5 mm radius, and 4 grams mass. Moreover, each link is connected with its parent link by a universal joint. As a result system has 100 DOF. Simulation under free-fall is performed for the time periods of 1.75 seconds using "ode 45" integrator of MATLAB. The configurations of the rope are shown in figure 7. The time taken was 3.44 sec. Comparison of the proposed algorithm with those available in the literature for the chain with the large DOF is shown in figure 8. The comparison showed that the proposed algorithm performed much better than

Table 4. Comparison of the computational count.

	Computational Count
Inverse dynamics	$(94n-84)M(82n-75)A$ (Proposed)
	$(93n-69)M(81n-65)A$ (Balafoutis *et al.*, 1991)
Forward dynamics	$(142n-124)M(139n-120)A$ (Proposed)
	$(173n-128)M(150n-133)A$ (Mohan and Saha, 2007)
	$(199n-198)M(174n-173)A$ (Featherstone, 1983)
	$\left(\frac{1}{6}n^3+10\frac{3}{2}n^2+40\frac{1}{3}n-51\right)M\left(\frac{1}{6}n^3+7n^2+50\frac{5}{6}n-51\right)A$ (Lilly and Orin, 1991)

M: Multiplication/Division; A: Addition/Subtraction

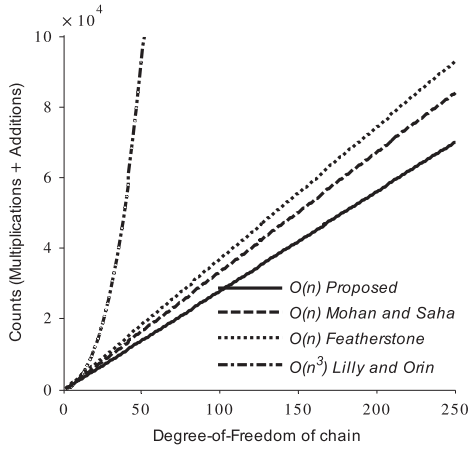

Figure 8. Comparison of counts for high DOF chain.

the algorithms available in the literature. Note that if the algorithm by Featherstone was used for the rope system CPU time of 4.54 sec would have been required in the same computer platform, i.e., E8400@3 GHz computing system.

5. Conclusions

This paper proposes recursive dynamics algorithms for open-loop multibody systems using the concept of decoupled natural orthogonal complement (DeNOC) matrices. However, the methodology is easily adaptable to closed-loop systems where the loops are cut open and replacing the cut joints with the appropriate constraint forces. The resulting system then becomes an open-loop system where the constraint forces need to be solved with other unknown. One such algorithm based on the DeNOC matrices is proposed in Chaudhary and Saha (2009). The proposed recursive algorithms were shown using the 4-DOF gripper and 100-DOF rope system. It is shown that the latter system, i.e., the system with large number of DOF, benefits more compared to any other similar algorithms available in the literature.

Acknowledgment

The authors would like to thank Dr. Sandipan Bandyopadhyay, Assistant Professor in the Department of Engineering Design at IIT Madras, for introducing the problem of multibody system with large Degrees-Of-Freedom like the rope system.

References

[1] Angeles, J. and Ma, O., "Dynamic Simulation of n-Axis Serial Robotic Manipulators using a Natural Orthogonal Complement," *International J. of Robotics Research*, **7**(5), pp. 32–47, 1988.

[2] Automated Dynamic Analysis of Mechanical System (ADAMS), 2004, Version 2005.0.0, MSC. Software.

[3] Bae, D. S. and Haug, E. J., "A Recursive Formulation for Constrained Mechanical System Dynamics: Part I, Open Loop Systems," *Mechanics of Structure and Machine*, **15**, pp. 359–382, 1987.

[4] Chaudhary, H. and Saha, S. K., *Dynamics and Balancing of Multibody Systems*, Springer, Berlin, 2009.

[5] Featherstone, R., "The Calculation of Robotic Dynamics using Articulated Body Inertias," *International J. of Robotics Research*, **2**, pp. 13–30, 1983.

[6] Fritzkowski, P. and Kaminski, H., "Dynamics of a Rope Modeled as a Multi-Body System with Elastic Joints," *Computational Mechanics*, **46**, pp. 901–909, 2010.

[7] Matlab, Version 7.4 Release 2007a, MathWorks, Inc, 2007.

[8] Mohan, A. and Saha, S. K., "A Recursive, Numerically Stable, and Efficient Algorithm for Serial Robots," *Multibody System Dynamics*, **17**(4), pp. 291–319, 2007.

[9] Rodriguez, G., Jain, A. and Kreutz-Delgado, K., "A Spatial Operator Algebra for Manipulator Modeling and Control," *International J. of Robotics Research*, **10**(4), pp. 371–381, 1991.

[10] Rosenthal, D. E., "An order n Formulation for Robotic Systems", *Journal of Astronautical Sciences*, **38**(4), pp. 511–529, 1990.

[11] Saha, S. K., "A Decomposition of the Manipulator Inertia Matrix," *IEEE Trans. on Robotics and Automation*, **13**(2), pp. 301–304, 1997.

[12] Saha, S. K., "Dynamics of Serial Multibody Systems using the Decoupled Natural Orthogonal Complement Matrices," *ASME J. of Applied Mechanics*, **66**, pp. 986–996, 1999a.

[13] Saha, S. K., "Analytical Expression for the Inverted Inertia Matrix of Serial Robots," International Journal of Robotic Research, **18**(1), pp. 116–124, 1999b.

[14] Shah, S. V., Saha, S. K. and Dutt, J. K., "Modular Recursive Dynamics Algorithm for Multibody Systems using the DeNOC Matrices", *Eccomas Thematic Conference on Multibody Dynamics*, Warsaw, Poland, 2009.