# A recursive approach for mechanical system design sensitivity analysis

Daesung Bae*

Abstract

Recursive formulas have been effective in solving the equations of motion for large scale constrained mechanical systems. However, derivation of the formulas has been limited to individual terms in the equations of motion, such as velocity, acceleration, and generalized forces. The recursive formulas are generalized in this paper. The velocity transformation method is employed to transform the equations of motion from Cartesian to the joint spaces. Computational structure of the equations of motion in the joint space is carefully examined to classify all necessary computational operations into several categories. The generalized recursive formula for each category is then developed and applied whenever such a category of computation is encountered. Since the velocity transformation method yields the equations of motion in a compact form and computational efficiency is achieved by generalized recursive formulas, the proposed method is not only easy to implement but is also efficient. A library of generalized recursive formulas is developed to implement a dynamic analysis algorithm using backward difference.

## 1. Introduction

Erdman[1] has developed a design method for special purpose mechanisms. Kinematic equations that are formulated for a specific mechanism are directly used to develop a design process. Design methods for general mechanical systems have been presented in.[2]

In designing a structural system, numerical optimization has become already a routine procedure. Design sensitivity and optimization methods have been developed for size, shape, configuration, and topology of structural systems.[3] The first- and second-order design sensitivity analyses

* Department of Mechanical Engineering
Hanyang University, Ansan, South Korea
(dabae@email.hanyang.ac.kr)

using Trefftz method have been presented by Kita.[4] In[3], the configuration design method has been successfully applied for kinematically driven systems. In contrast to structural design, there exist few general-purpose codes with design-optimization capabilities for mechanical systems. One of the major difficulties is to establish an efficient and reliable way to analyze the design sensitivity of dynamic responses due to a design change. The objective of this research is to develop an efficient and reliable method for the design sensitivity analysis of general mechanical systems.

There are two kinds of methods in developing the governing equations of design sensitivity; direct differentiation method and adjoint-variable method. In the direct differentiation method the governing equations are obtained by differentiating the equations of motion and the constraints.[2] The adjoint-variable method was developed in optimal control[5] and involves the forward numerical integration for the dynamic analysis and the backward numerical integration for the sensitivity analysis. Since the backward numerical integration may incur numerical errors, this research employs the direct differentiation method.

There are several methods in defining the generalized coordinates for dynamic analysis of mechanical systems. Cartesian coordinates have been used in several commercial codes.[6,7] The natural-coordinate method [8] uses points and unit vectors as its generalized coordinates. The orientation of a body is represented by unit vectors. Therefore, the orientation matrix is quadratic in the natural coordinates and its Jacobian is linear. To systematically formulate the equations of motion in relative coordinates[9], Wittenburg proposed the velocity-transformation method. For computational efficiency, Hooker proposed a recursive formulation for the dynamic analysis of a satellite which has a tree topology[10]. He showed that the computational cost of the formulation increases only linearly with respect to the number of bodies. Featherstone also proposed a recursive formulation to calculate the acceleration of robot arms using screw notation.[11] These ideas have been extended by many researches for multibody rigid and flexible systems in.[12-14] Recently the recursive formulation was generalized in[14] to improve both implementation and efficiency.

The first fully three-dimensional applications of the design sensitivity analysis were demonstrated by Mani.[15] The velocity-transformation method was used to derive the governing equations of design sensitivity.

Even though the formulations proposed in the previous studies were for the general mechanical systems, their applications were confined to relatively simple problems due to the complexity of the formulations. The formulation complexity problem was resolved by using a computer algebra.

Constrained mechanical systems are represented by differential equations of motion and algebraic constrained constraint equations, which are often called the overdetermined differential algebraic system (ODAS). Several solution methods have been proposed to solve the ODAS in.[16,17] In particular, the parameterization method treated the ODAS as an ordinary differential equations (ODEs) on the kinematic constraint manifolds of the system. The stability and convergence of the method were proved in.[17]

This paper employs the velocity-transformation method[9] to derive the governing equations of motion and design sensitivity. Since the virtual displacement and acceleration relationships between the Cartesian and relative coordinates are substituted simultaneously in the velocity transformation method, the governing equations will appear in a compact matrix form. Note that the matrix operations can be computed in a recursive way. Therefore, the matrix form not only makes it easy to debug and understand the computer program but also assures computational efficiency by using the generalized recursive formulas.[14]

The recursive kinematic relationships are derived, then generalized in Section 2. The governing equations of design sensitivity and their solution method are presented in Section 3. A set of generalized recursive formulas is derived and applied to evaluate the terms in the equation of motion and design sensitivity in Section 4. A numerical example is presented in Section 5. Finally, conclusions are drawn in Section 6.

## 2 Relative coordinate kinematics and recursive formulas

### 2.1 Coordinate systems and relative kinematics

Consider a pair of bodies, as shown in Fig. 1. The X-Y-Z is the inertial reference frame and primed coordinate systems are the body reference frames. Double primed coordinate systems denote the joint reference frames. The orientation of the body reference frame is denoted by **A**.

Translational and angular velocities of the $x_i'-y_i'-z_i'$ frame, the reference frame for body i, in the X-Y-Z frame are respectively defined as $\dot{\mathbf{r}}_i$ and $\omega_i$. Twist velocity in the $x_i'-y_i'-z_i'$ frame is defined as

$$\mathbf{Y}_i \equiv \begin{bmatrix} \dot{\mathbf{r}}_i' \\ \omega_i' \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i^T\dot{\mathbf{r}}_i \\ \mathbf{A}_i^T\omega_i \end{bmatrix} \tag{1}$$

Define a cross product matrix associated with vectors **a** *and* **b** *to denote the vector cross product as follows.*

$$\tilde{\mathbf{a}}\mathbf{b} = \mathbf{a} \times \mathbf{b}$$

Recursive velocity formula for a pair of contiguous bodies have been derived in[18] as

$$\mathbf{Y}_i = \mathbf{B}_{(i-1)i1}\mathbf{Y}_{(i-1)} + \mathbf{B}_{(i-1)i2}\mathbf{v}_{(i-1)i} \tag{2}$$

where $\mathbf{v}_{(i-1)i}$ denotes the relative velocity vector for joint (i-1)i and the matrices $\mathbf{B}_{(i-1)i1}$ and $\mathbf{B}_{(i-1)i2}$ are

$$\mathbf{B}_{(i-1)i1} = \begin{bmatrix} \mathbf{A}_{(i-1)i}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{(i-1)i}^T \end{bmatrix} \begin{bmatrix} \mathbf{I} - \left(\tilde{\mathbf{S}}_{(i-1)i}' + \tilde{\mathbf{d}}_{(i-1)i}' - \mathbf{A}_{(i-1)i}\tilde{\mathbf{s}}_{(i-1)i}'\mathbf{A}_{(i-1)i}^T\right) \\ \mathbf{0} \qquad\qquad \mathbf{I} \end{bmatrix}$$

$$\mathbf{B}_{(i-1)i2} = \begin{bmatrix} \mathbf{A}_{(i-1)i}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{(i-1)i}^T \end{bmatrix} \begin{bmatrix} \left(\mathbf{d}_{(i-1)i}'\right)_{\mathbf{q}(i-1)i} + \mathbf{A}_{(i-1)i}\tilde{\mathbf{s}}_{(i-1)i}'\mathbf{A}_{(i-1)i}^T\mathbf{H}_{(i-1)i}' \\ \mathbf{H}_{(i-1)i}' \end{bmatrix} \tag{3}$$

where $\mathbf{A}_{(i-1)i}^T = \mathbf{A}_{(i-1)}^T\mathbf{A}_i$, $\mathbf{S}_{(i-1)i}' = \mathbf{A}_{(i-1)}^T\mathbf{s}_{(i-1)i}$, $\mathbf{d}_{(i-1)i}' = \mathbf{A}_{(i-1)}^T\mathbf{d}_{(i-1)i}$, $\mathbf{S}_{i(i-1)}' = \mathbf{A}_i^T\mathbf{s}_{i(i-1)}$, and $(\mathbf{d}_{(i-1)i}')_{q(i-1)i} = \partial\mathbf{d}_{(i-1)i}'/\partial\mathbf{q}_{(i-1)i}$. The matrix $\mathbf{H}_{(i-1)i}'$ is determined by the axis of rotation and $\mathbf{q}_{(i-1)i}$ denotes the relative generalized coordinate vector for joint (i-1)i. The vectors $\mathbf{s}_{i(i-1)}'$ and $\mathbf{s}_{(i-1)i}'$ are defined in Fig 1. It is important to note that the

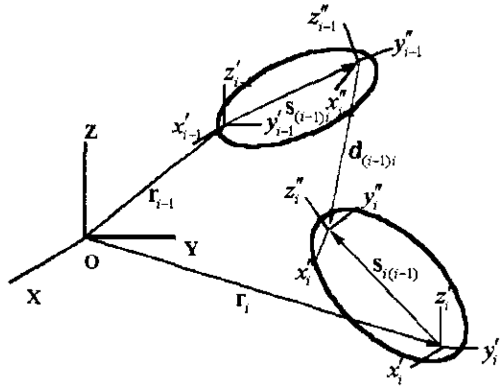


*Fig.1 Kinematic relationship between two adjacent bodies*

matrices $\mathbf{B}_{(i-1)i1}$ and $\mathbf{B}_{(i-1)i2}$ are only functions of the relative coordinates of the joint between bodies i-1 and i. As a consequence, further partial differentiations of the matrices $\mathbf{B}_{(i-1)i1}$ and $\mathbf{B}_{(i-1)i2}$ in Eq. (2) with respect to other relative generalized coordinates than $\mathbf{q}_{(i-1)i}$ becomes zero. *This property will play a key role in simplifying* the generalized recursive formulas in section 4.

### 2.2 Generalization of the velocity recursive formula

In order to generalize the velocity recursive formula, consider a mechanical system which consists of 6 bodies, as shown in Fig. 2. The system has a closed loop which is opened to form the tree structure in Fig. 3. The velocity $\mathbf{Y}_1$ for body 1 in the tree structure is obtained by replacing i by 1 in Eq. (2) as

$$\mathbf{Y}_1 = \mathbf{B}_{011}\mathbf{Y}_0 + \mathbf{B}_{012}\mathbf{v}_{01} \tag{4}$$

Similarly, the velocity $\mathbf{Y}_2$ for body 2 can be obtained as follows.

$$\mathbf{Y}_2 = \mathbf{B}_{121}\mathbf{Y}_1 + \mathbf{B}_{122}\mathbf{v}_{12} \tag{5}$$

*Substitution of Eq. (4) into Eq. (5) yields*

$$\mathbf{Y}_2 = \mathbf{B}_{121}\mathbf{B}_{011}\mathbf{Y}_0 + \mathbf{B}_{121}\mathbf{B}_{1012}\mathbf{v}_{01} + \mathbf{B}_{122}\mathbf{v}_{12} \tag{6}$$

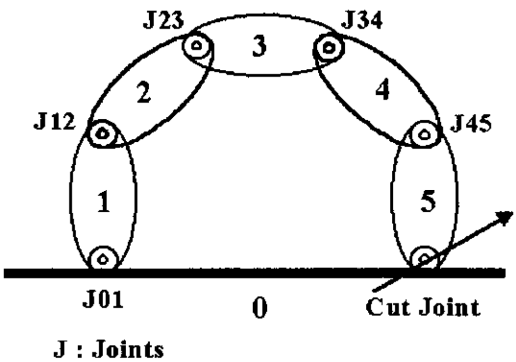


Fig. 2 A chain mechanism

If similar steps are taken for all bodies in Fig. 2, the Cartesian velocity $\mathbf{Y}_5$ is obtained as

$$\begin{aligned}\mathbf{Y}_5 = &\mathbf{B}_{451}\mathbf{B}_{341}\mathbf{B}_{231}\mathbf{B}_{121}\mathbf{B}_{011}\mathbf{Y}_0\\ &+\mathbf{B}_{451}\mathbf{B}_{341}\mathbf{B}_{231}\mathbf{B}_{121}\mathbf{B}_{012}\mathbf{v}_{01}\\ &+\mathbf{B}_{451}\mathbf{B}_{341}\mathbf{B}_{231}\mathbf{B}_{122}\mathbf{v}_{12}\\ &+\mathbf{B}_{451}\mathbf{B}_{341}\mathbf{B}_{232}\mathbf{v}_{23}\\ &+\mathbf{B}_{451}\mathbf{B}_{342}\mathbf{v}_{34}\\ &+\mathbf{B}_{452}\mathbf{v}_{45}\end{aligned} \quad (7)$$

Thus, the Cartesian velocity $\mathbf{Y}$ for all bodies is obtained in the following compact matrix form:

$$\mathbf{Y} = \mathbf{B}\mathbf{v} \quad (8)$$

where the matrix $\mathbf{B}$ is the transformation matrix between the Cartesian and relative generalized velocities and

$$\mathbf{Y} = \left[\mathbf{Y}_0^T, \mathbf{Y}_1^T, \mathbf{Y}_2^T, \mathbf{Y}_3^T, \mathbf{Y}_4^T, \mathbf{Y}_5^T\right]^T \quad (9)$$

$$\mathbf{v} = \left[\mathbf{Y}_0^T, \mathbf{Y}_{12}^T, \mathbf{Y}_{23}^T, \mathbf{Y}_{34}^T, \mathbf{Y}_{45}^T\right]^T \quad (10)$$

The Cartesian velocity $\mathbf{Y} \in \mathbf{R}^{nc}$ with a given $\mathbf{v} \in \mathbf{R}^{nr}$ ,where nc and nr respectively denote the numbers of the Cartesian and relative coordinates, can be evaluated either by using Eq. (8) obtained from symbolic substitutions or by using Eq. (2) with recursive numeric substitutions of $\mathbf{Y}_i\mathbf{s}$. Since both formulas give an identical result and recursive numeric substitution is proven to be more efficient[10], the matrix multiplication $\mathbf{Bv}$ with a given $\mathbf{v}$ will be actually evaluated by using Eq. (2). Since $\mathbf{v}$ in Eq. (8) is an arbitrary vector in $\mathbf{R}^{nr}$, Eqs. (2) and (8), which are computationally equivalent, are actually valid for any vector $\mathbf{v} \in \mathbf{R}^{nr}$ such that

$$\mathbf{X} = \mathbf{B}\mathbf{x} \quad (11)$$

and

$$\mathbf{X}_i = \mathbf{B}_{(i-1)i1}\mathbf{X}_{(i-1)} + \mathbf{B}_{(i-1)i2}\mathbf{x}_{(i-1)i} \quad (12)$$

where $\mathbf{X} \in \mathbf{R}^{nc}$ is the resulting vector of multiplication of $\mathbf{B}$ and $\mathbf{x}$.

As a result, the transformation of $\mathbf{x} \in \mathbf{R}^{nr}$ into $\mathbf{Bx} \in \mathbf{R}^{nc}$ is actually calculated by recursively applying Eq. (12) to achieve computational efficiency in this research.

### 2.3 Generalization of the force recursive formula

The generalized recursive formula for the transformation of $\mathbf{x} \in \mathbf{R}^{nr}$ into a new vector $\mathbf{Bx}$ in $\mathbf{R}^{nc}$ was derived in section 2.2. Conversely, it is often necessary to transform a vector $\mathbf{G}$ in $\mathbf{R}^{nc}$ into a new vector $\mathbf{g} = \mathbf{B}^T\mathbf{G}$ in $\mathbf{R}^{nr}$. Such a transformation can be found in the generalized force computation in the relative coordinate system with a known force in the Cartesian coordinate system. The virtual work done by a Cartesian force $\mathbf{Q} \in \mathbf{R}^{nc}$ is obtained as follows.

$$\delta W = \delta\mathbf{Z}^T\mathbf{Q} \quad (13)$$

where $\delta\mathbf{Z}$ must be kinematically admissible for all joints for a tree structure consisting of n serial bodies. Substitution of $\delta\mathbf{Z} = \mathbf{B}\delta\mathbf{q}$ into Eq. (13) yields

$$\delta W = \delta\mathbf{q}^T\mathbf{B}^T\mathbf{Q} = \delta\mathbf{q}^T\mathbf{Q}^* \quad (14)$$

where $\mathbf{Q}^* = \mathbf{B}^T\mathbf{Q}$.

The recursive formula for $\mathbf{Q}^*$ has been obtained in.[18]

$$\mathbf{Q}^*_{i(i+1)} = \mathbf{B}^T_{i(i+1)2}\left(\mathbf{Q}_{i+1} + \mathbf{S}_{i+1}\right) \qquad ,i+4,\ldots,0 \quad (15)$$

where

$$\mathbf{S}_5 = \mathbf{0}$$
$$\mathbf{S}_{i+1} = \mathbf{B}^T_{(i+1)(i+2)i}\left(\mathbf{Q}_{i+2} + \mathbf{S}_{i+2}\right) \tag{16}$$

Since $\mathbf{Q}$ is an arbitrary vector in $\mathbf{R}^{nc}$, Eqs. (15) and (16) are valid for any vector $\mathbf{Q}$ in $\mathbf{R}^{nc}$. As a result, the matrix multiplication of $\mathbf{B}^T\mathbf{G}$ is actually evaluated to achieve computational efficiency in this research by

$$\begin{aligned} \mathbf{g}_{i(i+1)} &= \mathbf{B}^T_{(i+1)2}\left(\mathbf{G}_{i+1} + \mathbf{S}_{i+1}\right) \\ \mathbf{S}_5 &= \mathbf{0} \qquad ,i = 4,\dots,0 \\ \mathbf{S}_i &= \mathbf{B}^T_{i(i+1)}\left(\mathbf{G}_{i+1} + \mathbf{S}_{i+1}\right) \end{aligned} \tag{17}$$

where $\mathbf{g} \in \mathbf{R}^{nr}$ is $\mathbf{B}^T\mathbf{G}$.

## 3 The governing equations of design sensitivity

### 3.1 Implicit numerical integration of equations of motion

The variational form of the equations of motion for constrained mechanical systems is

$$\delta\mathbf{q}^T\left\{\mathbf{B}^T\left(\mathbf{M}\dot{\mathbf{Y}} + \Phi_Z^T\lambda - \mathbf{Q}\right)\right\} = 0 \tag{18}$$

where $\delta\mathbf{q}$ must be kinematically admissible for all tree structure joints, $\lambda \in \mathbf{R}^m$ is the Lagrange multiplier vector for cut joints[9] and m is the number of cut constraints. $\Phi \in \mathbf{R}^m$ and $\Phi_Z$ represent the position-level constraint vector and the constraint Jacobian matrix, respectively. The mass matrix $\mathbf{M}$ and the force vector $\mathbf{Q}$ are defined as follow.

$$\mathbf{M} = \mathrm{diag}\left(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{nbd}\right) \tag{19}$$

$$\mathbf{Q} = \left(\mathbf{Q}_1^T, \mathbf{Q}_2^T, \dots, \mathbf{Q}_{nbd}^T\right)^T \tag{20}$$

where nbd denotes the number of bodies. Since $\delta\mathbf{q}$ is arbitrary, The following equations of motion are obtained.

$$\mathbf{F} = \mathbf{B}^T\left(\mathbf{M}\dot{\mathbf{Y}} + \Phi_Z^T\lambda - \mathbf{Q}\right) = \mathbf{0} \tag{21}$$

The equations of motion can be implicitly rewritten by introducing $\mathbf{v} \equiv \dot{\mathbf{q}}$ as

$$\mathbf{F}\left(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}, \lambda\right) = \mathbf{0} \tag{22}$$

Successive differentiations of the position level constraint yield

$$\dot{\Phi}\left(\mathbf{q}, \mathbf{v}\right) = \Phi_\mathbf{q}\mathbf{v} - \nu = \mathbf{0} \tag{23}$$

$$\ddot{\Phi}\left(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}\right) = \Phi_\mathbf{q}\dot{\mathbf{v}} - \gamma = 0 \tag{24}$$

Equation (22) and all levels of constraints comprise the overdetermined differential algebraic system (ODAS). An algorithm for the backward differentiation formula (BDF) to solve the ODAS is given in[16] as follows.

$$\mathbf{H}(\mathbf{p}) = \begin{bmatrix} \mathbf{H}(\mathbf{p}) \\ \ddot{\Phi} \\ \dot{\Phi} \\ \Phi \\ \mathbf{U}_0^T\left(\frac{h}{b_0}\mathbf{R}_1\right) \\ \mathbf{U}_0^T\left(\frac{h}{b_0}\mathbf{R}_2\right) \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{q}, \mathbf{v}, \dot{\mathbf{v}}, \lambda) \\ \Phi_\mathbf{q}\dot{\mathbf{v}} - \gamma \\ \Phi_\mathbf{q}\mathbf{v} - \nu \\ \Phi_{(\mathbf{q})} \\ \mathbf{U}_0^T\left(\frac{h}{b_0}\dot{\mathbf{v}} - \mathbf{v} - \varsigma_1\right) \\ \mathbf{U}_0^T\left(\frac{h}{b_0}\dot{\mathbf{v}} - \mathbf{q} - \varsigma_2\right) \end{bmatrix} = 0 \tag{25}$$

where $\zeta_1 \equiv \frac{1}{b_0}\Sigma_{i=1}^k\ b_i\mathbf{v}_{(-i)}$ and $\zeta_2 \equiv \frac{1}{b_0}\Sigma_{i=1}^k\ b_i\mathbf{q}_{(-i)}$, in which k is the order of integration, $b_i$ are the BDF coefficients and $\mathbf{p} \equiv \left[\mathbf{q}^T, \mathbf{v}^T, \dot{\mathbf{v}}^T, \lambda^T\right]^T$. The columns of $\mathbf{U}_0 \in \mathbf{R}^{nr\times(nr-m)}$ constitute bases for the parameter space of the position-level constraints and is obtained by LU-decomposition of the constraint Jacobian so that the following matrix is nonsingular:

$$\begin{bmatrix} \Phi_\mathbf{q} \\ \mathbf{U}_0^T \end{bmatrix} \tag{26}$$

The number of equations and the number of unknowns in Eq. (25) are the same, and so Eq. (25) can be solved for $\mathbf{p}$. Newton Raphson method can be applied to obtain the solution **p**.

$$\mathbf{H_p}\Delta\mathbf{p} = -\mathbf{H} \quad (27)$$

$$\mathbf{p}^{i+1} = \mathbf{p}^{i} + \Delta\mathbf{p} \quad (28)$$

where

$$\mathbf{H_p} = \begin{bmatrix} \mathbf{F_q} & \mathbf{F_v} & \mathbf{F_{\dot{v}}} & \mathbf{F_\lambda} \\ \mathbf{\Phi_q} & 0 & 0 & 0 \\ \dot{\mathbf{\Phi}}_\mathbf{q} & \dot{\mathbf{\Phi}}_\mathbf{V} & 0 & 0 \\ \ddot{\mathbf{\Phi}}_\mathbf{q} & \ddot{\mathbf{\Phi}}_\mathbf{V} & \ddot{\mathbf{\Phi}}_{\dot{\mathbf{V}}} & 0 \\ \mathbf{U}_0^T & \beta_0\mathbf{U}_0^T & 0 & 0 \\ 0 & \mathbf{U}_0^T & \beta_0\mathbf{U}_0^T & 0 \end{bmatrix} \quad (29)$$

Recursive formulas for $\mathbf{H_p}$ and $\mathbf{H}$ in Eq. (27) will be derived in Section 4 to evaluate them efficiently.

Equation (25) is linear for the acceleration and the Lagrange multipliers but are nonlinear for the generalized coordinates and velocities. However, all variables are treated as nonlinear in solving them. Further investigations will be carried out in a near future to take advantage of the linearity for the acceleration and the Lagrange multipliers.

### 3.2 Implicit numerical integration of equations of design sensitivity

In general, the nonlinear constrained optimization problem can be written mathematically as follows:

$$\text{Minimize or Maximize} \;:\; \Omega(\mathrm{p},\tau) \quad \text{objective function} \quad (30)$$

Subject to :

$$\begin{aligned} & g_j(\mathrm{p},\tau) \le 0 && j = 1,m && \text{inequality constraints} \\ & h_k(\mathrm{p},\tau) = 0 && k = 1,l && \text{equality constraints} \\ & \tau_i^l \le \tau_i \le \tau_i^u && i = 1,n && \text{side constraints} \end{aligned} \quad (31)$$

where the $\tau$ is the vector of design variables. A mechanical system consists of bodies, joints, and force elements whose physical properties are described by various parameters. The geometric properties of a joint, the inertial properties of a body, and the compliance characteristics of a force element are candidate design variables. The objective function rm Omega in Eq. (30) and the constraint functions in Eq. (31) may be linear or nonlinear functions of the state and design variables. When an optimization algorithm takes a step or when a design engineer carries out a what-if study, the first derivatives of the objective function must be calculated as follows.

$$\frac{d\Omega}{d\tau} = \frac{\partial\Omega}{\partial \mathbf{p}}\frac{d\mathbf{p}}{d\tau} + \frac{\partial\Omega}{\partial\tau} \quad (32)$$

The most difficult term to compute in Eq. 32 is the $d\mathbf{p}/d\tau$ . Thus, this paper focuses on how to compute the $d\mathbf{p}/d\tau$ efficiently.

Differentiating Eq. (25) with respect to a design variable $\tau$ and appending the BDF yield the following equations of design sensitivity:

$$\Psi(\mathbf{p},\mathbf{p}_\tau) = \begin{bmatrix} \dfrac{d\mathrm{F}}{d\tau} \\ \dfrac{d\Phi}{d\tau} \\ \dfrac{d\dot{\Phi}}{d\tau} \\ \dfrac{d\ddot{\Phi}}{d\tau} \\ h'\mathbf{U}_0^T(\mathbf{R}_3) \\ h'\mathbf{U}_0^T(\mathbf{R}_4) \end{bmatrix} = \begin{bmatrix} \mathbf{F_q q_\tau} + \mathbf{F_v v_\tau} + \mathbf{F_{\dot{v}} \dot{v}_\tau} + \mathbf{F_\lambda \lambda_\tau} + \mathbf{F_\tau} \\ \Phi_\mathbf{q}\mathbf{q}_\tau + \Phi_\tau \\ \dot{\Phi}_\mathbf{q}\mathbf{q}_\tau + \dot{\Phi}_\mathbf{v}\mathbf{v}_\tau + \dot{\Phi}_\tau \\ \ddot{\Phi}_\mathbf{q}\mathbf{q}_\tau + \ddot{\Phi}_\mathbf{v}\mathbf{v}_\tau + \ddot{\Phi}_{\dot{\mathbf{v}}}\dot{\mathbf{v}}_\tau + \ddot{\Phi}_\tau \\ \mathbf{U}_0^T(h'\dot{\mathbf{v}}_\tau - \mathbf{v}_\tau - \varsigma_3) \\ \mathbf{U}_0^T(h'\mathbf{v}_\tau - \mathbf{q}_\tau - \varsigma_4) \end{bmatrix} = 0 \quad (33)$$

where $h' \equiv \frac{h}{b_o}$, $\zeta_3$ and $\zeta_4$ are collections of all previous values at integration knot points for $\mathbf{v}_\tau$ and $\mathbf{q}_\tau$ in the BDF and $\mathbf{p}_\tau \equiv [\mathbf{q}_\tau^T, \mathbf{v}_\tau^T, \dot{\mathbf{v}}_\tau^T, \lambda_\tau^T]^T$. Equations in Eq. (33) comprise the same number of equations as the unknowns and are solved for $\mathbf{p}_\tau$ as

$$\Psi_{\mathbf{p}_\tau}\mathbf{p}_\tau = -\eta \quad (34)$$

where $\eta = [\mathbf{F}_\tau^T, \Phi_\tau^T, \dot{\Phi}_\tau^T, \ddot{\Phi}_\tau^T, -(\mathbf{U}_0^T\varsigma_3)^T, -(\mathbf{U}_0^T\varsigma_4)^T]^T$. Since $\Psi_{\mathbf{p}_\tau}$ = $\mathbf{H_p}$, the matrix $\Psi_{\mathbf{p}_\tau}$ need not be calculated. Since various parameters in a system can be selected as the design vari-

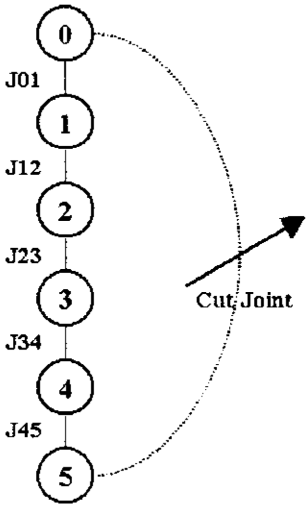


**Fig.3 Graph representation of the chain mechanism**

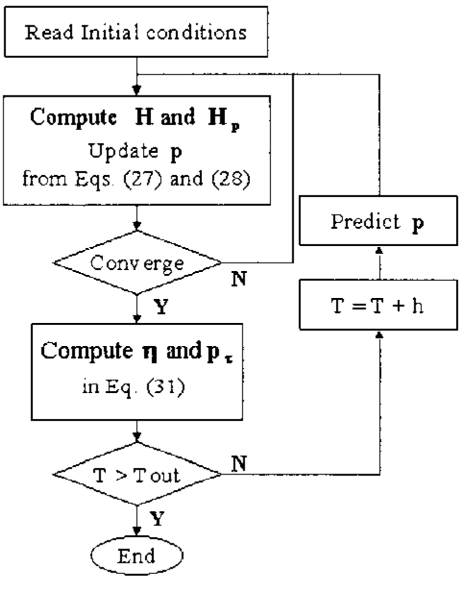


**Fig.4 Solution algorithm for sensitivity analysis**

ables, it is not easy to implement the formula for the right side $\eta$ of Eq. (34). For that reason, the finite-difference method (FDM) has been implemented for the calculation of $\eta$ in this paper. Though the proposed method is convenient to implement and accurate than the pure FDM, it still has inherent drawbacks of the FDM such as inefficiency for a design problem involving with a large number of design variables and difficulty in determining perturbation amount.

An implementation algorithm for the governing equations of motion and of design sensitivity is shown in Fig. 4.

## 4. Generalized recursive formulas and their applications

Careful inspection of the residual $\mathbf{H}$ and the Jacobian matrix $\mathbf{H}_p$ shows that types of necessary recursive formulas are classified into $\mathbf{Bx}$, $\mathbf{B}^T\mathbf{G}$, $\dot{\mathbf{B}}\mathbf{x}$, $(\mathbf{Bx})_q$, $(\mathbf{B}^T\mathbf{G})_q$, $(\dot{\mathbf{B}}\mathbf{x})_q$, $(\dot{\mathbf{B}}\mathbf{x})_v$ where $\mathbf{x} \in \mathbf{R}^{nr}$ and $\mathbf{G} \in \mathbf{R}^{nc}$ are arbitrary constant vectors. Formulas for $\mathbf{Bx}$ and $\mathbf{B}^T\mathbf{G}$ were derived in sections 2.2 and 2.3. The rest of formulas will be derived and then applied for computing $\mathbf{H}$ and $\mathbf{H}_p$ in this section. Note that all recursive formulas are tabulated in Table A.

Recursive formulas must be applied in a computational sequence that represents the system connectivity. The computational sequence can be determined from a representation of mechanical systems. A body and joint in a mechanical system are represented by a node and an edge in the corresponding graph, respectively. Nodes in the graph are divided into four disjoint sets in conjunction with a generalized coordinate $q_k$ to derive the recursive formulas systematically as

set I( $q_k$ ) = { outboard node of the edge having $q_k$ as its generalized coordinate }
set II( $q_k$ ) = { all outboard nodes of I( $q_k$ ) }
set III( $q_k$ ) = { all nodes between the base node and the inboard node of I( $q_k$ ), including the base and inboard nodes }
setIV( $q_k$ ) = { the complementary set of I( $q_k$ ) $\cup$ II( $q_k$ ) $\cup$ III( $q_k$ ) }

As an example, consider a governor mechanism and its graph representation shown in Figs. 5 and 6. The following sets of nodes in conjunction with $q_{24}$ (relative coordinate between nodes 2 and 4) for the graph shown in Fig. 6 are defined.

**Table A Summary of recursive formulas**

| Recursive formulas | $i \in \mathbf{I}(q_k)$ | $i \in \mathbf{II}(q_k)$ |
|---|---|---|
| $\mathbf{X}_q = (\mathbf{Bx})_q$ | $(\mathbf{X}_i)_{q_k} = (\mathbf{B}_{(i-1)i1})_{q_k} \mathbf{X}_{i-1} + (\mathbf{B}_{(i-1)i2})_{q_k} \mathbf{x}_{(i-1)i}$ | $(\mathbf{X}_i)_{q_k} = \mathbf{B}_{(i-1)i1} (\mathbf{X}_{i-1})_{q_k}$ |
| $\mathbf{g}_q = (\mathbf{B}^T\mathbf{G})_q$ | $(\mathbf{g}_{(i-1)i})_{q_k} = (\mathbf{B}_{(i-1)i2})_{q_k}{}^T \mathbf{S}_i$<br>$(\mathbf{S}_{i-1})_{q_k} = (\mathbf{B}_{(i-1)i1})_{q_k}{}^T \mathbf{S}_i$ | $(\mathbf{g}_{(i-1)i})_{q_k} = 0$<br>$(\mathbf{S}_{i-1})_{q_k} = 0$ |
| $\dot{\mathbf{X}}_q = (\dot{\mathbf{B}}\mathbf{x})_q$ | $(\dot{\mathbf{X}}_i)_{q_k} = (\dot{\mathbf{B}}_{(i-1)i1})_{q_k} \mathbf{X}_{i-1} + (\mathbf{B}_{(i-1)i1})_{q_k} \dot{\mathbf{X}}_{i-1} + (\dot{\mathbf{B}}_{(i-1)i2})_{q_k} \mathbf{x}_{(i-1)i}$ | $(\dot{\mathbf{X}}_i)_{q_k} = \dot{\mathbf{B}}_{(i-1)i1} (\mathbf{X}_{i-1})_{q_k} + \mathbf{B}_{(i-1)i1} (\dot{\mathbf{X}}_{i-1})_{q_k}$ |
| $\dot{\mathbf{X}}_v = (\dot{\mathbf{B}}\mathbf{x})_v$ | $(\dot{\mathbf{X}}_i)_{v_k} = (\dot{\mathbf{B}}_{(i-1)i1})_{v_k} \mathbf{X}_{i-1} + (\dot{\mathbf{B}}_{(i-1)i2})_{v_k} \mathbf{x}_{(i-1)i}$ | $(\dot{\mathbf{X}}_i)_{v_k} = \dot{\mathbf{B}}_{(i-1)i1} (\mathbf{X}_{i-1})_{v_k} + \mathbf{B}_{(i-1)i1} (\dot{\mathbf{X}}_{i-1})_{v_k}$ |

| Recursive formulas | $i \in \mathbf{III}(q_k)$ | $i \in \mathbf{IV}(q_k)$ |
|---|---|---|
| $\mathbf{X}_q = (\mathbf{Bx})_q$ | $(\mathbf{X}_i)_{q_k} = 0$ | $(\mathbf{X}_i)_{q_k} = \mathbf{B}_{(i-1)i1} (\mathbf{X}_{i-1})_{q_k}$ |
| $\mathbf{g}_q = (\mathbf{B}^T\mathbf{G})_q$ | $(\mathbf{g}_{(i-1)i})_{q_k} = \mathbf{B}^T{}_{(i-1)i2} (\mathbf{S}_i)_{q_k}$<br>$(\mathbf{S}_{i-1})_{q_k} = (\mathbf{B}^T{}_{(i-1)i1} (\mathbf{S}_i)_{q_k}$ | $(\mathbf{g}_{(i-1)i})_{q_k} = 0$<br>$(\mathbf{S}_{i-1})_{q_k} = 0$ |
| $\dot{\mathbf{X}}_q = (\dot{\mathbf{B}}\mathbf{x})_q$ | $(\dot{\mathbf{X}}_i)_{q_k} = 0$ | $(\dot{\mathbf{X}}_i)_{q_k} = 0$ |
| $\dot{\mathbf{X}}_v = (\dot{\mathbf{B}}\mathbf{x})_v$ | $(\dot{\mathbf{X}}_i)_{v_k} = 0$ | $(\dot{\mathbf{X}}_i)_{v_k} = 0$ |

| Recursive formulas | $i \in \mathbf{I}(q_k)$ or $i \in \mathbf{II}(q_k)$ or $i \in \mathbf{III}(q_k)$ or $i \in \mathbf{IV}(q_k)$ |
|---|---|
| $\mathbf{X} = \mathbf{Bx}$ | $\mathbf{X}_i = \mathbf{B}_{(i-1)i1} \mathbf{X}_{i-1} + \mathbf{B}_{(i-1)i2} \mathbf{x}_{(i-1)i}$ |
| $\mathbf{g}_q = \mathbf{B}^T\mathbf{G}$ | $\mathbf{g}_{(i-1)i} = \mathbf{B}^T{}_{i(i-1)2} (\mathbf{G}_{i+1} + \mathbf{S}_{i+1})$<br>$\mathbf{S}_n =$<br>$\mathbf{S}_i = \mathbf{B}^T{}_{i(i-1)1} (\mathbf{G}_{i+1} + \mathbf{S}_{i+1})$ |
| $\dot{\mathbf{X}} = \dot{\mathbf{B}}\mathbf{x}$ | $\dot{\mathbf{X}}_i = \dot{\mathbf{B}}_{(i-1)i1} \mathbf{X}_{i-1} + \mathbf{B}_{(i-1)i1} \dot{\mathbf{X}}_{i-1} + \dot{\mathbf{B}}_{(i-1)i2} \mathbf{x}_{(i-1)i}$ |

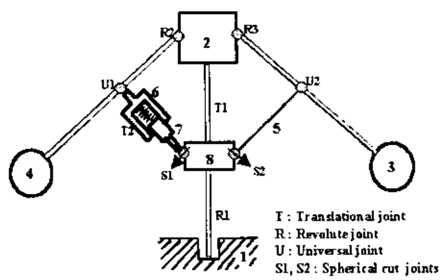


**Fig. 5 A governor mechanism**

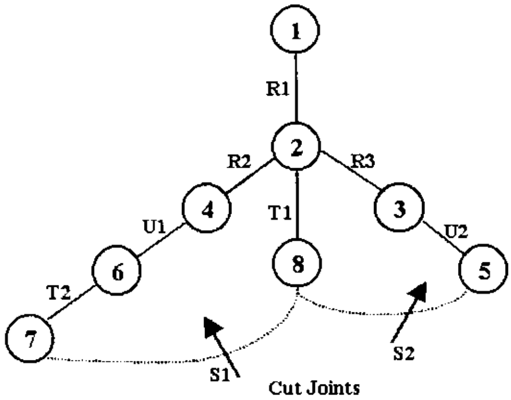


**Fig. 6 Graph representation of the governor mechanism**

set I ( $q_{24}$ ) = { node 4 }
set II ( $q_{24}$ ) = { nodes 6 and 7 }
set III ( $q_{24}$ ) = { nodes 1 and 2 }
set IV ( $q_{24}$ ) = { nodes 3, 5, and 8 }

## 4.1 Recursive formula for $(\dot{\mathbf{X}} = \dot{\mathbf{B}}\mathbf{x})$

The recursive formula for $\dot{\mathbf{B}}\mathbf{x} \in \mathbf{R}^{nc}$ is easily obtained by differentiating Eq. (12),

$$\dot{\mathbf{X}}_i = \dot{\mathbf{B}}_{(i-1)i1} \mathbf{X}_{i-1} + \mathbf{B}_{(i-1)i1} \dot{\mathbf{X}}_{i-1} + \dot{\mathbf{B}}_{(i-1)i2} \mathbf{x}_{(i-1)i} \quad (35)$$

This recursive formula can be applied to compute the Cartesian acceleration $\dot{\mathbf{Y}}$, with known relative velocity and acceleration.

## 4.2 Recursive formula for $\mathbf{X}_q = (\mathbf{Bx})_q$

In order to obtain the recursive formula for $(\mathbf{Bx})_q$, Eq. (12) is differentiated with respect to $q_k$ for k = 1,···,cdots, nr as follows.

$$(\mathbf{X}_i)_{q_k} = \left(\mathbf{B}_{(i-1)i1}\right)_{q_k} \mathbf{X}_{i-1} + \mathbf{B}_{(i-1)i1} (\mathbf{X}_{i-1})_{q_k} + \left(\mathbf{B}_{(i-1)i2}\right)_{q_k} \mathbf{x}_{(i-1)i} \quad (36)$$

Since the matrices $\mathbf{B}_{(i-1)i1}$ and $\mathbf{B}_{(i-1)i2}$ depend only on the relative coordinates for joint (i-1)i, their partial derivatives with respect to generalized coordinates other than $\mathbf{q}_{(i-1)i}$ vanish. In other words, the partial derivatives vanish if $q_k$ does not belong to set I ( $q_k$ ).

Therefore, if body i is an element of set II ( $q_k$ ), Eq. (36) becomes

$$(\mathbf{X}_i)_{q_k} = \dot{\mathbf{Y}}_{(i-1)i1}(\mathbf{X}_{i-1})_{q_k} \tag{37}$$

If body i belongs to set III ( $q_k$ ) $\cup$ IV ( $q_k$ ), $\mathbf{X}_i$ is not affected by $q_k$. As a result, Eq. (37) is further simplified as follows.

$$(\mathbf{X}_i)_{q_k} = 0 \tag{38}$$

If body i is an element of set I ( $q_k$ ), body i-1 is naturally its inboard body and it belongs to set III ( $q_k$ ). Using Eq. (38), Eq. (36) becomes

$$(\mathbf{X}_i)_{q_k} = \left(\mathbf{B}_{(i-1)i1}\right)_{q_k}\mathbf{X}_{i-1} + \left(\mathbf{B}_{(i-1)i2}\right)_{q_k}\mathbf{x}_{(i-1)i} \tag{39}$$

This recursive formula can be applied to compute the partial derivative of the Cartesian velocity with respect to relative coordinates $\mathbf{Y}_q$. As an example, if Eqs. (37), (38) and (39) are applied to compute $\mathbf{Y}_{q24}$, the resulting equations are shown in Fig. 7.

The recursive formulas for $(\mathbf{B}^T\mathbf{G})_q$, $(\dot{\mathbf{B}}\mathbf{x})_q$ and $(\dot{\mathbf{B}}\mathbf{x})_v$ are obtained as in Table A by following the similar steps taken in this sections.

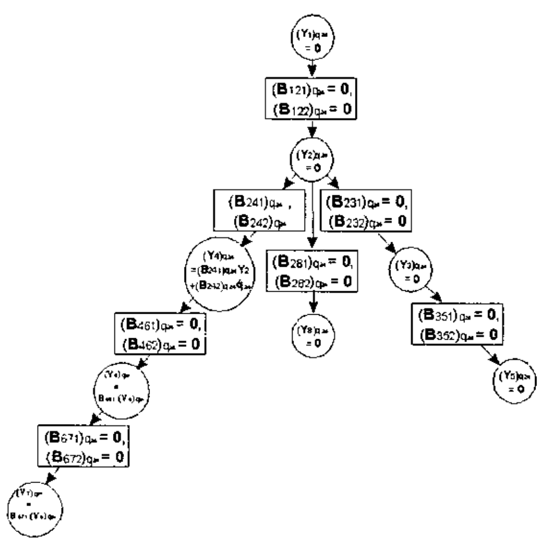

**Fig. 7 Computation sequence for $\mathbf{Y}_{q24}$**

# 5. Numerical example

To show the validity of the proposed formulation, dynamic and design sensitivity analyses of a passenger vehicle were performed. The MacPherson strut and multi-link suspensions were employed as its front and rear suspensions, respectively. The MacPherson strut suspension consists of a knuckle, a lower control arm, a strut, and a rack. The chassis and lower control arm are linked with a bushing element, the lower control arm and knuckle, the strut and chassis with ball joints, the knuckle and strut with a translational joint, the steering rack and knuckle with a tie rod, and the chassis and knuckle with a spring-dashpot element. The inertia properties and spring and damping constants are shown in Table B.1.

A multi-link suspension system consists of a knuckle, a strut, two toe control arms, a camber control arm and a trail link. The toe control arms and chassis, the knuckle and toe control arms, the trail link and knuckle, the trail link and chassis, and the camber control arm and chassis are linked with bushing elements. The camber control arm and knuck-

**Table B.1 Inertia properties and spring/damping constants of the front suspension**

| Body | Mass(kg) | Moment of inertia ( $kg \cdot m^2$ ) |
|---|---|---|
| Chassis | 1460.0 | 484, 2344, 2245 |
| Rack | 1.0 | 1.0, 1.0, 2.0 |
| Lower control arm | 3.0 | 2.0, 4.0, 2.0 |
| Tie rod | 5.0 | 4.0, 4.0, 4.0 |
| Knuckle | 4.0 | 3.0, 6.0, 3.0 |
| Strut | 2.0 | 1.0, 1.0, 2.0 |
| Spring constant | 18639 N/m | |
| Damping coefficient | 1386 Ns/m | |

**Table B.2 Inertia properties and spring/damping constants of the rear suspension**

| Body | Mass(kg) | Moment of inertia ( $kg \cdot m^2$ ) |
|---|---|---|
| Toe control arm | 2.0 | .0, 1.0, 2.0 |
| Chamber control arm | 2.0 | 2.0, 3.0, 2.0 |
| Trail link | 2.0 | 2.0, 2.0, 2.0 |
| knunckl | 3.0 | 3.0, 4.0, 3.0 |
| Stru | 2.0 | 2.0, 3.0, 2.0 |
| Spring constant | 21582 N/m | |
| Damping coefficient | 1021 Ns/m | |

le are linked by a ball joint. The material properties and spring and damping constants are shown in Table B.2. In summary, the system consists of 35 bodies, 34 joints, 30 bushings, and 6 springs and dampers and has 148 degrees of freedom.

A J-turn simulation of the vehicle was carried out with an initial velocity of 80km/h and step steering input shown in Fig. 8. Roll acceleration obtained from the J-turn simulation is shown in Fig. 9.

The damping coefficient of the suspension system was chosen as a design variable to observe the effect of the damping coefficient on the roll angle.The proposed sensitivity analysis was carried out and the sensitivity of the roll acceleration with respect to the damping coefficient change was obtained, as in Fig. 10.

The sensitivity result was validated against that of the FDM calculation.

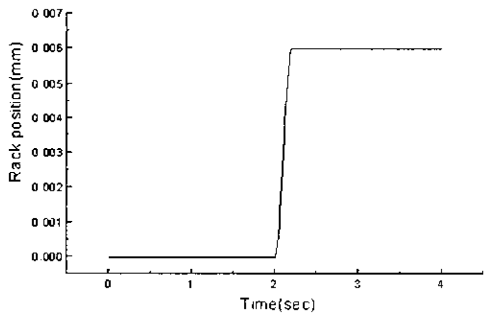


**Fig. 8 Rack position**

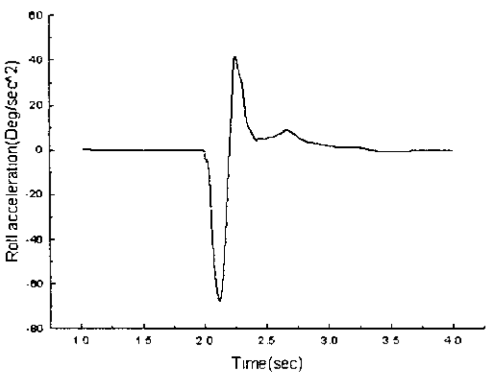


**Fig. 9 Roll acceleration of chassis**

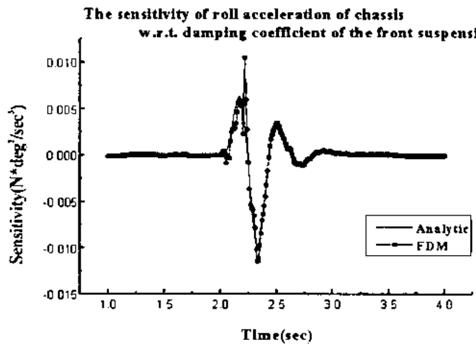


**Fig. 10 The sensitivity of roll acceleration**

Since the absolute value of the sensitivity is very small, the integration error-tolerance must be very small as well. Otherwise, accurate FDM results can not be obtained. The error tolerance of $10^{-5}$ for the FDM analysis was used for this example. The sensitivities of proposed method and FDM are shown to be close in Fig. 10, which validates the proposed method. The sensitivity analysis was performed on a IBM compatible computer(266 MHz) and took about 10 min. This indicates that the sensitivity analysis of a fairly complicated system can be accomplished in a moderate period on a desktop computer if the number of design variables is not too excessive.

# 6. Conclusions

To compute the sensitivity for mechanical design, the finite difference method(FDM) has been used. But the method has always had a problem which is how to determine the variations of the design variables. These variations are closely affected the stability and accuracy for numerical analysis. Since the design sensitivity analysis method is proposed is analytically derived the generalized recursive formulas, the problem is swept away. The velocity-transformation method was employed to transform the equations of motion from the Cartesian to the relative coordinate system. The computational structure of the equations of motion was examined to classify all necessary computations into several categories. The generalized recursive formula for each category was then applied whenever such a category of computation was encountered. Since the velocity-transformation method yields the equations of motion in a compact form and computational efficiency is achieved by the generalized recursive formulas, the proposed method is not only easy to implement but also efficient. Since the direct differentiation method is used to formulate the governing equations of design sensitivity whose right hand side is computed by the FDM. The proposed method is semi-analytic. As an example, the design sensitivity analysis of a large scale vehicle system due to a damping coefficient change was performed.

The computing time indicated that the sensitivity based design iteration of a large scale mechanical system is possible on a PC level computer with the proposed method if the number of design variables is not too excessive.

## Reference


(1) A. G. Erdman and G. N. Sandor, 1991, *Mechanism Design: Analysis and Synthesis, Volume I*, Prentice-Hall.

(2) R. Serban and J. S. Freeman, 1996, *Direct Differentiation Methods for The Design Sensitivity of Multibody Dynamic Systems*, ASME Design Engineering Technical Conferences and Computers in Engineering Conference, 96-DETC/DAC-1087, pp. 1-7.

(3) H. W. Kim, D. S. Bae, and K. K. Choi, 1999, *Configuration Design Sensitivity Analysis of Kinematic Responses of Mechanical Systems, Mech. Struct. and Machines*, Volume 27, No. 2, pp 203-215.

(4) E. Kita, Y. Katoka, and N. Kamiya. 1999, *First- and Second-Order design sensitivity analysis scheme based on Trefftz method, ASME J. Mechanical Design*, Volume 121, pp. 84-91.

(5) P. *Whittle*, 1996, *Optimal Control : basic and beyond*, Wiley.

(6) DADS User's Manual, CADSI, Iowa, USA

(7) ADAMS User's Manual, MDI, Michigan, USA

(8) J. Garc'ia de Jal'on, J. Unda, and A. Avell°Øo, 1986, *Natural Coordinates for the Computer Analysis of Multibody Systems, Computer Methods in Applied Mechanics and Engineering*, Volume 56, pp. 309-327.

(9) J. Wittenburg, 1977, *Dynamics of Systems of Rigid Bodies*, B. G. Teubner, Stuttgart.

(10) W. Hooker and G. Margulies, 1965, The Dynamical *Attitude Equtation for an n-body Satellite, Journal of the Astrnautical Science*, Volume 12, pp. 123-128.

(11) R. Featherstone, 1983, *The Calculation of Robot Dynamics Using Articulated-Body Inertias, Int. J. Roboics Res.*, Volume 2, pp. 13-30.

(12) D. S. Bae and E. J. Haug, 1988, *A Recursive Formulation for Constrained Mechanical System Dynamics: Part II.Closed Loop Systems, Mech. Struct. and Machines*, Volume 15, No. 4, pp. 481-506.

(13) S. S. Kim and E. J. Haug, 1988, *A Recursive Formulation for Flexible Multibody Dynamics. Part I: Open-Loop Systems, Computer Methods in Applied Mechanics and Engineering*, Volume 71, pp. 293-314.

(14) D. S. Bae, J. M. Han, and H. H. Yoo, 1999, *A Generalized Recursive Formulation for Constrained Mechanical System Dynamics, Mech. Struct. and Machines*, Volume 27, No. 3, pp 293-315.

(15) H. Ashrafiuon, N. K. Mani, 1990, *Analysis and Optimal Design of Spatial Mechanical Systems, Journal of ASME*, Volume 112, pp. 200-207.

(16) J. Yen, E. J. Haug, and F. A. Potra, 1993, *Constrained Equations of Motion in Multibody Dynamics as ODEs on Manifolds, SIAM J. Numer. Anal.*, Volume 30, No. 2, pp 553-568.

(17) M. G. Lee and E. J. Haug, 1992, *Stability and Convergence for Difference Approximations of Differential-Algebraic Equations of Mechanical System Dynamics*, Technical Report R-157, Center for Simulation and Design Optimization, Department of Mechanical Engineering, and Department of Mathematics, The University of Iowa, Iowa City, Iowa.

(18) J. Angeles, 1997, *Fundamentals of Robotic Mechanical Systems*, Springer.