On the Improvement of a Fully Recursive Formulation for the Dynamic Analysis of Multibody Systems

Sheen-Gil Kang, Yong-San Yoon*
Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, ME3022,
Yusong-gu, Daejon 305-701, Korea

Virtual work in multibody systems is frequently expressed as the inner product of the virtual displacement and the resultant force at the centroid. But provided that the resultant force is converted into the equipollent forces there is no restriction on where the analysis reference point is placed. There are basically three candidate points : the centroid, joint point and the instant global origin. The traditional fully recursive formulation uses the centroid, but the present work verifies that the instant global origin always shows better efficiency (e.g. 86% CPU time of the centroid for quarter car model) and joint point shows the efficiency between that of the centroid and the instant global origin. A discussion on how important it is to define the analysis reference point properly in a fully recursive formulation is also presented.

Key Words : Fully Recursive Formulation, Analysis Reference Frame, Multibody System, Dynamic Analysis

1. Introduction

The most important types of coordinate currently used to define the motion of multibody systems are joint coordinates and the Cartesian coordinates. These coordinate systems each have advantage, but the joint coordinate system may be the best choice if the simulation time is a critical factor. Remarkable improvements in the forward multibody dynamics have naturally come from the robotics applications requiring fast analysis. Luh and et al. (Luh, 1980) developed an efficient recursive O(N) algorithm for inverse dynamics of single open chained manipulator. This algorithm was reused by Walker and Orin (Walker, 1982) for an O(N^3) forward dynamic algorithm. Later, Tsai and Haug (Tsai, 1991a ; Tsai, 1991b) developed an O(N^3) algorithm for constrained multibody systems. They used two main ideas : the composite body concept and a 6 dimensional velocity state vector at the instant global origin without any explanation of physical meanings. The composite body concept is associated with Walker and Orin's (Walker, 1982) which enables us to construct a mass matrix of O(N^2) order instead of O(N^3), and the 6 dimensional velocity state vector is related with screw theory (e.g. Featherstone, 1987 ; McCarthy, 1990) that allows the velocity transformation matrix to be simple and compact. On the other hand, Amstrong (Amstrong, 1979) and Featherstone (Featherstone, 1983 ; Featherstone, 1987) developed a fully recursive O(N) algorithm. Afterward Bae and Haug (Bae, 1987 ; 1988) extended the algorithm to constrained multibody systems. But their algorithm for the centroidal reference frame is different from Featherstone (Featherstone, 1987) and Tsai and Haug (Tsai, 1991a). It is apparent that it is favorable to follow Featherstone's approach, but this fact has never been proved clearly in the open literature reported in the past, which motivates this study. Featherstone emphasized only the concise notation without noticing the numeri-

* Corresponding Author,
E-mail : ysyoon@kaist.ac.kr
TEL : +82-42-869-3022; **FAX** : +82-42-869-3210
Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology, ME3022, Yusong-gu, Daejon 305-701, Korea. (Manuscript **Received** April 22, 2002; **Revised** November 4, 2002)

cal advantages derivable from the screw notation. These advantages grow more for closed-loop systems that he had not dealt with.

The joint coordinates formulation can be improved further by several techniques such as sparcity of the system matrices (Negrut, 1997), parallel processing (e.g. Tsai, 1991b), subsystem technique (e.g. Kim, 1999) and so forth. In this study, a fully recursive formulation (Bae, 1987; 1988) based on the joint coordinates is improved by applying the alternative velocity transformation matrices associated with the screw theory. Although a fully recursive formulation may not always be the best choice for all types of systems, there is a good reason for improving this formulation since it is possible to analyze subsystems by using the same approach as an efficient formulation developed by Tsai and Haug (Tsai, 1991a). This unified approach enables very large systems to be analyzed quickly. It is because the best algorithm at the boundary of subsystems can be selected without any overheads associated with switching of algorithm.

2. Computation Methods

When a body-fixed centroidal reference frame is moving with acceleration $\ddot{\mathbf{r}}^C$ and angular acceleration $\dot{\omega}$ relative to the inertial frame, the variational Newton-Euler equations for the motion of the centroidal reference frame { C }(Haug, 1989),

$$\delta\mathbf{r}^{C^T}(m\ddot{\mathbf{r}}^C-\mathbf{F}^C)+\delta\boldsymbol{\pi}^T(\mathbf{J}^C\dot{\boldsymbol{\omega}}+\tilde{\boldsymbol{\omega}}\mathbf{J}^C\boldsymbol{\omega}-\boldsymbol{\tau})=0 \quad (1)$$

must hold for any virtual displacement vector $\delta\mathbf{r}^C$ of the centroid and virtual rotation vector $\delta\boldsymbol{\pi}$, or more compactly,

$$\delta\mathbf{Z}^{C^T}(\mathbf{M}^C\dot{\mathbf{Y}}^C-\mathbf{Q}^C)=0 \qquad (2)$$

where mass matrix and force vector are

$$\mathbf{M}^C=\begin{pmatrix} m\mathbf{I} & 0 \\ 0 & \mathbf{J}^C \end{pmatrix},\ \mathbf{Q}^C=\begin{pmatrix} \mathbf{F}^C \\ \boldsymbol{\tau}-\tilde{\boldsymbol{\omega}}\mathbf{J}^C\boldsymbol{\omega} \end{pmatrix} \qquad (3)$$

and displacement state vector and velocity state vector are

$$\delta\mathbf{Z}^C=\begin{pmatrix} \delta\mathbf{r}^C \\ \delta\boldsymbol{\pi} \end{pmatrix} \text{ and } \mathbf{Y}^C=\begin{pmatrix} \dot{\mathbf{r}}^C \\ \boldsymbol{\omega} \end{pmatrix}.$$

If the another reference frame { P } is defined at a position translated by $\mathbf{s}^P$ from the centroidal reference frame { C } without rotation, then the relation between the virtual displacement state vectors of the two frames, or

$$\begin{pmatrix} \delta\mathbf{r}^C \\ \delta\boldsymbol{\pi} \end{pmatrix}=\begin{pmatrix} \mathbf{I} & \tilde{\mathbf{s}}^P \\ 0 & \mathbf{I} \end{pmatrix}\begin{pmatrix} \delta\mathbf{r}^P \\ \delta\boldsymbol{\pi}^P \end{pmatrix} \text{ or } \delta\mathbf{Z}^C=\mathbf{T}^P\delta\mathbf{Z}^P \qquad (4)$$

holds and the relation between the respective velocity and the acceleration state vectors also hold as follows:

$$\mathbf{Y}^C=\mathbf{T}^P\mathbf{Y}^P \qquad (5)$$

$$\dot{\mathbf{Y}}^C=\mathbf{T}^P\dot{\mathbf{Y}}^P+\dot{\mathbf{T}}^P\mathbf{Y}^P \qquad (6)$$

Substituting Eqs. (4) and (6) into Eq. (2), the variational Newton-Euler equations of motion can be expressed for the origin of a new analysis reference frame { P } as

$$\delta\mathbf{Z}^{P^T}(\mathbf{M}^P\dot{\mathbf{Y}}^P-\mathbf{Q}^P)=0 \qquad (7)$$

where modified mass matrix and force vector are defined as

$$\mathbf{M}^P=\mathbf{T}^{P^T}\mathbf{M}^C\mathbf{T}^P,\ \mathbf{Q}^P=\mathbf{T}^{P^T}(\mathbf{Q}^C-\mathbf{M}^C\dot{\mathbf{T}}^P\mathbf{Y}^P) \qquad (8)$$

Here, $\mathbf{M}^P\dot{\mathbf{Y}}^P-\mathbf{Q}^P$ is the equipollent resultant force vector corresponding to the new analysis point.

In this study, we compared three basic points as shown in Fig. 1. Joint point is not explained further here but will be discussed later. The traditional fully recursive algorithm (Bae, 1987; 1988) uses the centroid as the reference point.

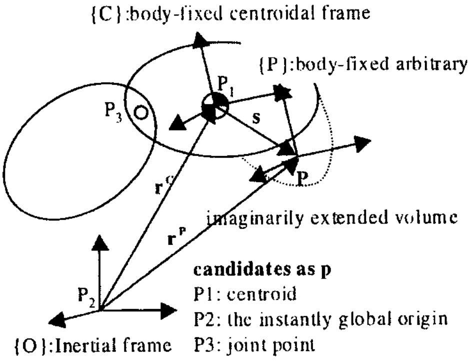
{C}:body-fixed centroidal frame
{P}:body-fixed arbitrary frame
P1
P3
s
P
r^C
r^P
imaginarily extended volume
P2
{O}:Inertial frame
candidates as p
P1: centroid
P2: the instantly global origin
P3: joint point

Fig. 1 Candidates for analysis reference point

Introducing the fully recursive algorithm, differences in the formulation according to the reference point used will be discussed.

Any multibody system can be spanned to tree like structure by cutting a joint per each independent closed loop. If a body of a pair of bodies in a chain from leaf to the root is nearer to root of the tree, it is called the lower body, and the other is called the upper body. Upper body **j** of the adjacent two bodies always has a unique lower body and a unique associated joint **j**.

Fig. 2 shows the kinematic configuration of the two adjacent bodies. In the figure, lower body **i** is denoted by body **j-1** to avoid confusing index **i** and **j** from vague printout. All body parameters (e.g. centroid, joints, forced points, etc.) are defined at body reference frame $\{\mathbf{O}_j\}$. This frame is defined at joint **j** but fixed on upper body **j**.

Provided that expressions for parameter conversion are well defined prior to the main analysis, no restrictions are placed on where the body reference frame should be defined (e.g. centroid with principal axes of body or inertial frame). But the final frame for the parameters to be expressed just before the analysis is the joint reference frame in the joint-coordinates formulation. In order to minimize the preprocessing work associated with body parameters, we defined the body reference frames at the joint.

From kinematics, the orientation matrix of the reference frame $\{\mathbf{O}_j\}$ is obtained recursively as

$$\mathbf{A}_j=\mathbf{A}_i\mathbf{C}_j\mathbf{R}_j \tag{9}$$

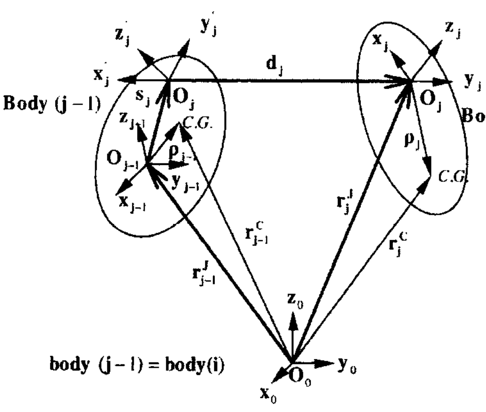

Fig. 2 Kinematics of adjacent bodies

where $\mathbf{C}_j$ is the constant orientation matrix of the reference frame $\{\mathbf{O}'_j\}$ to the adjacent lower reference frame $\{\mathbf{O}_i\}$ and $\mathbf{R}_j$ is the subsequent orientation matrix from the reference frame $\{\mathbf{O}_j\}$ to the reference frame $\{\mathbf{O}'_j\}$, which is the function of joint angular coordinates $\mathbf{q}^{\omega}_j$.

The orientation matrix of the centroid is obtained as

$$\mathbf{A}^C_j=\mathbf{A}_i\mathbf{C}_j\mathbf{R}_j\mathbf{C}^C_j \tag{10}$$

where $\mathbf{C}^C_j$ is the constant orientation matrix of the centroidal reference frame relative to the body reference frame $\{\mathbf{O}_j\}$. But matrix $\mathbf{A}^C_j$ is unnecessary for theoretical dynamic analysis.

Next, the relative positions are recovered from the body parameters as

$$\begin{aligned}\mathbf{s}_j&=\mathbf{A}_i\mathbf{s}'_j\\ \boldsymbol{\rho}_j&=\mathbf{A}_j\boldsymbol{\rho}'_j,\end{aligned} \tag{11}$$

where $\mathbf{s}'_j$, ρ'_j are constant vectors described in the body reference frames $\{\mathbf{O}_i\}$ and $\{\mathbf{O}_j\}$ respectively.

The translation of the reference frame $\{\mathbf{O}_j\}$ from the initial state is recovered from the joint coordinates as

$$\mathbf{d}_j=\mathbf{A}_i\mathbf{C}_j\mathbf{d}'_j=\mathbf{A}_i\mathbf{C}_j(\mathbf{d}'_{j,0}+\mathbf{q}^V_j) \tag{12}$$

where $\mathbf{q}^V_j$ is the translational coordinate of joint j.

The final translation of the reference frame $\{\mathbf{O}_j\}$ relative to the reference frame $\{\mathbf{O}_i\}$ is obtained as

$$\mathbf{r}^J_{ij}=\mathbf{s}_j+\mathbf{d}_j \tag{13}$$

Using the recovered vectors from the body parameters or joint coordinates, the origin of the body reference frame and the centroidal reference frame are obtained recursively as

$$\begin{aligned}\mathbf{r}^J_j&=\mathbf{r}^J_i+\mathbf{r}^J_{ij}\\ \mathbf{r}^C_j&=\mathbf{r}^J_j+\boldsymbol{\rho}_j\end{aligned} \tag{14}$$

The velocity transformation matrices are calculated from local matrices as

$$\begin{aligned}\mathbf{B}^{\omega}_j&=\mathbf{A}^C_i\mathbf{C}_j\mathbf{B}^{\omega'}_j\\ \mathbf{B}^V_j&=\mathbf{A}^C_i\mathbf{C}_j\mathbf{B}^{V'}_j\end{aligned} \tag{15}$$

where $\mathbf{B}^{V'}_j$ and $\mathbf{B}^{\omega'}_j$ can be obtained analytically as

$$\begin{aligned}\mathbf{B}^{\omega'}_j&=\partial(-[\dot{\mathbf{R}}_j\mathbf{R}^T_j]_{2,3},\ [\dot{\mathbf{R}}_j\mathbf{R}^T_j]_{1,3},\ -[\dot{\mathbf{R}}_j\mathbf{R}^T_j]_{1,2})^T/\partial\dot{\mathbf{q}}^{\omega}_j\\ \mathbf{B}^{V'}_j&=\partial\mathbf{d}'_j/\partial\mathbf{q}^V_j\end{aligned} \tag{16}$$

From the velocity transformation matrices, the relative angular velocities are obtained as

$$\boldsymbol{\omega}_{lj}=\mathbf{B}_j^{\omega}\dot{\mathbf{q}}_j^{\omega} \tag{17}$$

and the relative pure linear velocity due to translational coordinates bcomes

$$\mathbf{v}_{lj}=\mathbf{B}_j^{V}\dot{\mathbf{q}}_j^{V} \tag{18}$$

The angular velocity of body **j** is obtained as

$$\boldsymbol{\omega}_j=\boldsymbol{\omega}_l+\boldsymbol{\omega}_{lj} \tag{19}$$

From the first time derivative of the relative vectors given by

$$\begin{aligned}\dot{\mathbf{r}}_{lj}^{J}&=\tilde{\boldsymbol{\omega}}_l\mathbf{r}_{lj}^{J}+\mathbf{v}_{lj}\\ \dot{\boldsymbol{\rho}}_j&=\tilde{\boldsymbol{\omega}}_j\boldsymbol{\rho}_j\end{aligned} \tag{20}$$

the linear velocity at joint **j** is obtained recursively as

$$\begin{aligned}\dot{\mathbf{r}}_j^{J}&=\dot{\mathbf{r}}_l^{J}+\dot{\mathbf{r}}_{lj}^{J}\\ \dot{\mathbf{r}}_j^{C}&=\dot{\mathbf{r}}_j^{J}+\dot{\boldsymbol{\rho}}_j\end{aligned} \tag{21}$$

So far, all expressions are common irrespective of the analysis reference point but from nowon differences will emerge according to the reference point.

Method 1:

The velocity at centroid **j** can be alternatively expressed as

$$\dot{\mathbf{r}}_j^{C}=\dot{\mathbf{r}}_j^{J}-\tilde{\boldsymbol{\rho}}_j\boldsymbol{\omega}_j=\dot{\mathbf{r}}_l^{C}-\tilde{\mathbf{r}}_{lj}^{C}\boldsymbol{\omega}_l+\mathbf{B}_j^{V}\mathbf{q}_j^{V}-\tilde{\boldsymbol{\rho}}_j\mathbf{B}_j^{\omega}\dot{\mathbf{q}}^{\omega} \tag{22}$$

Defining the velocity state vector at the centroid such as

$$\mathbf{Y}^{C}=\begin{pmatrix}\dot{\mathbf{r}}_j^{C}\\ \boldsymbol{\omega}_j\end{pmatrix} \tag{23}$$

the velocity state vector relation between the adjacent two centroids is expressed as

$$\mathbf{Y}_j^{C}=\mathbf{B}_{j,1}^{C}\mathbf{Y}_l^{C}+\mathbf{B}_{j,2}^{C}\dot{\mathbf{q}}_j \tag{24}$$

where the velocity transformation matrices are defined as

$$\mathbf{B}_{j,1}^{C}=\begin{pmatrix}\mathbf{I} & -\tilde{\mathbf{r}}_{lj}^{C}\\ 0 & \mathbf{I}\end{pmatrix},\ \mathbf{B}_{j,2}^{C}=\begin{pmatrix}\mathbf{B}_j^{V} & -\tilde{\boldsymbol{\rho}}_j\mathbf{B}_j^{\omega}\\ 0 & \mathbf{B}_j^{\omega}\end{pmatrix} \tag{25}$$

Acceleration state vector relation is obtained from the time derivative of Eq. (24) as

$$\dot{\mathbf{Y}}_j^{C}=\mathbf{B}_{j,1}^{C}\dot{\mathbf{Y}}_l^{C}+\mathbf{B}_{j,2}^{C}\ddot{\mathbf{q}}_j+\mathbf{D}_j^{C} \tag{26}$$

where the velocity square term is obtained as

$$\mathbf{D}_j^{C}=\begin{pmatrix}\tilde{\boldsymbol{\omega}}_l(\dot{\mathbf{r}}_{lj}^{C}+\mathbf{v}_{lj})+\tilde{\boldsymbol{\omega}}_{lj}\dot{\boldsymbol{\rho}}_j-\tilde{\boldsymbol{\rho}}_j(\tilde{\boldsymbol{\omega}}_l\boldsymbol{\omega}_{lj}+\mathbf{A}_l\mathbf{C}_j\dot{\mathbf{B}}_j^{\omega'}\dot{\mathbf{q}}_j^{\omega})\\ \tilde{\boldsymbol{\omega}}_l\boldsymbol{\omega}_{lj}+\mathbf{A}_l\mathbf{C}_j\dot{\mathbf{B}}_j^{\omega'}\dot{\mathbf{q}}_j^{\omega}\end{pmatrix} \tag{27}$$

Here, $\dot{\mathbf{B}}_j^{\omega'}$ is the time derivative of $\mathbf{B}_j^{\omega'}$ in Eq. (16). All the calculations are repeated to the tree end bodies of the system.

Next, after the mass matrix and force vector are initialized by Eq. (3), they are transformed with the velocity transformation matrices $\mathbf{B}_{j,1}$ and $\mathbf{B}_{j,2}$ as

$$\begin{aligned}\mathbf{Q}_{j,1}&=\mathbf{B}_{j,1}^{T}(\mathbf{Q}_j-\mathbf{M}_j\mathbf{D}_j)\\ \mathbf{Q}_{j,2}&=\mathbf{B}_{j,2}^{T}(\mathbf{Q}_j-\mathbf{M}_j\mathbf{D}_j)\\ \mathbf{M}_{j,11}&=\mathbf{B}_{j,1}^{T}\mathbf{M}_j\mathbf{B}_{j,1}\\ \mathbf{M}_{j,21}&=\mathbf{B}_{j,2}^{T}\mathbf{M}_j\mathbf{B}_{j,1}\\ \mathbf{M}_{j,22}&=\mathbf{B}_{j,2}^{T}\mathbf{M}_j\mathbf{B}_{j,2}\end{aligned} \tag{28}$$

where superscript 'C' of **B**, and **Q** is omitted for convenience.

These transformed values are used to update the mass matrix and force vector of the adjacent lower body as

$$\begin{aligned}\mathbf{M}_l^{new}&=\mathbf{M}_l^{old}+\mathbf{M}_{j,11}-\mathbf{M}_{j,21}^{T}\mathbf{M}_{j,22}^{-1}\mathbf{M}_{j,21}\\ \mathbf{Q}_l^{new}&=\mathbf{Q}_l^{old}+\mathbf{Q}_{j,1}-\mathbf{M}_{j,21}^{T}\mathbf{M}_{j,22}^{-1}\mathbf{Q}_{j,2}\end{aligned} \tag{29}$$

and repeatedly applied until the ground is reached.

Finally, from the base body to the tree end body, the acceleration terms are determined recursively by

$$\begin{aligned}\ddot{\mathbf{q}}_j&=\mathbf{M}_{j,22}^{-1}(\mathbf{Q}_{j,2}-\mathbf{M}_{j,21}\dot{\mathbf{Y}}_l)\\ \dot{\mathbf{Y}}_j&=\mathbf{B}_{j,1}\dot{\mathbf{Y}}_l+\mathbf{B}_{j,2}\ddot{\mathbf{q}}_j+\mathbf{D}_j\end{aligned} \tag{30}$$

Comments

1. Bae and Haug (Bae, 1987; 1988) originally published this method. They calculated the inertia using $\mathbf{J}_j=(\mathbf{A}_j\mathbf{C}_j^{C})\mathbf{J}_j'(\mathbf{A}_j\mathbf{C}_j^{C})^{T}$. Afterward, Tsai and Haug (Tsai, 1991a) pointed out that the calculation of the inertia by $\mathbf{J}_j=\mathbf{A}_j(\mathbf{C}_j^{C}\mathbf{J}_j'\mathbf{C}_j^{C^{T}})\mathbf{A}_j^{T}$ is more efficient than the former version since it is enough to calculate or directly enter $\mathbf{C}_j^{C}\mathbf{J}_j'\mathbf{C}_j^{C^{T}}$ only once prior to the analysis.
2. Calculations of $\mathbf{Q}_{j,1}$, $\mathbf{M}_{j,11}$ and $\mathbf{M}_{j,21}$ in Eq. (28) require at least 6, 48, 6**n** scalar multiplications where **n** is the joint degree of freedom, respectively.

Method 2:

In this method, the velocity of the point that coincide with the global origin O at the instant but fixed on body j,

$$\dot{\mathbf{r}}_j^O=\dot{\mathbf{r}}_j^J+\tilde{\mathbf{r}}_j^J\boldsymbol{\omega}_j=\dot{\mathbf{r}}_i^O+\mathbf{B}_j^v\dot{\mathbf{q}}_j^v+\tilde{\mathbf{r}}_j^J\mathbf{B}_j^\omega\dot{\mathbf{q}}_j^\omega \tag{31}$$

is used. This point is constant with respect to the inertial frame but not constant with respect to the body-fixed reference frame.

With the velocity state vector defined at the instant global origin given by

$$\hat{\mathbf{Y}}_j=\begin{pmatrix}\dot{\mathbf{r}}_j^O\\ \boldsymbol{\omega}_j\end{pmatrix} \tag{32}$$

the velocity transformation matrices are defined differently from Eq. (25) as

$$\hat{\mathbf{B}}_{j,1}=\begin{pmatrix}\mathbf{I} & 0\\ 0 & \mathbf{I}\end{pmatrix},\ \hat{\mathbf{B}}_{j,2}=\begin{pmatrix}\mathbf{B}_j^v & \tilde{\mathbf{r}}_j^J\mathbf{B}_j^\omega\\ 0 & \mathbf{B}_j^\omega\end{pmatrix} \tag{33}$$

The acceleration state vector relations are

$$\dot{\hat{\mathbf{Y}}}_j=\hat{\mathbf{B}}_{j,1}\dot{\hat{\mathbf{Y}}}_i+\hat{\mathbf{B}}_{j,2}\ddot{\mathbf{q}}_j+\hat{\mathbf{D}}_j \tag{34}$$

where the velocity square term is

$$\hat{\mathbf{D}}_j=\begin{pmatrix}\tilde{\boldsymbol{\omega}}_i\mathbf{v}_{ij}+\dot{\tilde{\mathbf{r}}}_j^J\boldsymbol{\omega}_{ij}+\tilde{\mathbf{r}}_j^J(\tilde{\boldsymbol{\omega}}_i\boldsymbol{\omega}_{ij}+\mathbf{A}_i\mathbf{C}_j\dot{\mathbf{B}}_j^{\omega'}\dot{\mathbf{q}}_j^\omega)\\ \tilde{\boldsymbol{\omega}}_i\boldsymbol{\omega}_{ij}+\mathbf{A}_i\mathbf{C}_j\dot{\mathbf{B}}_j^{\omega'}\dot{\mathbf{q}}_j^\omega\end{pmatrix} \tag{35}$$

The translation from the centroid to the global origin is $\mathbf{s}^P=-\mathbf{r}_j^C$, thus $\mathbf{T}_j^P=\begin{pmatrix}\mathbf{I} & -\tilde{\mathbf{r}}_j^C\\ 0 & \mathbf{I}\end{pmatrix}$ in Eq. (4). Substituting $\mathbf{T}_j^P$ into Eq. (8), we obtain the modified mass matrix and force vector given by

$$\begin{aligned}\hat{\mathbf{M}}_j&=\begin{pmatrix}m_j\mathbf{I} & -m_j\tilde{\mathbf{r}}_j^C\\ m_j\tilde{\mathbf{r}}_j^C & \mathbf{J}_j^C-m_j\tilde{\mathbf{r}}_j^C\tilde{\mathbf{r}}_j^C\end{pmatrix}\\ \hat{\mathbf{Q}}_j&=\begin{pmatrix}\mathbf{F}_j^C+m_j\dot{\tilde{\mathbf{r}}}_j^C\boldsymbol{\omega}_j\\ \boldsymbol{\tau}_j-\tilde{\boldsymbol{\omega}}_j\mathbf{J}_j^C\boldsymbol{\omega}_j+\tilde{\mathbf{r}}_j^C(\mathbf{F}_j^C+m_j\dot{\tilde{\mathbf{r}}}_j^C\boldsymbol{\omega}_j)\end{pmatrix}\end{aligned} \tag{36}$$

Remaining steps have exactly the same forms as the traditional version except that Eqs. (33), (35), (36) take the place of (25), (27), (3).

Comments

1. Velocity transformation matrix $\hat{\mathbf{B}}_{j,1}$ is the identity matrix, so the matrix multiplications with $\mathbf{Q}_{j,1}$, $\mathbf{M}_{j,11}$ and $\mathbf{M}_{j,21}$ become unnecessary.
2. The calculation of $\dot{\hat{\mathbf{Y}}}_j$ in Eq. (34) requires 6 less scalar multiplications than in Eq. (26) since the calculation of $\hat{\mathbf{B}}_{j,1}\dot{\hat{\mathbf{Y}}}_i$ becomes unnecessary.
3. Construction of mass matrix and force vector in Eq. (36) requires 24 more scalar multiplications than in Eq. (3).

Loop closure constraints

Loop closure constraints can be expressed at the acceleration level as

$$\Phi_{Zi}^k\dot{\mathbf{Y}}_i+\Phi_{Zj}^k\dot{\mathbf{Y}}_j=\gamma^k \tag{37}$$

where k denotes the constraint number.

Tables 1 and 2 show the constraint Jacobian and the right hand side for basic constraint functions, respectively (Haug, 1989). All of the frequently used mechanical constraints are built from combinations of the basic constraint functions(Refer Haug, 1989).

If all of the constraints are initialized respectively by using tables 1, 2, they can be expressed compactly as

$$\Sigma\,\Phi_{Zj}\dot{\mathbf{Y}}_j=\gamma \tag{38}$$

where the dimension of Eq. (38) is equal to the number of total constraints of the system.

The constraint Jacobian of body j is transformed by

$$\begin{aligned}\Phi_{Zj,1}&=\Phi_{Zj}\mathbf{B}_{j,1}\\ \Phi_{Zj,2}&=\Phi_{Zj}\mathbf{B}_{j,2}\end{aligned} \tag{39}$$

Table 1 Jacobian of basic constraints

Constraint function	Φ_{Zi}		Φ_{Zj}	
$\Phi^{d1}(\mathbf{a}_i,\mathbf{a}_j)$	0	$-\mathbf{a}_j^T\tilde{\mathbf{a}}_i$	0	$-\mathbf{a}_i^T\tilde{\mathbf{a}}_j$
$\Phi^{d2}(\mathbf{a}_i,\mathbf{s}_i,\mathbf{s}_j)^{(1)}$	$-\mathbf{a}_i^T$	$\mathbf{a}_i^T\tilde{\mathbf{s}}_i-\mathbf{p}_{i,j}^T\tilde{\mathbf{a}}_i^{(2)}$	$\mathbf{a}_i^T$	$-\mathbf{a}_i^T\tilde{\mathbf{s}}_j$
$\Phi^{s}(\mathbf{s}_i,\mathbf{s}_j)$	$-\mathbf{I}$	$\tilde{\mathbf{s}}_i$	$\mathbf{I}$	$-\tilde{\mathbf{s}}_j$
$\Phi^{ss}(\mathbf{s}_i,\mathbf{s}_j)$	$-2\mathbf{p}_{i,j}^T$	$2\mathbf{p}_{i,j}^T\tilde{\mathbf{s}}_i$	$2\mathbf{p}_{i,j}^T$	$-2\mathbf{p}_{i,j}^T\tilde{\mathbf{s}}_j$

(1) $\mathbf{s}_i=\mathbf{p}_i-\mathbf{r}_i^C$

(2) $\mathbf{p}_{i,j}=\mathbf{p}_j-\mathbf{p}_i$

Table 2 Right hand side of basic constraints

Constraint function	γ
$\gamma^{d1}(\mathbf{a}_i,\mathbf{a}_j)$	$-\mathbf{a}_j^T\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i-2(\tilde{\boldsymbol{\omega}}_j\mathbf{a}_j)^T(\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i)-\mathbf{a}_i^T\tilde{\boldsymbol{\omega}}_j\tilde{\boldsymbol{\omega}}_j\mathbf{a}_j$
$\gamma^{d2}(\mathbf{a}_i,\mathbf{s}_i,\mathbf{s}_j)$	$-\mathbf{p}_{i,j}^T\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i-2\dot{\mathbf{p}}_{i,j}^T\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i-\mathbf{a}_i^T(\tilde{\boldsymbol{\omega}}_j\dot{\mathbf{s}}_j-\tilde{\boldsymbol{\omega}}_i\dot{\mathbf{s}}_i)$
$\gamma^{s}(\mathbf{s}_i,\mathbf{s}_j)$	$-\tilde{\boldsymbol{\omega}}_j\dot{\mathbf{s}}_j-\tilde{\boldsymbol{\omega}}_i\dot{\mathbf{s}}_i$
$\gamma^{ss}(\mathbf{s}_i,\mathbf{s}_j)$	$-2\mathbf{p}_{i,j}^T(\tilde{\omega}_j\dot{\mathbf{s}}_j-\tilde{\boldsymbol{\omega}}_i\dot{\mathbf{s}}_i)-2\dot{\mathbf{p}}_{i,j}^T\dot{\mathbf{p}}_{i,j}$

and updates the constraint Jacobian of the adjacent lower body i as

$$\Phi_{Zi}^{new}=\Phi_{Zi}^{old}+\Phi_{Zj,1}-\Phi_{Zj,2}\mathbf{M}_{j,22}^{-1}\mathbf{M}_{j,21} \tag{40}$$

Eqs. (39) and (40) are repeated from the tree end body to the ground.

The effect of each body on the reduced mass matrix Φ_λ of the system and the right hand side $\bar{\gamma}$ can be summed as

$$\begin{aligned}\Phi_\lambda&=\sum\Phi_{Zj,2}\mathbf{M}_{j,22}^{-1}\Phi_{Zj,2}^{T}\\ \bar{\gamma}&=\gamma-\sum(\Phi_{Zj,2}\mathbf{M}_{j,22}^{-1}\mathbf{Q}_{j,2}+\Phi_{Zj}\mathbf{D}_j)\end{aligned} \tag{41}$$

Finally the Lagrange multiplier λ is determined with the reduced matrix Φ_λ and the constraint right hand side $\bar{\gamma}$ by

$$\Phi_\lambda\lambda=\bar{\gamma} \tag{42}$$

Under the constraints, Eq. (30) must be modified as

$$\ddot{\mathbf{q}}_j=\mathbf{M}_{j,22}^{-1}(\mathbf{Q}_{j,2}-\mathbf{M}_{j,21}\dot{\mathbf{Y}}_i+\Phi_{Zj,2}^{T}\lambda) \tag{43}$$

By changing the joint definition frame by using Eq. (6) from the centroid to the instant global origin, we obtain the modified constraint equation as

$$\Phi_{Zi}^{k}\dot{\hat{\mathbf{Y}}}_i+\Phi_{Zj}^{k}\dot{\hat{\mathbf{Y}}}_j=\hat{\gamma}^k \tag{44}$$

where the modified Jacobian matrix and the right hand side of constraints are defined as

$$\begin{aligned}\Phi_{Zi}^{k}&=\Phi_{Zi}^{k}\mathbf{T}_i^{P}=(\Phi_{ric}^{k}\,\vdots\,-\Phi_{ric}^{k}\tilde{\mathbf{r}}_i^{C}+\Phi_{\pi i}^{k})\\ \Phi_{Zj}^{k}&=\Phi_{Zj}^{k}\mathbf{T}_j^{P}=(\Phi_{rjC}^{k}\,\vdots\,-\Phi_{rjC}^{k}\tilde{\mathbf{r}}_j^{C}+\Phi_{\pi j}^{k})\\ \hat{\gamma}^k&=\gamma^k-\Phi_{Zi}^{k}\dot{\mathbf{T}}_i^{P}\hat{\mathbf{Y}}_i-\Phi_{Zj}^{k}\dot{\mathbf{T}}_j^{P}\hat{\mathbf{Y}}_j\\ &=\gamma^k+\Phi_{Zi}^{k}\dot{\tilde{\mathbf{r}}}_i^{C}\boldsymbol{\omega}_i+\Phi_{Zj}^{k}\dot{\tilde{\mathbf{r}}}_j^{C}\boldsymbol{\omega}_j\end{aligned} \tag{45}$$

which can be calculated analytically in advance. Tables 3 and 4 show the modified constraint Jacobian and the right hand side of constraints obtained from Eq. (45). By comparing the tables, we can find that tables 3 and 4 can be readily obtained from tables 1 and 2 by replacing $\mathbf{s}$ and $\dot{\mathbf{s}}$ with $\mathbf{p}$ and $\dot{\mathbf{p}}$ but maintaining others such as $\mathbf{a}$, $\mathbf{p}_{i,j}$ and $\dot{\mathbf{p}}_{i,j}$ as before. It can be explained by the fact that the position vector $\mathbf{s}$(a line vector) is affected by the choice of frames but the unit axis vector $\mathbf{a}$ (a free vector) or the difference vector $\mathbf{p}_{i,j}$ (a free vector) of the position vectors $\mathbf{p}_i$ and $\mathbf{p}_j$ is not.

Although the input parameters into the constraint functions are different, the initialization of constraints is the same as the original version and the initialized constraint equations are expressed as

Table 3 Modified Jacobian of basic constraints

Constraint function	Φ_{Zi}		Φ_{Zj}	
$\Phi^{d1}(\mathbf{a}_i,\mathbf{a}_j)$	0	$-\mathbf{a}_j^T\tilde{\mathbf{a}}_i$	0	$-\mathbf{a}_i^T\tilde{\mathbf{a}}_j$
$\Phi^{d2}(\mathbf{a}_i,\mathbf{p}_i,\mathbf{p}_j)$	$-\mathbf{a}_i^T$	$\mathbf{a}_i^T\tilde{\mathbf{p}}_i-\mathbf{p}_{i,j}^T\tilde{\mathbf{a}}_i$	$\mathbf{a}_i^T$	$-\mathbf{a}_i^T\tilde{\mathbf{p}}_j$
$\Phi^{s}(\mathbf{p}_i,\mathbf{p}_j)$	$-\mathbf{I}$	$\tilde{\mathbf{p}}_i$	$\mathbf{I}$	$-\tilde{\mathbf{p}}_j$
$\Phi^{ss}(\mathbf{p}_i,\mathbf{p}_j)$	$-2\mathbf{p}_{i,j}^T$	$2\mathbf{p}_{i,j}^T\tilde{\mathbf{p}}_i$	$2\mathbf{p}_{i,j}^T$	$-2\mathbf{p}_{i,j}^T\tilde{\mathbf{p}}_j$

Table 4 Modified right hand side of basic constraints

Constraint function	$\hat{\gamma}$
$\gamma^{d1}(\mathbf{a}_i,\mathbf{a}_j)$	$-\mathbf{a}_j^T\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i-2(\tilde{\boldsymbol{\omega}}_j\mathbf{a}_j)^T(\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i)-\mathbf{a}_i^T\tilde{\boldsymbol{\omega}}_j\tilde{\boldsymbol{\omega}}_j\mathbf{a}_j$
$\gamma^{d2}(\mathbf{a}_i,\mathbf{p}_i,\mathbf{p}_j)$	$-\mathbf{p}_{i,j}^T\tilde{\boldsymbol{\omega}}_i\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i-2\dot{\mathbf{p}}_{i,j}^T\tilde{\boldsymbol{\omega}}_i\mathbf{a}_i-\mathbf{a}_i^T(\tilde{\boldsymbol{\omega}}_j\dot{\mathbf{p}}_j-\tilde{\boldsymbol{\omega}}_i\dot{\mathbf{p}}_i)$
$\gamma^{s}(\mathbf{p}_i,\mathbf{p}_j)$	$-\tilde{\boldsymbol{\omega}}_j\dot{\mathbf{p}}_j-\tilde{\boldsymbol{\omega}}_i\dot{\mathbf{p}}_i$
$\gamma^{ss}(\mathbf{p}_i,\mathbf{p}_j)$	$-2\mathbf{p}_{i,j}^T(\tilde{\omega}_j\dot{\mathbf{p}}_j-\tilde{\boldsymbol{\omega}}_i\dot{\mathbf{p}}_i)-2\dot{\mathbf{p}}_{i,j}^T\dot{\mathbf{p}}_{i,j}$

$$\sum\Phi_{Zj}\dot{\hat{\mathbf{Y}}}_j=\hat{\gamma} \tag{46}$$

Eqs. (39) ~ (43) can be reused but the calculation of $\Phi_{Zj,1}$ in Eq. (39) is unnecessary at this time because $\hat{\mathbf{B}}_{j,1}$ is the identity matrix.

3. Results

However carefully implemented element by element the traditional version is, it requires at least 36+6**n** more scalar multiplications for each body than the new version where **n** is the degree of freedom of the associated joint.

Furthermore, for the closed-loop systems, modified constraint Jacobian matrix and right hand side have the same form as the traditional version but the transformation of the constraint Jacobian matrix related with $\mathbf{B}_{j,1}$ becomes unnecessary in the new version. Thus we can verify that the new version can save 6**m** scalar multiplications for each body where **m** is the number of constraints.

Fig. 3 shows the suspension system of a car that consists of 6 bodies (including a fixed chassis) and 1 spring-damper. This system is modeled with 10 joint coordinates by cutting 3 spherical

Table 5 Elapsed CPU time for dynamic analysis of a chassis fixed quarter car model

	Centroid	Instant global origin
CPU Time	0.50msec/frame	0.43msec/frame
Ratio	100%	86%

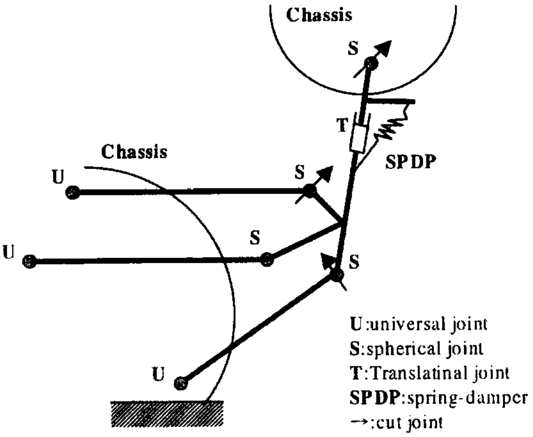

Fig. 3 Chassis-fixed quarter car model.

joints ($\mathbf{m}=9$). For this example, 108 scalar multiplications are saved for each body with spherical joint ($\mathbf{n}=3$, $\mathbf{m}=9$). Elapsed time is checked by simulating ten-thousand cycles under the gravity field. Table 5 shows the elapsed time per cycle on an Intel Pentium CPU. From the result, we can find that the fully recursive formulation can save 14% CPU time of the original version for a chassis-fixed quarter car model without much modification.

4. Conclusions

In the traditional fully recursive algorithm, the velocity transformation matrices transform the velocity in the joint coordinates into the Cartesian velocity at the centroid. On the other hand, in this study, we improved the fully recursive algorithm by using the alternative velocity transformation matrices transforming into the Cartesian velocity at a point fixed on the body but identical with the global origin at the instant of calculation. Except that the mass matrix, force vector, Jacobian matrix etc. are initialized differently, all of the remaining steps are exactly the same as the traditional version. But from this minor change, the fully recursive formulation can save at least $36+6\mathbf{n}+6\mathbf{m}$ scalar multiplications per body where $\mathbf{n}$, $\mathbf{m}$ are the joint degrees of freedom and the dimension of related constraints. Simulating on an Intel Pentium CPU, the new version achicved 14% saving of the CPU time for a chassis-fixed quarter car model.

In addition to the centroid and the instant global origin, we tested joint point as the analysis reference point and it turned out that this point has the middle efficiency.

In conclusion, the new version based on the instant global origin as the analysis reference point is recommended because it always achieves better efficiency compared with the traditional version based on the centroid. If this new version is encapsulated in the subsystem-oriented module with other algorithms, it is expected that large systems can be analyzed in the fastest way by selecting the best algorithm without any computational overheads associated with the switching of analysis algorithms.

References

Amstrong, W. W., 1979, "Recursive Solution to the Equations of Motion of an N-Link Manipulator," *Proc. 5th World Congress on Theory of Machines and Mechanisms*, Vol. 2, pp. 1343~1346.

Bae, D. S. and Haug, E. J., 1987-88, "A Recursive Formulation for Constrained Mechanical System Dynamics : Part II. Closed Loop System," *Mech. Struct. & Mach.*, Vol. 15, No. 4, pp. 481~506.

Featherstone, R., 1983, "The Calculation of Robot Dynamics Using Articulated Body Inertias," *Int. J. Robotics Res.*, Vol. 2, pp. 13~30.

Featherstone, R., 1987, *Robot Dynamics Algorithm, Kluwer, Boston*, pp. 89~105.

Haug, E. J., 1989, *Computer Aided Kinematics and Dynamics of Mechanical Systems*, Allyn & Bacon, Boston, pp. 357, 421.

Kim, S. S. and Han, J. G., 1999, "A study on Subsystem Synthesis Method for Vehicle System Dynamics," *KSME* (*in Korean*), Vol. 23, No. 3, pp. 520～534.

Luh, J. Y. S., Walker, M. W. and Paul, R. P. C., 1980, "On-line Computational Scheme for Mechanical Manipulators," *J. Dyn. Sys. Measurement Contr.*, Vol. 102, pp. 69～76.

McCarthy, J. M., 1990, An Introduction to Theoretical Kinematics, The MIT Press, Cambridge, pp. 37～41.

Negrut, D., Serban, D. and Potra, F. A., 1997, "A Topology-Based Approach to Exploiting Sparcity in Multibody Dynamics: Joint Formulation," *Mech. Strut. & Mach.*, Vol. 25, No. 2, pp. 221～241.

Tsai, F. F. and Haug, E. J., 1991a, "Real-Time Multibody System Dynamic Simulation: Part I. A Modified Recursive Formulation and Topological Analysis," *Mech. Struct. & Mach.*, Vol. 19, No. 1, pp. 99～127.

Tsai, F. F. and Haug, E. J., 1991b, "Real-Time Multibody System Dynamic Simulation: Part II. A Parallel Algorithm and Numerical Results." *Mech. Struct. & Mach.*, Vol. 19, No. 2, pp. 129～162.

Walker, M. W. and Orin, D. E., 1982, "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *J. Dyn. Sys. Measurement Contr.*, Vol. 104, pp. 205～211.