한국공작기계학회논문집 Vol.10 No.2 2001. 4.
Transactions of the Korean Society of Machine Tool Engineers

A real time method of vehicle system dynamics

D. S. Bae*

Abstract

Super computers has been utilized to carry out vehicle dynamics in real time. This research propose an implicit integration method for vehicle state variables. Newton chord method is employed to solve the equations of motion and constraints. The equations of motion and constraints are formulated such that the Jacobian matrix for Newton chord method is needed to be computed only once for a dynamic analysis. Numerical experiments showed that the Jacobian matrix generated at the initial time could have been utilized for the Newton chord iterations throughout simulations under various driving conditions. Convergence analysis of Newton chord method with the proposed Jacobian updating method is carried out. The proposed algorithm yielded accurate solutions for a prototype vehicle multibody model in realtime on a 400 Mhz PC compatible.

Key Words : Implicit numerical method, Drive train, Newton chord method

1. Introduction

Realtime simulation capability has introduced a new concept of virtual prototyping in designing and testing of the mechanical systems. Several sophisticated vehicle driving simulators have been developed and used for design, testing, and medical and safety studies[1, 2].

High performance computers have been used for realtime simulation of multibody vehicle dynamics models consisting of many bodies and joints, a drive train model, stabilization bars, and tires. Vehicle models have been often simplified to run the models in realtime on relatively low cost computers such as PC compatibles. This research develops an efficient implementation algorithm with an implicit integration method and the relative generalized coordinate formulation so that the relatively low cost com-

* Department of mechanical engineering
hanyang university, ansan, south korea
(dabae@email.hanyang.ac.kr)

puter can be used for the realtime simulation of the multibody vehicle dynamics models without sacrificing modeling details.

The implicit integration method requires to generate and solve linear systems of augmented equations of motion and nonlinear systems of constraint equations [3,4]. The most time taking processes associated with the implicit method are generation and LU decomposition of the coefficient matrices of the augmented equations of motion and constraint Jacobian at every time steps. In order to save the computation time in these processes, Newton chord method is employed and the governing equations are formulated such that Jacobian matrices for Newton chord method need to be updated once in many integration steps. Convergence analysis of Newton chord method with the proposed Jacobian updating method is carried out for a prototype vehicle model.

Natural coordinate method was proposed by Garcia and applied for realtime simulation in Refs. [5,6]. The velocity transformation method has been developed by Wittenburg in Ref. [7]. Graph theory has been successfully applied to handle general mechanical systems with the relative generalized coordinates [8,9]. Featherstone has developed a recursive formulation for the equations of motion [10]. Computational cost has increased only linearly to the number of bodies in a system. The recursive formulation has been extended to general systems having closed loops in Ref. [11] and having flexible bodies in Ref. [12]. Variational vector calculus has been used to systematically derive the individual recursive formulas for the mass matrix, the constraint Jacobian matrix, and the generalized forces. The individual recursive formulas are categorized into several types and are generalized in this research to efficiently compute the residuals of Newton chord method.

Section 2 presents a prototype vehicle model. Kinematics and generalized recursive formulas for the vehicle are developed in section 3. A solution method for the overdetermined differential algebraic equations arising from the vehicle equations of motion and constraints is presented in section 4. Numerical algorithm is proposed in section 5. Numerical experiments are carried out in section 6. A lane change, J-turn, and bump run maneuver simulations of the vehicle model are performed to show the effectiveness and validity of the proposed method. Conclusions are drawn in section 7.

2. A multibody vehicle model

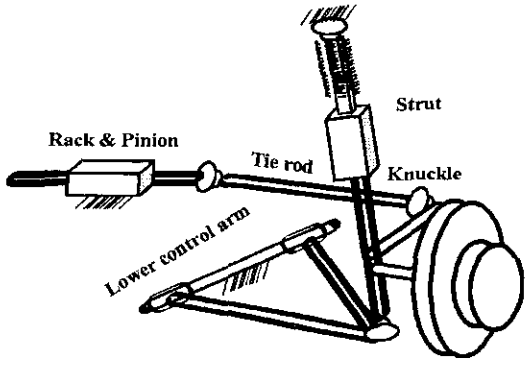

Fig. 1 MacPherson strut suspension

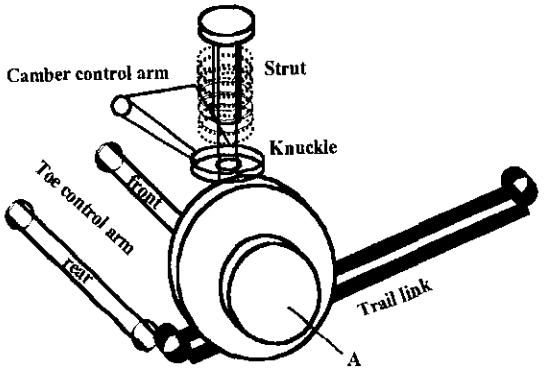

Fig. 2 4-links suspension

Though the proposed method is applicable to general multibody vehicle models, presentation focuses on a prototype vehicle model which consists of suspension, steering, drive train, and tire subsystems.

The prototype vehicle employs the MacPherson strut front and 4-link type rear suspension systems. The schematic diagram of both suspension systems are shown in Figs. 1 and 2, respectively. The MacPherson strut system consists of a lower control arm, a wheel knuckle, a

Table 1 Generalized coordinates for each joint

Joint number	Joint type	Connecting bodies	Generalized coordinates
J0	Free joint	ground-chassis(b0)	q0,q1,q2,q3,q4,q5
J1	Translational	chassis(b0)-rack(b1)	q6
J2	Revolute	chassis(b0)-lower control arm(b2)	q7
J3	Ball	chassis(b0)-strut(b4)	q8,q9,q10
J4	Translational	strut(b4)-knuckle(b3)	q11
J5	Revolute	chassis(b0)-lower control arm(b5)	q12
J6	Ball	chassis(b0)-strut(b7)	q13,q14,q15
J7	Translational	strut(b7)-knuckle(b6)	q16
J8	Revolute	chassis(b0)-camber control arm(b8)	q17
J9	Ball	camber control arm(b8)-knuckle(b9)	q18,q19,q20
J10	Revolute	chassis(b0)-camber control arm(b10)	q21
J11	Ball	camber control arm(b10)-knuckle(b11)	q22,q23,q24
J22	Revolute	knuckle(b3)-tire(T1)	q25
J23	Revolute	knuckle(b6)-tire(T2)	q26
J24	Revolute	knuckle(b9)-tire(T3)	q26
J25	Revolute	knuckle(b11)-tire(T4)	q27
none	none	Engine	q28

strut, and a tie-rod. The 4-link type consists of a knuckle, a strut, two toe control arms, a camber control arm, and a trail link. The rack and pinion mechanism that is connected to the wheel knuckle by a tie-rod transfers an driver input to the wheel assembly. The UA tire model is employed for the tire force computation [13]. The stabilization bar is modeled as two rotational springs that act along the two revolute joints between the chassis and the lower control arms. The drive train model presented in Ref. [14] and summarized in Fig. 3 is used in this research.

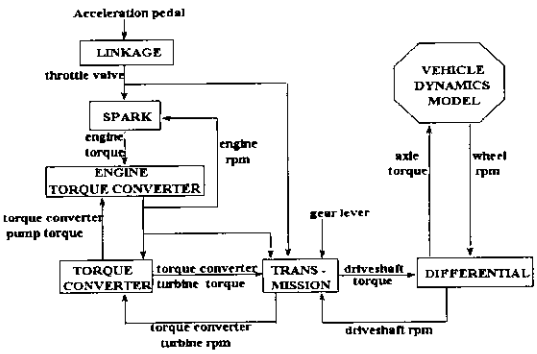

Fig. 3 A drive train model

In order to systematically derive the equations of motion and constraints, a mechanical system has been represented by a graph. A body and joint are represented by a node and edge, respectively. Fig. 4 shows the graph theoretic representation of the prototype vehicle system. The closed loops in the graph are opened to form the spanning tree in Fig. 5 by cutting ball and distance joints [7]. The resulting generalized coordinates for each joint in the spanning tree are shown in Table 1.

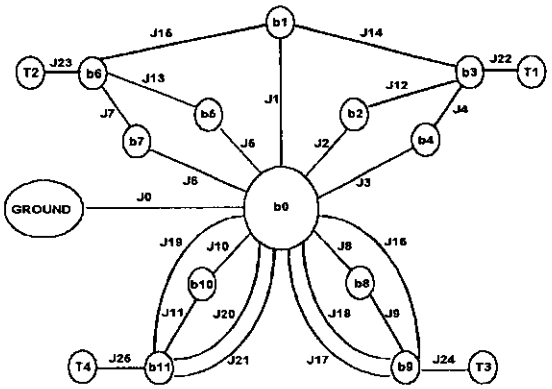

Fig. 4 Graph representation for the prototype vehicle

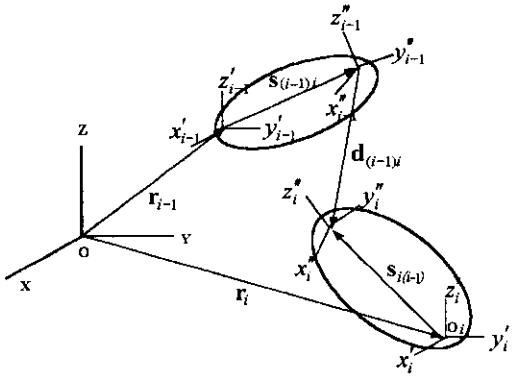

Fig. 6 Kinematic relationship between two adjacent rigid bodies

3. Kinematics and generalization of recursive formulas

Consider a pair of contiguous bodies shown in Fig. 6 from the spanning tree in Fig. 5. Body **i-1** is assumed to be an inboard body of body **i** in the spanning tree. The position of point $\mathbf{O}_i$ is obtained as

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \mathbf{s}_{(i-1)i} + \mathbf{d}_{(i-1)i} - \mathbf{s}_{i(i-1)} \quad (1)$$

The angular velocity in the body reference frame is obtained as

$$\omega'_i = \mathbf{A}^{\mathrm{T}}{}_{(i-1)i}\,\omega'_{i-1} + \mathbf{A}^{\mathrm{T}}{}_{(i-1)i}\mathbf{H}'_{(i-1)i}\dot{\mathbf{q}}_{(i-1)i} \quad (2)$$

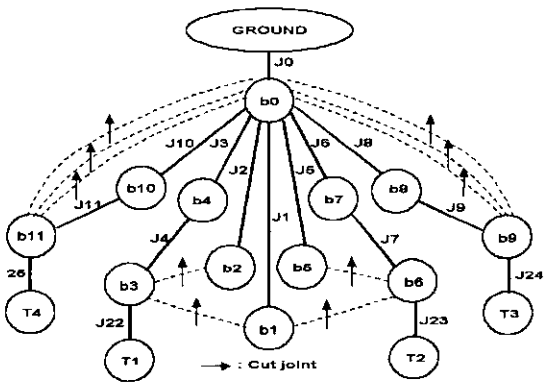

Fig. 5 Spanning tree for the prototype vehicle

where $\mathbf{H}'$ is determined by the axis of rotation. $\mathbf{A}$ denotes an orientation matrix for a body reference frame and $\mathbf{A}_{(i-1)i} = \mathbf{A}_{(i-1)}{}^{\mathrm{T}}\mathbf{A}_i$. Differentiation of Eq.1 yields

$$\begin{aligned}\mathbf{A}_i\dot{\mathbf{r}}'_i = {} & \mathbf{A}_{(i-1)}\,\dot{\mathbf{r}}'_{(i-1)} \\ & - \mathbf{A}_{(i-1)}\,\tilde{\mathbf{s}}'_{(i-1)i}\,\omega'_{(i-1)} \\ & - \mathbf{A}_{(i-1)}\,\tilde{\mathbf{d}}'_{(i-1)i}\,\omega'_{(i-1)} \\ & + \mathbf{A}_i\tilde{\mathbf{s}}'_{i(i-1)}\,\omega'_i \\ & + \mathbf{A}_{(i-1)}\,(\mathbf{d}'_{(i-1)i})_{\mathbf{q}_{(i-1)i}}\,\dot{\mathbf{q}}_{(i-1)i}\end{aligned} \quad (3)$$

where symbols with tildes denote skew symmetric matrices associated with their vector elements, $\dot{\mathbf{r}} = \mathbf{A}\dot{\mathbf{r}}'$, and $\mathbf{q}_{(i-1)i}$ denotes relative coordinate vector. Substituting ω'_i of Eq.2 and multiplying both sides of Eq.3 by $\mathbf{A}_i^{\mathrm{T}}$ yield

$$\begin{aligned}\dot{\mathbf{r}}'_i = {} & \mathbf{A}^{\mathrm{T}}{}_{(i-1)i}\dot{\mathbf{r}}'_{(i-1)} \\ & - \mathbf{A}^{\mathrm{T}}{}_{(i-1)i}\,(\tilde{\mathbf{s}}'_{(i-1)i} + \tilde{\mathbf{d}}'_{(i-1)i} \\ & - \mathbf{A}_{(i-1)i}\tilde{\mathbf{s}}'_{i(i-1)}\mathbf{A}^{\mathrm{T}}{}_{(i-1)i})\omega'_{(i-1)} \\ & + \mathbf{A}^{\mathrm{T}}{}_{(i-1)i}\,((\mathbf{d}'_{(i-1)i})_{\mathbf{q}(i-1)i} \\ & + \mathbf{A}_{(i-1)i}\tilde{\mathbf{s}}'_{i(i-1)}\mathbf{A}^{\mathrm{T}}{}_{(i-1)i}\mathbf{H}'_{(i-1)i})\dot{\mathbf{q}}_{(i-1)i}\end{aligned} \quad (4)$$

$$\mathbf{Y}_i = \mathbf{B}_{(i-1)i1}\mathbf{Y}_{(i-1)} + \mathbf{B}_{(i-1)i2}\dot{\mathbf{q}}_{(i-1)i} \quad (5)$$

for which $\dot{\mathbf{A}}_i = \mathbf{A}_i \tilde{\omega}'_i$ are used. Combining Eqs.2 and 4 yields the following recursive velocity equation for a pair of contiguous bodies.

where

$$\mathbf{Y}_i = [\mathbf{r}_i'^T \omega_i'^T]^T \tag{6.a}$$

$$\mathbf{B}_{(i-1)i1} = \begin{bmatrix} \mathbf{A}_{(i-1)i}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{(i-1)i}^T \end{bmatrix} \begin{bmatrix} \mathbf{I} & -(\tilde{\mathbf{s}}'_{(i-1)i} + \tilde{\mathbf{d}}'_{(i-1)i} - \mathbf{A}_{(i-1)i}\tilde{\mathbf{s}}'_{i(i-1)}\mathbf{A}^T_{(i-1)i}) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{6.b}$$

$$\mathbf{B}_{(i-1)i2} = \begin{bmatrix} \mathbf{A}_{(i-1)i}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{(i-1)i}^T \end{bmatrix} \begin{bmatrix} (\mathbf{d}'_{(i-1)i})_{\mathbf{q}_{(i-1)i}} + \mathbf{A}_{(i-1)i}\tilde{\mathbf{s}}'_{i(i-1)}\mathbf{A}^T_{(i-1)i}\mathbf{H}'_{(i-1)i} \\ \mathbf{H}'_{(i-1)i} \end{bmatrix} \tag{6.c}$$

It is important to note that 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) **i**.

Similarly, the recursive virtual displacement relationship is obtained as follows.

$$\delta\mathbf{Z}_i = \mathbf{B}_{(i-1)i1}\delta\mathbf{Z}_{(i-1)} + \mathbf{B}_{(i-1)i2}\delta\mathbf{q}_{(i-1)i} \tag{7}$$

If the recursive formula in Eq.5 is respectively applied to all joints of the spanning tree in Fig. 5, the following relationship between the Cartesian and relative generalized velocities can be obtained:

$$\mathbf{Y} = \mathbf{B}\dot{\mathbf{q}} \tag{8}$$

The $\mathbf{B}_{(i-1)i}$' s of the recursive formulas are function of only the relative generalized coordinates. A virtual joint of the free joint which has six degrees of freedom has been defined between the ground and the chassis for uniform treatment of all bodies. The generalized velocities for the free joint are defined as $\mathbf{Y}_{free}$ in Eq.6.a to make the **B** matrix in Eq.8 free from the generalized coordinates for the free joint. Since the generalized velocities and time derivative of the generalized coordinates are different for the free joint, the latter is computed from the former for the numerical integration. The fact that the **B** matrix is free from the generalized coordinates for the free joint plays a key role in reducing the update frequency of the system Jacobian matrix for Newton chord iterations that will be presented in later sections.

The Cartesian velocity $\mathbf{Y} \in \mathbf{R}^{nc}$ with a given $\dot{\mathbf{q}} \in \mathbf{R}^{nr}$ can be evaluated either by using Eq.8 obtained from symbolic substitutions or by using Eq.5 with recursive numeric substitution of $\mathbf{Y}_i$s. The nc and nr denote the numbers of the Cartesian and relative generalized coordinates, respectively. Since both formulas give an identical result and recursive numeric substitution is proven to be more efficient [11], matrix multiplication $\mathbf{B}\dot{\mathbf{q}}$ with a given $\dot{\mathbf{q}}$ will be actually evaluated by using Eq.5. Since $\dot{\mathbf{q}}$ in Eq.11 is an arbitrary vector in $\mathbf{R}^{nr}$, Equations.5 and 8, which are computationally equivalent, are actually valid for any vector $\mathbf{x} \in \mathbf{R}^{nr}$ such that

$$\mathbf{X} = \mathbf{B}\mathbf{x} \tag{9.a}$$

and

$$\mathbf{X}_i = \mathbf{B}_{(i-1)i1}\mathbf{X}_{(i-1)} + \mathbf{B}_{(i-1)i2}\mathbf{x}_{(i-1)i} \tag{9.b}$$

where $\mathbf{x} \in \mathbf{R}^{nc}$ is the resulting vector of multiplication of **B** and $\mathbf{x}$. As a result, transformation of $\mathbf{x} \in \mathbf{R}^{nr}$ into $\mathbf{Bx} \in \mathbf{R}^{nc}$ is actually calculated by recursively applying Eq.9.b to achieve computational efficiency in this research.

Inversely, it is often necessary to transform a vector **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 joint space with a known force in the Cartesian space. The virtual work done by a Cartesian force $\mathbf{Q} \in \mathbf{R}^{nc}$ is obtained as follows.

$$\delta W = \delta\mathbf{Z}^T\mathbf{Q} \tag{10}$$

where $\delta\mathbf{Z}$ must be kinematically admissible for all joints in a system. Substitution of $\delta\mathbf{Z} = \mathbf{B}\delta\mathbf{q}$ into Eq.10 yields

$$\delta W = \delta \mathbf{q}^T \mathbf{B}^T \mathbf{Q} = \delta \mathbf{q}^T \mathbf{Q}^* \quad (11)$$

wher $\mathbf{Q}^* \equiv \mathbf{B}^T\mathbf{Q}$. Equation 11 can be written in a summation form as

$$\delta W = \sum_{i=0}^{nbd-1} \delta \mathbf{q}_{i(i+1)}^T \mathbf{Q}_{i(i+1)}^* \quad (12)$$

where nbd denotes the number of bodies in a system. On the other hand, the symbolic substitution of the recursive virtual displacement relationship Eq.7 into Eq.10 along all chains (starting from the terminal bodies toward inboard bodies) yields

$$\delta W = \sum_{i=0}^{n-1} \delta \mathbf{q}_{i(i+1)}^T \{\mathbf{B}_{i(i+1)2}^T (\mathbf{Q}_{i+1} + \mathbf{S}_{i+1})\} \quad (13)$$

where

$$\mathbf{S}_n = \mathbf{0}$$
$$\mathbf{S}_{i+1} \equiv \mathbf{B}_{(i+1)(i+2)1}{}^T (\mathbf{Q}_{i+2} + \mathbf{S}_{i+2}) \quad (14)$$

By equating Eqs.12 and 13, the following recursive formula for $\mathbf{Q}^*$ is obtained.

$$\mathbf{Q}_{i(i+1)}^* = \mathbf{B}_{i(i+1)2}^T (\mathbf{Q}_{i+1} + \mathbf{S}_{i+1}),\ i = nbd-1, \cdots, 0 \quad (15)$$

where $\mathbf{S}_{i+1}$ is defined in Eq.14. Since $\mathbf{Q}$ is an arbitrary vector in $\mathbf{R}^{nc}$, Equations 14 and 15 are valid for any vector $\mathbf{G}$ 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

$$\mathbf{g}_{i(i+1)} = \mathbf{B}_{i(i+1)2}^T (\mathbf{G}_{i+1} + \mathbf{S}_{i+1})$$
$$\mathbf{S}_n = \mathbf{0} \qquad i = nbd-1, \cdots, 0$$
$$\mathbf{S}_i = \mathbf{B}_{i(i+1)1}^T (\mathbf{G}_{i+1} + \mathbf{S}_{i+1}) \quad (16)$$

where $\mathbf{g}$ is the result of $\mathbf{B}^T\mathbf{G}$.

4. Implicit integration of the equations of motion and constraints

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

$$\delta \mathbf{q}^T \{\mathbf{B}^T (\mathbf{M}\dot{\mathbf{Y}} + \Phi_Z^T \lambda - \mathbf{Q})\} = \mathbf{0} \quad (17)$$

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 and is the number of cut constraints. $\Phi \in \mathbf{R}^m$ and Φ_Z represent the position-level constraint vector and the constraint Jacobian matrix, respectively. The mass matrix and the force vector $\mathbf{Q}$ are defined as follow;

$$\mathbf{M} = \mathrm{diag}(\mathbf{M}_1,\ \mathbf{M}_2, \cdots,\ \mathbf{M}_{nbd}) \quad (18.a)$$
$$\mathbf{Q} = (\mathbf{Q}_1^T,\ \mathbf{Q}_2^T, \cdots,\ \mathbf{Q}_{nbd}^T) \quad (18.b)$$

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

$$\mathbf{F} = \mathbf{B}^T (\mathbf{M}\dot{\mathbf{Y}} + \Phi_Z^T \lambda - \mathbf{Q}) = \mathbf{0} \quad (19)$$

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

$$\mathbf{F}(\mathbf{q},\ \mathbf{v},\ \dot{\mathbf{v}},\ \lambda) = \mathbf{0} \quad (20)$$

Successive differentiations of the position level constraint yield

$$\dot{\Phi}(\mathbf{q},\ \mathbf{v}) = \Phi_q \mathbf{v} - \mathbf{v} = \mathbf{0} \quad (21)$$

$$\ddot{\Phi}(\mathbf{q},\ \mathbf{v},\ \dot{\mathbf{v}}) = \Phi_q \dot{\mathbf{v}} - \gamma = \mathbf{0} \quad (22)$$

Equation 20 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 [21] as follows. ;

$$\mathbf{H}(\mathbf{p}) = \begin{bmatrix} \mathbf{F}(\mathbf{p}) \\ \ddot{\Phi} \\ \dot{\Phi} \\ \Phi \\ \mathbf{U}_0^T(\frac{h}{b_0}\mathbf{R}_1) \\ \mathbf{U}_0^T(\frac{h}{b_0}\mathbf{R}_2) \end{bmatrix} = \begin{bmatrix} \mathbf{F}(\mathbf{q},\ \mathbf{v},\ \dot{\mathbf{v}},\ \lambda) \\ \Phi_q \dot{\mathbf{v}} - \gamma \\ \Phi_q \mathbf{v} - \nu \\ \Phi(\mathbf{q}) \\ \mathbf{U}_0^T(\frac{h}{b_0}\dot{\mathbf{v}} - \mathbf{v} - \zeta_1) \\ \mathbf{U}_0^T(\frac{h}{b_0}\mathbf{v} - \mathbf{q} - \zeta_2) \end{bmatrix} = \mathbf{0} \quad (23)$$

where $\zeta_1 \equiv (1/b_0)\sum_{i=1}^{k} b_i \mathbf{v}_{(-i)}$ and $\zeta_2 \equiv (1/b_0)\sum_{i=1}^{k} b_i \mathbf{q}_{(-i)}$, in which k is the order of integration, b_is are the BDF coefficients and $\mathbf{p} \equiv [\mathbf{q}^T, \mathbf{v}^T, \dot{\mathbf{v}}^T, \lambda^T]^T$. 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:

$$\begin{bmatrix} \boldsymbol{\Phi}_q \\ \mathbf{U}_0^T \end{bmatrix} \tag{24}$$

The number of equations and the number of unknowns in Eq.23 are the same, and so Eq.23 can be solved for $\mathbf{p}$.

Newton Raphson method can be applied to obtain the solution $\mathbf{p}$.

$$\mathbf{H}_p \Delta \mathbf{p} = -\mathbf{H} \tag{25}$$

$$\mathbf{p}^{i+1} = \mathbf{p}^i + \Delta \mathbf{p} \tag{26}$$

where

$$\mathbf{H}_p = \begin{bmatrix} \mathbf{H}_q & \mathbf{F}_v & \mathbf{F}_v & \mathbf{F}_\lambda \\ \boldsymbol{\Phi}_q & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \dot{\boldsymbol{\Phi}}_q & \boldsymbol{\Phi}_v & \mathbf{0} & \mathbf{0} \\ \ddot{\boldsymbol{\Phi}}_q & \ddot{\boldsymbol{\Phi}}_v & \ddot{\boldsymbol{\Phi}}_v & \mathbf{0} \\ \mathbf{U}_0^T & \beta_0 \mathbf{U}_0^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_0^T & \beta_0 \mathbf{U}_0^T & \mathbf{0} \end{bmatrix} \tag{27}$$

Recursive formulas for $\mathbf{H}_p$ and $\mathbf{H}$ in Eq.25 will be derived in Section 4 to evaluate them efficiently. Equation 23 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.

5. Numerical algorithm

This section explains how the equations of motion and constraints presented in section 4 are implemented to reduce the computation time. The implementation algorithm is shown in Fig. 7. Note that Newton chord method in the position, velocity, and acceleration analyses uses the coefficient matrices that are evaluated and LU-decomposed at the initial time.

Since the proposed algorithm requires to compute only the residual vectors of the equations of motion and constraints, all computations are carried out in a vector oriented fashion by using the generalized recursive formulas presented in section 3. Consequently, computationally extensive matrix operation is unnecessary in the proposed algorithm. In order to further reduce the computation time, computation of the generalized forces and the velocity coupling terms are carried out only once right before the acceleration analysis and is excluded in the iteration loop for the acceleration analysis.

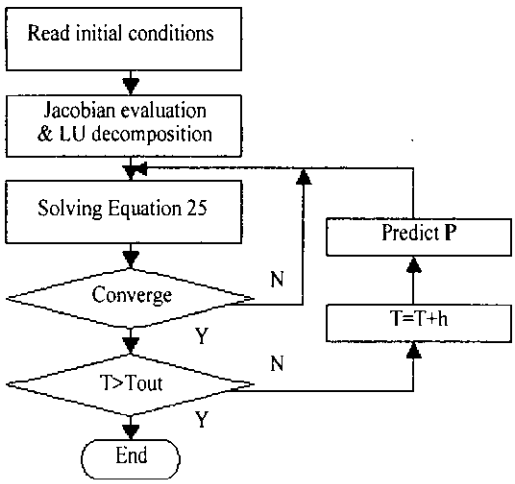

Fig. 7 Implementation algorithm

6. Numerical results

In the previous section, the equations of motion and kinematic constraints are formulated such that the coefficient matrices for the solution equations do not need to be frequently updated. Since a vehicle may be operated under various environments, it is very difficult to analytically

determine the update frequency of the coefficient matrices for a general vehicle system. As a consequence, this research performs several numerical experiments that cover some of the various environments as much as possible. Combination of the ground condition, driver inputs of steering, and acceleration or deceleration can cover some of the environments. Three types of simulation scenarios are considered. The first one is a J-turn simulation. The steering and acceleration pedal position inputs shown in Figs. 8 and 9 are given to the rack and pinion mechanism and the drive train model, respectively. The chassis is expected to experience some yaw, roll, and lateral acceleration. The second one is a lane change maneuver. The steering and pedal position inputs are shown in Figs. 9 and 10, respectively. The third one is a bump pass maneuver in which the suspension subsystems are expected to experience a rapid vertical motion. The pedal position shown in Fig. 9 is also used for this simulation.

In order to compare the results obtained from the proposed method, the DAE solution algorithm in Ref. [4] with a very small stepsize is executed to obtain the exact solutions. Fig. 12 shows the lateral accelerations of the chassis

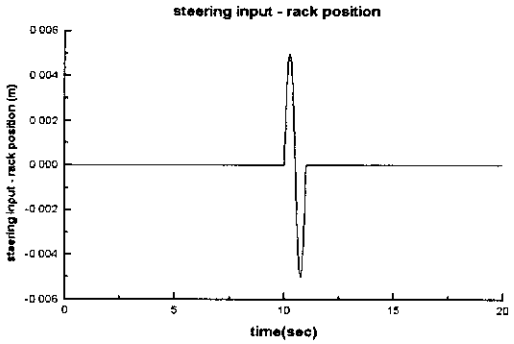

Fig. 10 Steering input for the lane change maneuver

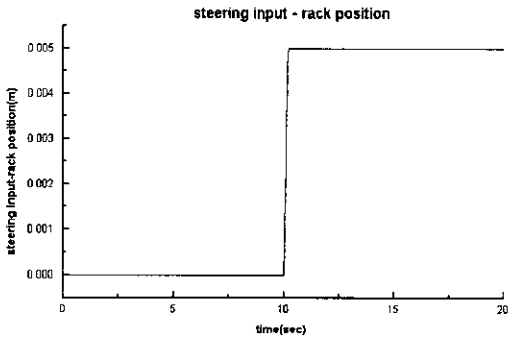

Figure 8. Steering input for the J-turn simulation

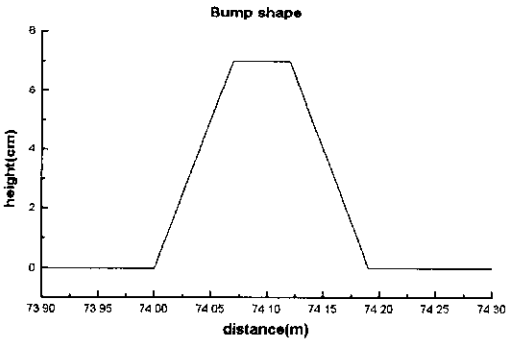

Fig. 11 Bump shape

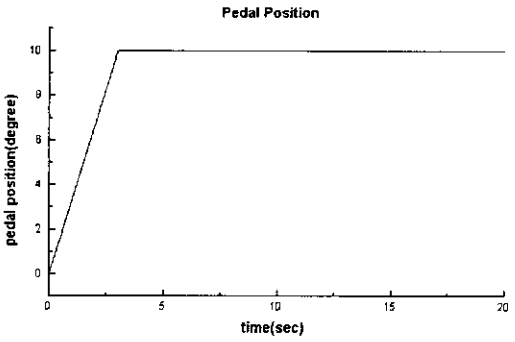

Fig. 9 Pedal position

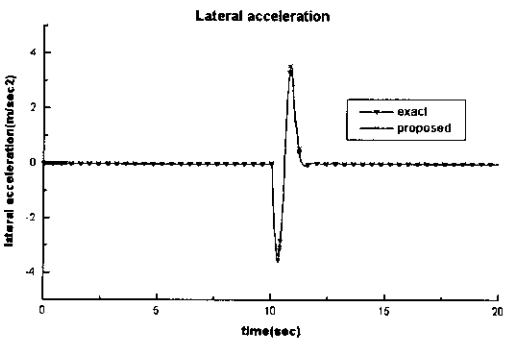

Fig. 12 Lateral acceleration for the lane change maneuver

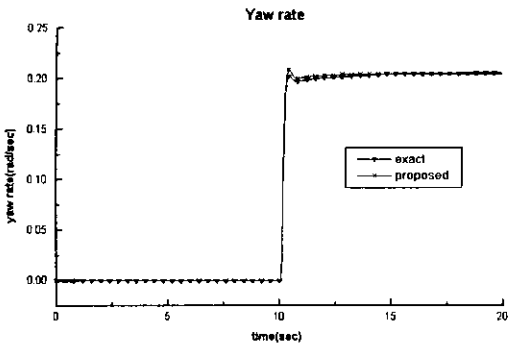

Fig. 13 Yaw rate for the J-turn simulation

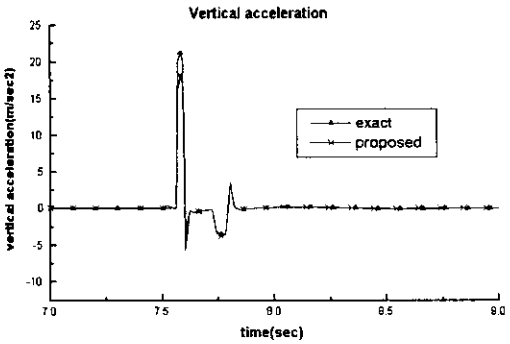

Fig. 14 Vertical acceleration for the bump pass simulation

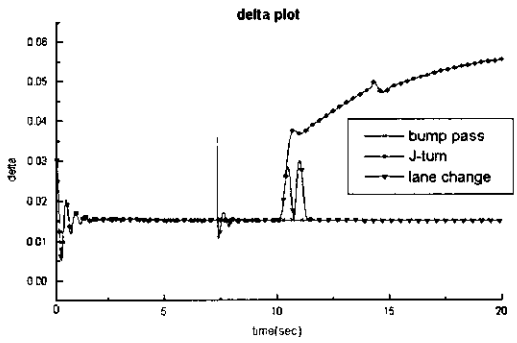

Fig. 15 Deltas for each simulation

Table 2 Computation time

methods	conventional	proposed
computation time per 1 step	12 msec	0.8 msec

for the lane change maneuver both from the proposed method and the original method. The yaw rates of the chassis for the J-turn simulation obtained from both methods are shown in Fig. 13. The vertical accelerations of the chassis for the bump pass simulation obtained from both methods are shown in Fig. 14. Note that both solutions closely match, which validates the results obtained from the proposed method. Computation times per each step for the conventional and proposed methods on a 400 MHz PC compatible are shown in Table 2. Note that a significant improvement in the computational efficiency has been achieved. Furthermore, since the 1 msec stepsize is used for the numerical integration, it is possible to run the prototype multibody vehicle model on a PC compatible in realtime by using the proposed algorithm.

The Δs in Eq. 4.10 for the position analysis are drawn in Fig. 15 for three simulations. Note that they remain to be very small throughout the simulations. The maximum number of Newton chord iterations was 4, which is possible due to the small Δ.

7. Conclusions

This research proposes an efficient implementation algorithm for implicit numerical integration methods so that low cost computers can be used for the realtime simulation of multibody vehicle dynamics models. The relative generalized coordinates and the generalized recursive formulas are used to reduce the size of the governing equations and computation time, respectively. Newton chord method is employed to solve the kinematic constraints and the linear system of equations of motion. The equations of motion and constraints are formulated on the body reference frame to make the mass matrix and the constraint Jacobian independent of the chassis motion. This made possible to compute the coefficient matrices of the solution equations only once for all Newton chord iterations. Convergence rate and order of the proposed Newton chord method are shown to be very close to these of the exact Newton method for the prototype vehicle model. Numerical simulations showed that the proposed method yields accurate solutions for various driving conditions in realtime on a 400 MHz PC com-

patible. The improvement of the proposed method over the conventional method for the prototype vehicle model was about 15 times.

Acknowledgements

This research was supported by the Hiqhway Research center, Hanyang University sponsored by KOSEF.

References

(1) J. Drosdol, and F. Panik, 1985, The Daimler-Benz Driving Simulator : A tool for Vehicle Development, SAE paper 850334.

(2) J. S. Freeman, G. Watson, Y. E. Papelis, T. C. Lin, A. Tayyab, R. A. Romano, and J. G. Kuhl, 1995, The Iowa Driving Simulator : An Implementation and Application Overview, SAE paper 950174.

(3) F. A. Potra, and W. C. Rheinboldt, 1989, On the Numerical Solution of Euler-Lagrange Equations, NATO Advanced Research Workshop on Real-Time Integration Methods for Mechanical System Simulation, Snowbird, Utah, U. S. A.

(4) J. Yen, E. J. Haug, and F. A. Potra, 1990, Numerical Method for Constrained Equations of Motion in Mechanical System Dynamics, Technical Report R-92, Center for Simulation and Design Optimization, Department of Mechanical Engineering, and Department of Mathematics, The University of Iowa, Iowa City, Iowa.

(5) D. J. Garcia, J. Unda, A. Avello, 1986, Natural Coordinates for the Computer Analysis of Multibody Systems, Computer Methods in Applied Mechanics and Engineering, Vol. 56, pp. 309~327.

(6) E. Bayo, D. J. Garcia, A. Avello, J. Cuadrado, 1991, An Efficient Computational Method for Real Time Multibody Dynamic Simulation in Fully Cartesian Coordinates, Computer Methods in Applied Mechanics and Engineering, Vol. 92, pp. 377~395.

(7) J. Wittenburg, 1997, Dynamics of Systems of Rigid Bodies, B. G. Teubner, Stuttgart.

(8) R. Schwertassek, Reduction of Multibody Simulation time by Appropriate Formulation of Dynamical System Equations, Computer-Aided Analysis of Rigid and Flexible Mechanical Systems, pp. 447~482, NATO ASI Series, Kluwer Academic Publishers.

(9) P. E. Nikravesh, G. Gim, Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops, J. of Mechanical Design, Vol. 115, No. 6, pp. 143~149.

(10) R. Featherstone, 1983, The Calculation of Robot Dynamics Using Articulated-Body Inertias, Int. J. Roboics Res., Vol. 2 : 13-30.

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

(12) K. Changizi and A. A. Shabana, A recursive Formulation for the Dynamic Analysis of Open Loop Deformable Multibody Systems, Journal of Applied Mechanics, Vol. 55, pp. 687~693.

(13) G. Gim, J. Krishnasamy, P. E. Nikravesh, 1990, An Analytical Model of Pneumatic Tires for Vehicle Dynamic Simulations, Int. J. of Vehicle Design, Vol. 11-12.

(14) D. Cho, J. K. Hedrick, 1998, Automotive Powertrain Modeling for Control, Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, , Vol. 111, pp. 568~576.

(15) C. T. Kelley, 1995, Iterative Methods for Linear and Nonlinear Equations, SIAM press.

Nomenclature

$\mathbf{A}_i$: orientation matrix of ith body

$\mathbf{r}_i$: position vector of ith body

$\dot{\mathbf{r}}_i$: velocity vector of ith body

$\dot{\mathbf{r}}'_i$: velocity vector of ith body in the body reference frame

$\mathbf{s}_{i(i-1)}$: local position vector of ith body

$\mathbf{d}_{(i-1)i}$: distance vector from (i-1) body to ith body

ω_i : angular velocity
ω'_i : body reference frame angular velocity
$\dot{\mathbf{q}}_{(i-1)i}$: relative velocity
$\mathbf{B}$: velocity transformation matrix
$\mathbf{Y}$: Cartesian velocities
δW : virtual work
$\mathbf{Q}_i$: generalized force of ith body
$\mathbf{S}_i$: joint reaction force of ith body
$\mathbf{M}$: mass matrix
$\dot{\mathbf{Y}}$: Cartesian accelerations
$\Phi_{\mathbf{z}}$: the Jacobian of position constraint w.r.t. $\mathbf{z}$
λ : Lagrange multipliers
$\mathbf{p}$: generalized coordinates
$\mathbf{v}$: generalized velocities
$\dot{\mathbf{v}}$: generalized accelerations
Φ : position constraint
$\dot{\Phi}$: velocity constraint
$\ddot{\Phi}$: acceleration constraint
$\mathbf{H}$: residual
$\mathbf{H}_{\mathbf{p}}$: the Jacobian of $\mathbf{H}$ w.r.t. $\mathbf{p}$