120 KSME Journal, Vol. 4, No. 2, pp. 120~127, 1990.

COMPARATIVE STUDY OF DYNAMIC ANALYSIS TECHNIQUES IN VEHICLE SIMULATION

S.H. Shin* and W.S. Yoo**

(*Received February 28, 1990*)

Several dynamic analysis techniques are compared with a planar vehicle model. Cartesian formulation, suspension superelement technique, velocity transformation, and recursive formula are compared. The relation between the recursive formula and velocity transformation is investigated. When those techniques are applied to a planar vehicle with two independent suspensions, the efficiency of those methods was compared. The computational efficiency of the recursive formula was higher than those of other methods for the specific planar vehicle example.

Key Words: Recursive Formula, Velocity Transformation, Suspension Superelement

1. INTRODUCTION

In response to the needs of the aerospace industry in the mid-1960s, Hooker and Margulies(Hooker, Margulies, 1965) analyzed a spacecraft as an open-chain linkaged rigid body system with revolute Joints. Since then, many other methods have been developed for the modeling of multibody systems. The development of computer-based methods for multibody dynamics has proceeded simultaneously in three fields: spacecraft dynamics(Hooker, Margulies, 1965; Jerkovsky, 1978 Keat, 1984), machine dynamics(McCullough, Haug, 1985; Kim, Vanderploeg, 1986; Nikravesh, 1988) and robotics (Hollerbach, 1980; Craig, 1986). Although their points of view are different, there is much in common among these three areas.

The differences among the existing methods for the dynamic analysis of multibody systems lie in the methods formulating the governing equations of motion, and in the types of coordinates being employed. In the field of mechanical systems, however, the use of Lagrange's equations predominates. So, the differences mainly come from the selection of coordinates.

Equation formulations using absolute(Cartesian)coordinates are simpler and more general than relative ones, but requires longer simulation times due to the large number of coordinates. It is also difficult to apply the control problem. Relative coordinate formulations need less simulation time, but have some restricutions in the treatment of constraint equations. And they do not directly determine user-oriented data, so it may be difficult to interpret the output directly. As a compromise, some papers use velocity transformation to increase simulation time without loss of generality (Keat, 1984; Kim, Vanderploeg, 1986).

*Department of Mechanical Engineering, Pusan National University, Pusan 609-735, Korea
**Department of Production & Mechanical Engineering, Pusan National University, Pusan 609-735, Korea.

Since the desired motions of an end-effector in roborts are often controlled by a set of joint torques, it is more conveniet to derive the euqations in terms of joint variables. In order to execute this process with real-time control, recursive formulas are often used(Hollerbach, 1980; Craig, 1986). On the other hand, the absolute coordinate system is often used to analyze moving basebody machinery and vehicles. Using Cartesian coordinates, for instance, it is easy to calcualte the tire reaction force from the condition of the road surface and the orientation of a vehicle in motion.

In this paper, several dynamic analysis techniques are compared. The relation between the recursive formula and velocity transformation is also investigated. Computer simulations are executed with a planar vehicle model.

2. SOME DYNAMIC ANALYSIS TECHNIQUES

There are several techniques for the dynamic analysis of multibody systems. But the comparison of those methods is difficult because of their dependence on the given system. In this paper, those techniques are compared with a planar vehicle shown in Fig. 1, which is composed of a chassis and two independent suspensions. If the wheel assemblies are modeled as separate bodies, their motions are confined to the direction which is perpendicular to the chassis. So, the degrees of freedom of the system shown in Fig. 1 is 5.

In this section, three methods-the Cartesian coordinate approach, the suspension superelement technique, and velocity transformation-are compared with the system shown in Fig. 1.

2.1 Cartesian Coordinate Approach (Wehage, Haug, 1981; Nikravesh, Chung, 1982)

(1) Number of Coordinates

For a free body in a plane, three coordinates are necessary to specify its motion. To describe the configuration of the system shown in Figs. 1, 9 coordinates are essential

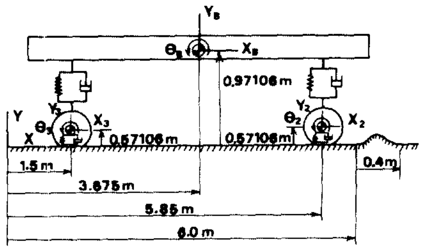


Fig. 1 Definition of coordinate system with cartesian coordinate

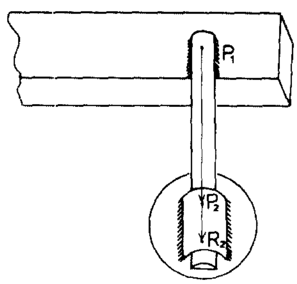


Fig. 2 Translational joint

$$Q=[x_B,y_B,\theta_B,x_2,y_2,\theta_2,x_3,y_3,\theta_3]^T \tag{1}$$

These 9 coordinates are dependent on each other through 4 constraint equations.

(2) Constraint Formulation

The kinematic joints in this model can be described as algebraic constraint equations. A translational joint between the chassis and the suspension is shown in Fig. 2. A translational joint allows relative translation of a pair of bodies along a common axis, but no relative rotation between the bodies. For the translational joint shown in Fig. 2, two constraint equations can be formulated by defining three points on the line of translation(Nikravesh, 1988). For the vehicle system of Figs. 1, 4 kinemtic constraint euqations can be formulated with 2 translational joints.

(3) Equations of Motion

The euqations of motion of a system with kinematic constraints can be derived by the Lagrange euqation with the Largrange multiplier technique. The complete equations of motion for a kinematically constrained mechanical system can be written as the mixed differential-algebraic euqation (Nikravesh, 1988).

$$\begin{bmatrix} M & \Phi_Q^T \\ \Phi_Q & 0 \end{bmatrix}\begin{Bmatrix} \ddot{Q} \\ -\lambda \end{Bmatrix}=\begin{Bmatrix} f \\ \gamma \end{Bmatrix} \tag{2}$$

subject to $\Phi(Q)=0$ and $\Phi_Q\dot{Q}=0$.
where $M=\text{diag}[M_B,M_2,M_3]$, $M_i=\text{diag}[m_i,m_i,J_i]$, f is the Cartesian force vector, and $\gamma=-(\Phi_Q\dot{Q})_Q\dot{Q}$. And Φ_Q is the Jacobian matrix of the constraint equation Φ and Cartesian coordinates vector Q. For the system of Fig. 1, the matrix size of the Eq.(2) is 13×13.

Since the constraint equations are highly nonlinear, Newton-Raphson or other methods are essential to solve kinematic constraint euqations.

Moreover, the constraint equations make the size of the equations of motion greater, and the simulation time becomes longer due to increments in the matrix size. Thus, a formulation without kinematic constraint equations is preferred for program efficiency.

2.2 Suspension Superelement Technique Approach (McCullough, Haug, 1985, ; Jung, Yoo, 1988)

A suspension superelement is a suspension subsystem that occurs in vehicle modeling often repeatedly in the same vehicle model. In this case, the euqations of motion of the system can be easily formulated using the equations of motion with one suspension superelement.

(1) Number of Coordinates

When a translational joint is defined between the chassis and the suspension subsystem, the configuration of the subsystem can be defined by a state variable d. In Fig. 3, a relative coordinate d_2 is defined to specify the location of the front wheel with respect to the local X_BY_B frame attached to the basebody(chassis). Defining another relative coordinate d_3 for the rear wheel, the vector of coordinates for the vehicle shown in Fig. 1 becomes.

$$q=[x_B,y_B,\theta_B,d_2,d_3]^T \tag{3}$$

(2) Equations of Motion

In Fig. 3, the location of the front wheel can be written as

$$\begin{aligned} r_2 &= r_B+A^Br_{B2} \\ &= r_B+A(^BS_{B2}+^BU_2d_2) \end{aligned} \tag{4}$$

where r_B is the vector from the XY frame to the origin of the X_BY_B frame, $^BS_{B2}$ is the location of the suspension attachment point at the chassis with respect to the X_B Y_B frame, $^BU_2$ is a unit vector representing the direction of the relative motion with respective to the X_B Y_B frame, $^Br_{B2}$ is the vector of the front wheel location defined in the X_B Y_B frame, and A is the transformation matrix from the X_B Y_B fram to the XY frame. The transformation matrix A can be written as

$$A=\begin{bmatrix} C\theta_B & -S\theta_B \\ S\theta_B & C\theta_B \end{bmatrix} \tag{5}$$

where θ_B is the angle between the X_B Y_B frame and the XY frame.

The velocity of the wheel can be obtained as

$$\dot{r}_2=\dot{r}_B+B^Br_{B2}\dot{\theta}_B+A^BU_2\dot{d}_2 \tag{6}$$

where the B matrix is the derivative of the transformation matrix A with respect to θ_B. The kinetic energy of the vehicle system shown in Fig. 1 is

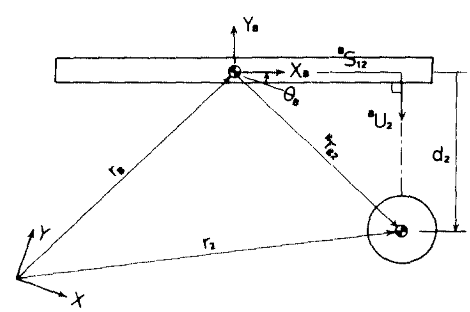


Fig. 3 Definition of coordinate system with relative coordinate

$$T=1/2[m_c\ \dot{r}_B^T\dot{r}_B+J_c\dot{\theta}_B^2+\sum_{i=2}^{3}(m_{s_i}\dot{r}_i^T\dot{r}_i+J_{s_i}\dot{\theta}_B^2)] \tag{7}$$

where m_c and J_c are the mass and moments of inertia of the chassis, and m_{s_i} and J_{s_i} are the mass and moments of inertia of the i-th suspension. Lagrange equations of motion can be used to derive the equations of motion using the kinetic energy expression of Eq. (7). The detailed derivation can be found in(Jung, Yoo, 1988).

The mechanical system superelement concept is used to take advantage of the efficiency of relative coordinates. However, it is only efficient in the case of the recurrence of the same kinematic structure in a system. If there are different suspensions in the vehicle system, the derivation of the equation of motion becomes complicated and tedious.

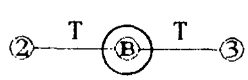
2.3 Velocity Transformation Approach (Shin, S.H., Kim, J.Y. and Yoo, W.S., 1989)

Using the transformation operater formation by Jerkovsky (Jerkovsky, 1978) and topological tree analysis(Wittenburg, 1977) in spacecrft dynamics, Kim(Kim, Vanderploeg, 1986) developed a very efficient method. He used the essence of both of the coordinate systems, the generality of the Cartesian coordinate system, and the computational efficiency of the relative coordinate system.

(1) Topological System Analysis

Graph theory is an effective method of identifying the topological structure of large scale multibody dynamic systems. The graphical representation of the vehicle model shown in Fig. 1 and its path matrix are as follows(Wittenburg, 1977)

$$\pi=\begin{bmatrix}1&0&0\\1&1&0\\1&0&1\end{bmatrix} \tag{8}$$



(2) Velocity Transformation

The vector of the Cartesian velocities $\dot{Q}$ of the system can be written as

$$\dot{Q}=[\dot{x}_B,\dot{y}_B,\dot{\theta}_B,\dot{x}_2,\dot{y}_2,\dot{\theta}_2,\dot{x}_3,\dot{y}_3,\dot{\theta}_3]^T \tag{9}$$

The relation between the Cartesian velocities $\dot{Q}$ and the relative velocities $\dot{q}$ from Eq.(3) makes the velocity transformation matrix T as

$$\dot{Q}=T\dot{q} \tag{10}$$

The time derivative Eq.(10) yields an acceleration transformation equation as

$$\ddot{Q}=T(q)\ddot{q}+\dot{T}(q,\dot{q})\dot{q} \tag{11}$$

The matrices T and $\dot{T}$ for the system shown in Fig. 1 are as following ;

$$T=\left[\begin{array}{cc|c|c} I_2 & 0 & 0 & 0\\ 0 & 1 & & \\ \hline I_2 & -\tilde{r}_{B2} & U_2 & 0\\ 0 & 1 & 0 & 0\\ \hline I_2 & -\tilde{r}_{B3} & 0 & U_3\\ 0 & 1 & & 0 \end{array}\right]_{9\times5}$$

$$\dot{T}=\left[\begin{array}{cc|c|c} 0 & & 0 & 0\\ \hline 0 & -\dot{\tilde{r}}_{B2} & \dot{\tilde{\theta}}_2 U_2 & \\ 0 & 0 & 0 & 0\\ \hline 0 & -\dot{\tilde{r}}_{B3} & & \dot{\tilde{\theta}}_3 U_3\\ 0 & 0 & 0 & 0 \end{array}\right]_{9\times5} \tag{12}$$

where the notation $\tilde{a}$ is defined by the cross product of two vectors a and b as $a\times b=\tilde{a}b=-\tilde{b}a$. U_2 is the unit vector representing the direction of the relative motion with respect to the XY frame, and r_{B2} and r_{B3} are the vectors of the front wheel and rear wheel in the XY frame, respectively. Using the velocity transformation matrix T of Eq.(12) and the velocity $\dot{q}$, the linear and angular velocities of body 2, becomes,

$$\begin{Bmatrix}\dot{X}_2\\ \dot{Y}_2\\ \dot{\theta}_2\end{Bmatrix}=\begin{bmatrix} I_2 & -r_{B2} & U_2 & \\ & & & 0\\ 0 & 1 & 0 & \end{bmatrix}\begin{Bmatrix}\dot{X}_B\\ \dot{Y}_B\\ \dot{\theta}_B\\ \dot{d}_2\\ \dot{d}_3\end{Bmatrix}$$

$$=\begin{Bmatrix}\begin{pmatrix}\dot{X}_B\\ \dot{Y}_B\end{pmatrix}+\dot{\theta}_B\times r_{B2}\\ \dot{\theta}_B\end{Bmatrix}+\begin{Bmatrix}U_2\dot{d}_2\\ 0\end{Bmatrix} \tag{13}$$

(3) Equations of Motion

Using the transformation relations of Eq.(10) and Eq.(11), the equations of motion with Cartesian coordinates can be converted to those of relative coordinates as(Kim, Vanderploeg, 1986)

$$\begin{aligned} &T^TM\ddot{Q}=T^Tf\\ &T^TM(T\ddot{q}+\dot{T}\dot{q})=T^Tf\\ &(T^TMT)\ddot{q}=T^T(f-M\dot{T}\dot{q}) \end{aligned} \tag{14}$$

where $M=\mathrm{diag}[M_B,M_2,M_3]$, $M_i=\mathrm{diag}[m_i,m_i,J_i]$.

3. RECURSIVE FORMULA APPROACH (Shin, Yoo, 1988)

The recursive formula for the dynamic analysis of robot systems is well deveoped(Hollerbach, 1980 ; Craig, 1986). But most of the research is confined to a fixed basebody with a single tree structure, which is most common in robot manipulators.

Bae(Bae, Haug, 1987, 1988) derived a recursive formualtion for constrained mechanical system using variation and vector calculus. Graphic definition of the system is used to define computational sequences for parallel computation.

In this section, the recursive formula is obatined using the velocity relations of the previous links and the relative joint velocity.

Adding the force and moment relations and the basebody motion, the recursive dynamic simulation algorithm for a moving basebody and a multiple tree structure is derived.

3.1 Convention for Affixing Frames

A system in which a basebody is connected with many open loop systems, such as Fig. 4, is named a tree structure. The

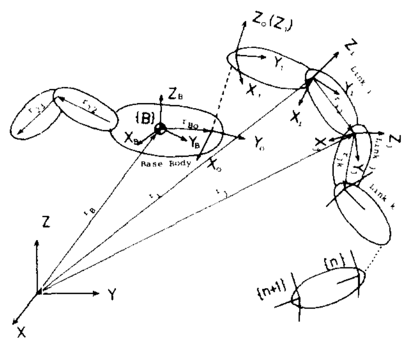


Fig. 4 Definition of joint axes

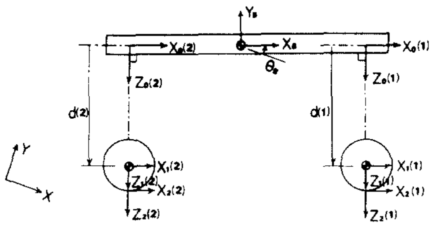


Fig. 5 Frame assignment in recursive formula

convention for affixing frames is as follows. For a basebody, it is the same as the conventional method in the Cartesian coordinate formulation. But for a tree, the following notations are employed :

(1) The frame$\{i\}$is attached to the link i on joint i, where the Z_i-axis coincides with the joint axis i

(2) $d_i(\phi_i)$is the distance(angle)from X_{i-1} to X_i measured along(about) Z_i

(3) The frame $\{0\}$ is attached to the basebody and satisfies the first joint axis condition of chain(Craig, 1986)as

$$\alpha_0 = a_0 = 0$$
$$\varepsilon_1 = \begin{bmatrix} 0 : \text{revolute joint} \\ 1 : \text{translational joint} \end{bmatrix} \tag{15}$$

where α_0 and a_0 are the twist angle and the length of link 0 respectively.

(4) The orientation of the last frame$\{n+1\}$ is identical to the frame$\{n\}$. So, $n+2$ frames exist in a tree with n links. The$\{n+1\}$ frame is used to specify the orientation of the end-effector(external)force.

Fig. 5 shows the frame assignments of the vehicle system.

3.2 Velocity Relations of Links

Let

$$(q_j, \varepsilon_j) = \begin{bmatrix} (\phi_j, 0) : \text{revolute joint} \\ (d_j, 1) : \text{translational joint} \end{bmatrix}$$

where q_j is the joint variable of link j.

In Fig. 4, the relative angular velocity between link j and link i is expressed as

$$\Omega_j = (1-\varepsilon_j) U_j \dot{q}_j \tag{16}$$

which is zero for the translational joint$(\varepsilon_j = 1)$, and $U_j \dot{\phi}_j$ for the revolute joint$(\varepsilon_j = 0, q_j = \phi_j)$. The absolute angular velocity of link j,ω_j,can be expressed by adding the angular velocity of link i,ω_i,and the relative angular velocity Ω_j as

$$\omega_j = \omega_i + \Omega_j \tag{17}$$

The angular velocity of link k in Fig. 4 is

$$\begin{aligned} \omega_k &= w_j + \Omega_k \\ &= (\omega_i + \Omega_j) + \Omega_k \end{aligned} \tag{18}$$

where Ω_k is the relative angular velocity between link j and link k.

If the absolute angular velocity of a basebody is ω_B, then the absolute angular velocity of link k is

$$\omega_k = \omega_B + \sum_{j=1}^{k} \Omega_j \tag{19}$$

The absolute velocity of link j, V_j,is

$$V_j = V_i + \omega_i \times r_{ij} + \nu_j \tag{20}$$

where $\nu_j = \varepsilon_j U_j \dot{q}_j$, which is zero for the revolute joint$(\varepsilon_j = 0)$ and $U_j\ \dot{d}_j$ for the translational joint$(\varepsilon_j = 1, q_j = d_j)$. And r_{ij} is the distance between r_j and r_i as shown in Fig. 4. The absolute velocity of link k, V_k,is

$$\begin{aligned} V_k &= V_j + \omega_j \times r_{jk} + \nu_k \\ &= (V_i + \omega_i \times r_{ij} + \nu_j) + (\omega_i + \Omega_j) \times r_{jk} + \nu_k \\ &= V_i + \omega_i \times (r_{ij} + r_{jk}) + \Omega_j \times r_{jk} + \nu_j + \nu_k \\ &= V_i + \omega_i \times r_{ik} + \Omega_j \times r_{jk} + (\nu_j + \nu_k) \end{aligned} \tag{21}$$

If the absolute velocity of a basebody is V_B, then the absolute velocity of link k is

$$V_k = V_B + \omega_B \times r_{Bk} + \sum_{j=1}^{k} (\Omega_j \times r_{jk} + \nu_j) \tag{22}$$

where r_{Bk} is the distance vector from the mass center of the basebody to the origin of the frame$\{k\}$

Equations(19) and (22) can be rewritten together as

$$\begin{aligned} \begin{Bmatrix} V_k \\ \omega_k \end{Bmatrix} &= \begin{bmatrix} V_B + \omega_B \times r_{Bk} \\ \omega_B \end{bmatrix} + \sum_{j=1}^{k} \begin{bmatrix} \Omega_j \times r_{jk} + \nu_j \\ \Omega_j \end{bmatrix} \\ &= \begin{bmatrix} V_B + \omega_B \times r_{Bk} \\ \omega_B \end{bmatrix} + \sum_{j=1}^{k} \begin{bmatrix} (1-\varepsilon_j) U_j \times r_{jk} + \varepsilon_j U_k \\ (1-\varepsilon_j) U_j \end{bmatrix} \dot{q}_j \end{aligned} \tag{23}$$

When Eq.(23) is applied to the system with the translational joint shown in Fig. 1, V_2 and ω_2 become

$$\begin{Bmatrix} V_2 \\ \omega_2 \end{Bmatrix} = \begin{bmatrix} V_B + \omega_B \times r_{B2} \\ \omega_B \end{bmatrix} + \begin{bmatrix} U_2 \\ 0 \end{bmatrix} \dot{d}_2 \tag{24}$$

Equation(24) is the same expression as Eq.(13)from velocity transformation. So, Eq.(23) can be simply represented as

$$\dot{Q}_k = T_{kB} + \sum_{j=1}^{k} T_{jk} \dot{q}_j \tag{25}$$

where T_{jk} and T_{kB} are the $(j+1, k)$-th and the $(k,1)$-th elements of the transformation matrix T in Eq.(12), respectively.

3.3 Acceleration Relation of Links

Differentiating Eq.(25), the linear and angular acceleration of link k can be obtained.

$$\ddot{Q}_k = \dot{T}_{kB} + \sum_{j=1}^{k}(T_{jk}\ddot{q}_j + \dot{T}_{jk}\dot{q}_j) \tag{26}$$

It can be rewritten as

$$\begin{aligned}\dot{V}_k &= \dot{V}_B + \dot{\omega}_B \times r_{Bk} + \omega_B \times \dot{r}_{Bk} \\ &+ \sum_{j=1}^{k}[\{\omega_j \times (1-\varepsilon_j)U_j\} \times r_{jk} + (1-\varepsilon_j)U_j \times \dot{r}_{jk} \\ &+ \omega_j \times \varepsilon_j U_j]\dot{q}_j + \sum_{j=1}^{k}\{(1-\varepsilon_j)U_j \times r_{jk} + \varepsilon_j U_j\}\ddot{q}_j \\ \dot{\omega}_k &= \dot{\omega}_B + \sum_{j=1}^{k}\{\omega_j \times (1-\varepsilon_j)U_j\}\dot{q}_j + \sum_{j=1}^{k}(1-\varepsilon_j)U_j\ddot{q}_j\end{aligned} \tag{27}$$

where $\dot{r}_{jk} = V_k - V_j$.

3.4 Recursive Formula

Inserting $k+1$ instead of k in Eq.(19) and (22), the recursive formula can be derived. For the angular velocity of link $k+1$, the recursive formula is

$$\begin{aligned}\omega_{k+1} &= \omega_B + \sum_{j=1}^{k+1}\Omega_j \\ &= (\omega_B + \sum_{j=1}^{k}\Omega_j) + \Omega_{k+1} \\ &= \omega_k + \Omega_{k+1}\end{aligned} \tag{28}$$

The recursive formula for the linear velocity becomes

$$\begin{aligned}V_{k+1} &= V_B + \omega_B \times r_{Bk+1} + \sum_{j=1}^{k+1}(\Omega_j \times r_{jk+1} + \nu_j) \\ &= V_B + \omega_B \times (r_{Bk} + r_{kk+1}) + \sum_{j=1}^{k}[\Omega_j \times (r_{jk} + r_{kk+1}) \\ &+ \nu_j] + \Omega_{k+1} \times r_{k+1k+1} + \nu_{k+1} \\ &= [V_B + \omega_B \times r_{Bk} + \sum_{j=1}^{k}(\Omega_j \times r_{jk} + \nu_j)] \\ &+ (\omega_B + \sum_{j=1}^{n}\Omega_j) \times r_{kk+1} + \nu_{k+1} \\ &= V_k + \omega_k \times r_{kk+1} + \nu_{kk+1}\end{aligned} \tag{29}$$

The recursive formula for the linear and angular acceleration can be obtained in the same way (Shin, Yoo, 1988) and the results are

$$\begin{aligned}\dot{V}_{k+1} &= \dot{V}_k + \dot{\omega}_k \times r_{kk+1} + \omega_k \times \dot{r}_{kk+1} + \omega_{k+1} \\ &\times \varepsilon_{k+1}U_{k+1}\dot{q}_{k+1} + \varepsilon_{k+1}U_{k+1}\ddot{q}_{k+1} \\ \dot{\omega}_{k+1} &= \dot{\omega}_k + (1-\varepsilon_{k+1})U_{k+1}\ddot{q}_{k+1} + \omega_{k+1} \\ &\times (1-\varepsilon_{k+1})U_{kk+1} + \dot{q}_{k+1}\end{aligned} \tag{30}$$

3.5 Inverse Dynamics in Robotics

Figure 6 shows the forces acting on a link. If the point of mass center of link k is C_k, the acceleration of the mass center is represented as

$$\dot{V}_{C_k} = \dot{V}_k + \dot{\omega}_k \times r_{kC_k} + \omega_k \times \dot{r}_{kC_k} \tag{31}$$

where r_{kC_k} is the distance vector from the origin of frame$\{k\}$ to the mass center of link k, and $\dot{r}_{kC_k} = V_{C_k} - V_k$. The resultant force and moment acting on link k are

$$\begin{aligned}F_k &= m_k\dot{V}_{C_k} \\ &= f_k - f_{k+1} + m_k\bar{g} \\ N_k &= I_{C_k}\dot{\omega}_k + \omega_k \times I_{C_k}\omega_k\end{aligned} \tag{32}$$

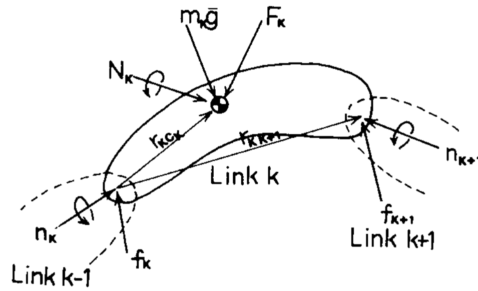


Fig. 6 Force equilibrium of a link

$$\begin{aligned}&= n_k - n_{k+1} - r_{kC_k} \times f_k + (r_{kk+1} - r_{kC_k}) \times (-f_{k+1}) \\ &= n_k - n_{k+1} + r_{kC_k} \times (f_{k+1} - f_k) - r_{kk+1} \times f_{k+1}\end{aligned}$$

where f_k and n_k are the force and moment exerted on link k by link $k-1$.

Summarizing the abvove equations, the recursive inverse dynamics formula becomes

For the k-th tree having n links

(1) Outward Iteration

$$\begin{aligned}\omega_{k+1} &= \omega_k + (1-\varepsilon_{k+1})U_{k+1}\dot{q}_{k+1} \\ V_{k+1} &= V_k + \omega_{k+1} \times r_{kk+1} + \varepsilon_{k+1}U_{k+1}\dot{q}_{k+1} \\ \dot{\omega}_{k+1} &= \dot{\omega}_k + (1-\varepsilon_{k+1})U_{k+1}\ddot{q}_{k+1} + \omega_{k+1} \\ &\times (1-\varepsilon_{k+1})U_{k+1}\dot{q}_{k+1} \\ \dot{V}_{k+1} &= \dot{V}_k + \dot{\omega}_k \times r_{kk+1} + \omega_k \times \dot{r}_{kk+1} + \omega_{k+1} \\ &\times \varepsilon_{k+1}U_{k+1}\dot{q}_{k+1} + \varepsilon_{k+1}U_{k+1}\ddot{q}_{k+1} \\ \dot{V}_{C_{k+1}} &= \dot{V}_{k+1} + \dot{\omega}_{k+1} \times r_{k+1C_{k+1}} + \omega_{k+1} \times \dot{r}_{k+1C_{k+1}} \\ F_{k+1} &= m_{k+1}\dot{V}_{C_{k+1}} \\ N_{k+1} &= I_{C_{k+1}}\dot{\omega}_{k+1} + \omega_{k+1} \times I_{C_{k+1}}\omega_{k+1}\end{aligned} \tag{33}$$

(2) Inward Iteration

$$\begin{aligned}f_k &= f_{k+1} + F_k - m_k\bar{g} \\ n_k &= n_{k+1} + N_{k+1} + r_{kC_k} \times (f_k - f_{k+1}) + r_{kk+1} \times f_{k+1} \\ \tau_k &= n_k^T U_k\end{aligned} \tag{34}$$

The derived recursive formula are very similar to the equations in robotics (Craig, 1986), but they are written with absolute components instead of local components.

3.6 Dynamics in Mechanical Systems

Equations (33) and (34) are very similar to the equations in robotics (Craig, 1986), but these equations are applicable only to a single tree. If the system has multiple trees, the algorithm must be modified.

Figure 7 shows a basebody connected to m trees. $r_{B0}(k)$ vector is the distance vector from frame$\{B\}$ to frame$\{0\}$ of the k-th tree. In Fig. 7, f_B and n_B are the force and moment acting on the mass center of the basebody, respectively. The resultant force and moment acting on the basebody are

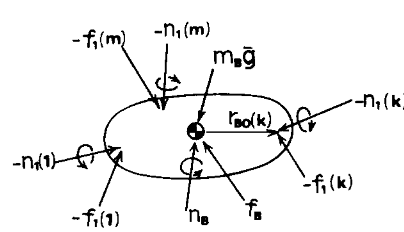


Fig. 7 F.B.D. of a Basebody

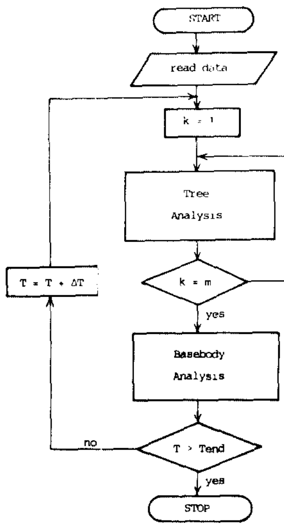


Fig. 8 Simulation algorithm of recursive dynamic analysis

$$F_B = f_B - \sum_{k=1}^{m}[f_1(k)] + m_B\vec{g}$$
$$N_B = n_B - \sum_{k=1}^{m}[n_1(k) + r_{B0}(k)\times f_1(k)] \qquad (35)$$

When the number of links in a tree without the basebody is n, the simulation algorithm for recursive dynamic analysis is obtained from tree analysis and basebody analysis. The flow chart of the recursive dynamic algorithm is shown in Fig. 8. The detailed algorithm for the tree analysis and the basebody analysis in the flow chart are as following;

(1) Tree Analysis

Step 1: compute the orientations of
frame{0} $w.r.t$ frame{B} : ${}^B_0R(k)$
frame{i} $w.r.t$ frame{$i-1$} : $\{{}^{i-1}_{i}R(k)\}_{i=1}^{n+1}$
frame{i} $w.r.t$ the absolute frame: $\{W_i(k)\}_{i=1}^{n+1}$
joint axis unit vector $w.r.t$ the absolute frame: $\{U_i(k)\}_{i=1}^{n}$

Step 2: compute distance vectors
from frame{B} to frame{0} : $r_{B0}(k)$
from frame{i} to frame{$i+1$} : $\{r_{ii+1}(k)\}_{i=0}^{n}$
from frame{i} to the mass center C_i : $\{r_{iC_i}(k)\}_{i=1}^{n}$

Step 3: compute angular velocities of links: $\{\omega_i(k)\}_{i=1}^{n}$

Step 4: compute linear velocities of links: $\{V_i(k)\}_{i=1}^{n}$

Step 5: from model, compute
joint force: $\{f_i(k)\}_{i=1}^{n+1}$
joint torque: $\{n_i(k)\}_{i=1}^{n+1}$

Step 6: compute
the resultant force: $\{F_i(k)\}_{i=1}^{n}$
the resultant moment: $\{N_i(k)\}_{i=1}^{n}$

Step 7: compute
the linear acceleration of the mass center: $\{\dot{V}_{C_i}(k)\}_{i=1}^{n}$
the angular acceleration: $\{\dot{\omega}_i(k)\}_{i=1}^{n}$

Step 8: compute the linear accelerations of joints: $\{\dot{V}_i(k)\}_{i=1}^{n}$

Step 9: compute the accelerations of the joint variables: $\{\ddot{q}_i(k)\}_{i=1}^{n}$
for revolute joint ($\varepsilon=0$),
$\ddot{\phi}_{i+1} = {}^{i+1}W[\dot{\omega}_{i+1} - \dot{\omega}_i - \omega_{i+1}\times U_{i+1}\dot{\phi}_{i+1}]\cdot{}^{i+1}Z_{i+1}$
for translational joint ($\varepsilon=1$),
$\ddot{d}_{i+1} = {}^{i+1}W[\dot{V}_{i+1} - \dot{V}_i - \dot{\omega}_i\times r_{ii+1} - \omega_i\times\omega_i\times r_{ii+1})$
$\quad -2\omega_i\times U_{i+1}\dot{d}_{i+1}]\cdot{}^{i+1}Z_{i+1}$

Step 10: integrate acceleration joint coordinate: $\{\ddot{q}_i(k)\}_{i=1}^{n}$

(2) Basebody Analysis

Step 1: compute the resulant force F_B and the moment N_B for the basebody

Step 2: compute the linear acceleration $\dot{V}_B$ and the angular acceleration $\dot{\omega}_B$ for the basebody

Step 3: integrate accelerations of the basebody: $\dot{V}_B, \dot{\omega}_B$

where ${}^{i-1}_{i}R$ is the 3×3 rotation matrix of frame{i} relative to the frame{$i-1$} defined with Denavit-Hartenberg notation (Craig, 1986). When the orientation of the basebody with respect to the XY frame is W_B, then $W_i = W_B{}^B_0R{}^0_iR$.

4. COMPUTER SIMULATION

In order to analyze the transient response of the vehicle shown in Fig. 1, the following data are used in the simulation (Jung, Yoo, 1988)

Chassis
mass m_C : 1427.25kg
moment of inertia J_C : 6917.6kg · m^2

Wheel
mass m_S : 118.5kg
moment of inertia J_S : 1.333kg · m^2

Suspension spring
spring constant: 2.764×10^5N/m (3 times when spring deformation $|\Delta\ell| > 0.15$m)
damping coefficient: 1549.8N.sec/m (rebound)
4971.0N.sec/m (compression)

Tire
radius: 0.6m
spring constant: 2.82×10^5N/m

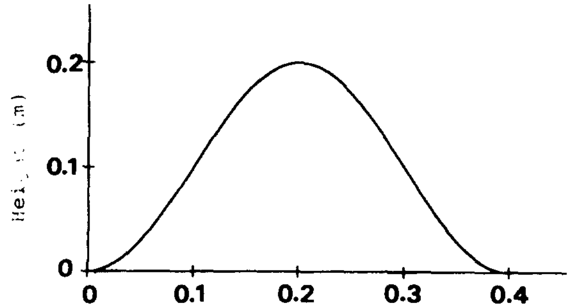


Fig. 9 Single bump profile

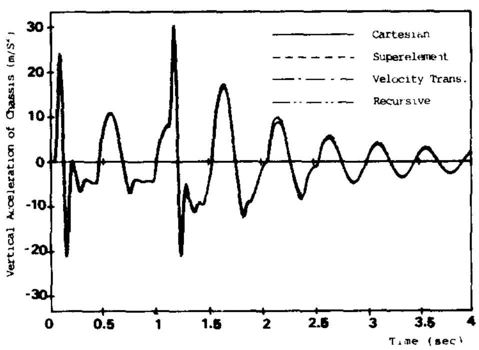


Fig. 10 Vertical acceleration of chassis

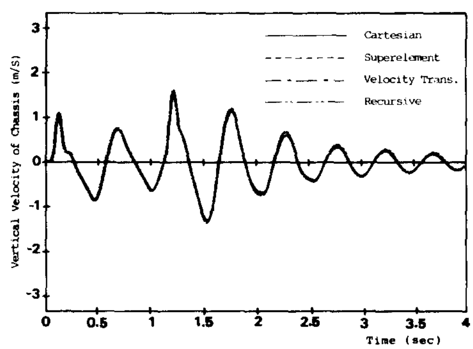


Fig. 11 Vertical velocity of chassis

damping coefficient : 925N. sec/m

For the tire reaction force, a linear force-deformation relation is used. The tire reaction force F_{tire} is assumed

$$F_{tire} = K_{tire} \times (\text{pen}) + C_{tire} \times (\dot{\text{pen}})$$

where K_{tire}, C_{tire}, and pen are spring constant, damping coefficient, and penetration of the tire, respectively. The datailed information about the vehicle model is referred to reference(Jung, Yoo, 1988). The vehicle travels over a single bump shown in Fig. 9 with a constant speed of 4m/sec. Since the front wheel and rear wheel completely go over the bump after 1.5 seconds, the simulation was done unitl 4 seconds to see the complete response.

Figures 10,11,12 show the results of computer simulations for the vehicle model shown in Fig. 1. The results of the four methods are almost identical, but the simulation times for the dynamic analysis technique are quite different, as shown in

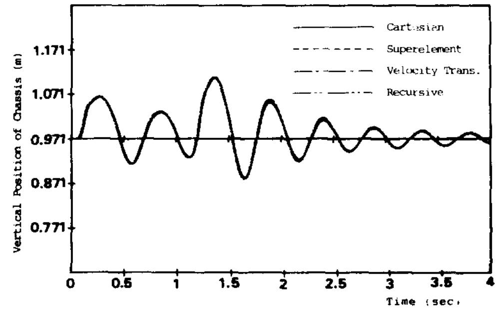


Fig. 12 Vertical position of chassis

Table 1 Simulation time(seconds)

Techniques \ Computer	PC/AT	Cyber 180/830
Cartesian formulation	1944.90	563.662
Velocity transformation	1171.89	335.241
Suspension superelement	593.79	145.036
Recursive formula	163.12	28.658

Table 1.

Using Runge-Kutta 4th order method with 0.001 integration time step, the programs are simulated for 4 seconds both on the Cyber 180/830 and on the PC/AT, which has an 80287 math coprocessor and 20MB of hard-disk memory. From the results of Table 1, following points are made.

(1) The recursive formula is the most efficient method for the vehicle model shown in Fig, 1. The recursive formula is well suited to this type of open loop system. The frame assignment of the recursive formula is somewhat complicated, however.

(2) The Cartesian coordinate formulation takes the longest simulation time because of constraint equations and the large matrix size of the differental-algebraic equation. In order to control constraint violations, a constraint stabilization method(Nikravesh, 1988) is used.

(3) The superelement technique is an efficient method, but it requires a tedious derivation for the equations of motion.

(4) The simulation time of the velocity transformation technique is about one half of the Cartesian coordinates formulation, but it still takes longer than other methods. It may be due to T and T matrix multiplication during the simulation process.

5. CONCLUSIONS

In this paper, a comparative study is made of computer oriented dynamic analysis techniques in mechanical systems, and a recursive dynamic simulation algorithm is derived. From the results of computer simulations using four techniques, the following conclusions are obtained :

(1) Using the velocity transformation matrix, which is often used in mechanical dynamics, a recursive inverse dynamics formula which is often used in robotics is derived.

(2) Using derived inverse dynamics, a recursive dynamic simulation algorithm is obtained. The efficiency of the derived recursive formula is higher than that of other methods for open loop vehicles.

ACKNOWLEDGEMENT

The authors would like to express their appreciation to the Korea Science and Engineering Foundation for Supporting this research.

REFERENCES

Bae, D.S. and Haug, E.J., 1987, "A Recursive Formulation for Constrained Mechanical System Dynamics: Part I. Open Loop Systems," Mechanics of Structures and Machines, Vol. 15(3), pp. 359~382.

Bae, D.S. and Haug, E.J., 1988, "A Recursive Formulation for Constrained Mechanical System Dynamics: Part II. Closed Loop Systems," Mechanics of Structures and Machines, Vol. 15(4), pp. 481~506.

Craig, J.J., 1986, "Introduction to Robotics," Addison-Wesley.

Hollerbach, J.M., 1980, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," IEEE Trans. on Systems, Man and Cybernatics, Vol. SMC-10, No. 11, pp. 730~736.

Hooker, W.W. and Margulies, G., 1965, "The Dynamical Attitude Equations for an n-Body Satellite," J. Astronautical Sciences, Vol. 12, No. 4, pp. 123~128.

Jerkovsky, W., 1978, "The Structure of Multibody Dynamics Equations,"Journal of Guidance and Control, Vol. 1, No. 3, May-June, pp. 173~182.

Jung, C.M. and Yoo, W.S., 1988. "Dynamic Analysis of a Vehicle with Suspension Superelement Technique," Transaction of KSME, Vol. 12, No. 3 pp. 450~456.

Keat, J.E. and Turner, J.D., 1984, "Equations of Motion of Multibody Systems for Application to Large Space Structure Deployment," AIAA Paper 84-1014-CP, AIAA Dynamics Specialist Conference, Palm Springs, CA.

Kim, S.S. and Vanderploeg, M.J., 1986, "A General and Efficient Method for Dynamic Analysis of Mechancal Systems Using Velocity Transformations,"ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 108, June, pp. 176~182.

McCullough, M.K. and Haug, E.J., 1985, "Dynamics of High Mobility Track Vehicles," ASME Journal of Mechanisms, Transmissions, and Automation in Design, May.

Nikravesh, P.E. and Chung, I.S., 1982, "Application of Euler Parameters to the Dynamic Analysis of Three Dimensional Constrained Mechanical Systems," Journal of Mechanical Design, Vol. 104, pp. 785~791.

Nikravesh, P.E., 1988, "Computer-Aided Analysis of Mechanical Systems," Prentice-Hall.

Shin, S.H. and Yoo, W.S., 1988, "Dynamic Analysis of a Multibody System Using Recursive-Fomula," Transaction of KSME, Vol. 12, No. 6, 1265~1272.

Shin, S.H., Kim, J.Y. and Yoo, W.S., 1989, "Computer-Oriented Dynamic Analysis Technique in Vehicle System," Proceedings of the 1989 ASME Design Technical Conference, Montreal, CANADA, DE-Vol. 19~3, pp. 35~43.

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

Wittenburg, J., 1977, "Dynamics of systems of Rigid Bodies," B.G. Teuber, Stuttgart.