Application of Modified Local Parametrization Method for the Constrained Multibody Dynamic Systems

Dong-Chan, Lee*, Sang-Ho, Lee and Chang-Soo Han*****

(Received October 29, 1996)

This paper presents a corrector method for analyzing the dynamic behaviour of constrained multibody systems. For correcting the state variables, this method uses Lagrange-Newton method, which is a nonlinear programming technique. The Lagrange-Newton method uses the Lagrangian function that is a combined form of state variables with constraints, and the iteration formulation for convergence can be derived by the Newton-Raphson method. This algorithm does not update the Lagrange multipliers in the iteration formulation, for correcting the state variables, and is to project the state variables on the constraint manifold, in contrast to the previous local parametrization method. The validity of the algorithm and numerical solutions is verified through the convergence theorem denoting the convergence order of numerical solutions and the dynamic analysis of the full vehicle 3D model. The numerical solutions are compared with the ADAMS solutions

Key Words: Lagrange-Newton Method, Modified Local Parametrization Method, Lagrangian, Differential-Algebraic Equations, Constraint Manifold

Nomenclature

x : State variable
λ : Lagrange multiplier
f : Original objective function
c : Constraint function
L : Lagrangian function
W : Two times partial differentiation of original objective function
A : Transpose of Jacobian matrix of constraint
a : Constraint gradient vector (normal vector, $a(x)=\nabla c(x)$)
g : First partial differentiation of original objective function for state variable
$\alpha, \beta, \varepsilon$: Arbitrary constant
M : Mass matrix for multibody system
ϕ : Constraint equation for multibody system
$\dot{\phi}$: Velocity constraint equation for multibody system
ϕ_x : Jacobian matrix of ϕ
γ : Right hand side of acceleration constraint equation for multibody system
Q : External forces acting on the system
π : Sum of absolute value of each constraint

Superscripts

$*$: Optimum point
k : Iteration number
m : Number of constraints

Subscripts

i : i-th constraint

* Automotive Technology Lab., IAE
** Graduate Student, Department of Precision Mechanical Engineering, Seoul, Hanyang University
*** Professor Associate, Department of Mechanical Engineering, AnSan, Hanyang University

1. Introduction

In order to understand the method to derive the equations of motion of the multibody systems, four choices of coordinates are discussed. The first choice is to use a set of independent coordi-

nates, which determine the position of bodies with the least possible number of state variables. A minimal set of second-order differential equations, which is given in terms of system independent variables, is obtained in which the constraint conditions are absent. However, the rapidly growing complexity in the derivation as the number of variables increases, and the high degree of nonlinearity of the equations of motion make these coordinates difficult to implement in a general purpose computer program. The second choice is that of relative coordinates, which define the orientation of each moving body with respect to either an on-moving body or another adjacent moving body. For an open tree structure, the number of relative coordinates is equal to the number of independent coordinates. For a closed loop system, constraint equations are imposed via Lagrange multipliers. The third choice is that of natural coordinates, which define a body using two or more moving coordinates rigidly attached to it. These moving coordinates are located preferably at the joints of the mechanism, and can be shared by adjacent bodies. However, the presence of a fully populated mass matrix renders these coordinates less attractive in parallel computation. The last choice is that of Cartesian coordinates, which define the position of each particle in each individual body in the system with respect to an inertial reference frame. If independent coordinates are used, the equations of motion are generated in terms of system degrees of freedom expressed in differential equation form. But if the coordinates except the independent coordinate are used, the resulting equations for multibody systems are given by a set of second-order differential equations augmented with algebraic constraint equations.

For solving the differential-algebraic equations, many methods are presented. Gear and Petzold(1982, 1984) presented the method augmenting the second-order governing equilibrium equations with twice time-differentiated constraint equations. However, numerical integration algorithms provide only an approximate solution. An approach to stabilize the constraint violations was proposed by Baumgarte(1972). But for some systems, this method suffers from ill-conditioning in the solution for Lagrange multipliers. A different approach to avoid constraint violations is to identify system dependent and independent variables from the given constraint Jacobian matrix. These algorithms include: the generalized coordinate partitioning scheme(Wehage, R. A. and Haug, E. J., 1982), the singular value decomposition(Walton, W. C. and Steeves, E. C., 1969; Singh, R. P., and Likins, P. W., 1985), the natural coordinates partitioning scheme(Garcia de Jalon, J. et al. 1987 ; Unda, J. et al. 1987), the null space scheme(Liang, C. G. et al. 1987; Ider, S. K. et al. 1988; Amirouche, F. M. L. et al. 1988).

Recently, the solution methods introduce the optimization techniques. These algorithms include: the augmented Lagrangian formulation (Bayo, E. and Avello, A., 1993), the penalty formulation method(Kurdila, A. J. and Narcowich F. J., 1993) and the local parametrization method(Potra, F. A., 1991). The augmented Lagrange formulation and penalty formulation have difficulty in finding the penalty parameters. The local parametrization method is the optimization technique that displacements and velocities integrated from the accelerations are projected on the tangent plane of the constraint manifold, and thus satisfy the constraint conditions. This method introduces the Lagrange multipliers in analysis steps and the numerical solutions are controlled by updating the Lagrange multipliers. The augmented Lagrangian formulation and penalty formulation methods are indirect in attempting to solve nonlinear constraint problems. A more direct and efficient approach is to iterate on the basis of certain approximations of the problem functions, in particular by using linear approximations of the constraint functions.

This paper presents the modified local parametrization method. The previous local parametrization method(Potra, F. A., 1991; Lee, S. H., Bae, D. S., Han, C. S., and Suh, M. S., 1994) is to update the Lagrange multipliers in the iteration formulations derived from the Lagrangian functions, for correcting displacement, velocity and acceleration values. The Lagrangian functions are a combined form of state variables with

constraints. The presented algorithm in this paper does not update the Lagrange multiplier in the iteration formulations, for correcting state variables. Instead it projects the state variables on the constraint manifold and checks if the admissible displacements and velocities that satisfy the constraint conditions exist. We can calculate the well-conditioned accelerations from the general DAE (Differential-Algebraic Equations). Thus, the number of the right hand side terms in the correction equations of state variables which need to the computed decreases and even though the update of Lagrange multipliers is not in the iteration formulations for correcting state variables, the numerical efficiency of solutions is not influenced. The validity of algorithm and unnecessity of the update of Lagrange multipliers are verified through the convergence theorem denoting the convergence order of numerical solutions. The numerical solutions of this algorithm are verified through the simulation of full vehicle 3D model and comparison with the numerical solutions of ADAMS.

2. Verification of Algorithm

2.1 Optimization scheme

A more direct and efficient approach to solve nonlinear constraint problems is to iterate on the basis of some approximations to the problem $\boldsymbol{f}(\boldsymbol{x})$ and, $\boldsymbol{c}(\boldsymbol{x})$, in particular by using linear approximations to the constraint function $\boldsymbol{c}(\boldsymbol{x})$. The solving method of nonlinear constraint problems is most simply explained as being Newton's method applied to find the stationary point of the Lagrangian function Eq. (1), and hence might be referred to as the Lagrange-Newton method.

$$\boldsymbol{L}(\boldsymbol{x}, \lambda) = \boldsymbol{f}(\boldsymbol{x}) + \sum_i \lambda_i \boldsymbol{c}_i(\boldsymbol{x}), \quad (i = 1, 2, \cdots\cdots m) \tag{1}$$

Let's apply the stationary point condition to the equality constraint problem Eq. (1) and define ∇ as in Eq. (3) so that equation Eq. (2) is the stationary point condition at, $\boldsymbol{x}^*$, λ^*.

$$\nabla \boldsymbol{L}(\boldsymbol{x}^*, \lambda^*) = \boldsymbol{0} \tag{2}$$

where, $\nabla = \begin{pmatrix} \nabla_x \\ \nabla_\lambda \end{pmatrix}$ (3)

A Taylor series for $\nabla \boldsymbol{L}$ about $\boldsymbol{x}^{(k)}$, $\lambda^{(k)}$ gives

$$\nabla \boldsymbol{L}(\boldsymbol{x}^{(k)} + \delta\boldsymbol{x}, \lambda^{(k)} + \delta\lambda) = \nabla \boldsymbol{L}^{(k)} + \left[\nabla^2 \boldsymbol{L}^{(k)}\right]\begin{pmatrix} \delta\boldsymbol{x} \\ \delta\lambda \end{pmatrix} + \cdots\cdots \tag{4}$$

where, $\nabla \boldsymbol{L}^{(k)} = \nabla \boldsymbol{L}(\boldsymbol{x}^{(k)}, \lambda^{(k)})$. Neglecting higher order terms and setting the left-hand side to zero by virtue of Eq. (2) gives the iteration

$$[\nabla^2 \boldsymbol{L}^{(k)}]\begin{pmatrix} \delta\boldsymbol{x} \\ \delta\lambda \end{pmatrix} = -\nabla \boldsymbol{L}^{(k)} \tag{5}$$

This is solved to give corrections $\delta\boldsymbol{x}$ and $\delta\lambda$. Formulae for $\nabla \boldsymbol{L}$ and $\nabla^2 \boldsymbol{L}$ are readily obtained from Eq. (1), giving the system

$$\begin{bmatrix} \boldsymbol{W}^{(k)} & \boldsymbol{A}^{(k)} \\ \boldsymbol{A}^{(k)T} & \boldsymbol{0} \end{bmatrix}\begin{pmatrix} \delta\boldsymbol{x} \\ \delta\lambda \end{pmatrix} = -\begin{pmatrix} \boldsymbol{g}^{(k)} + \boldsymbol{A}^{(k)}\lambda^{(k)} \\ \boldsymbol{c}^{(k)} \end{pmatrix} \tag{6}$$

$\boldsymbol{A}^{(k)}$ is the transpose of Jacobian matrix of constraint evaluated at $\boldsymbol{x}^{(k)}$,

$$\boldsymbol{W}^{(k)} = \nabla^2 \boldsymbol{f}(\boldsymbol{x}^{(k)}) + \sum_i \lambda_i^{(k)} \nabla^2 \boldsymbol{c}_i(\boldsymbol{x}^{(k)}) \tag{7}$$

is the Hessian matrix $\nabla_x^2 \boldsymbol{L}(\boldsymbol{x}^{(k)}, \lambda^{(k)})$, and

$$\boldsymbol{g}^{(k)} = \nabla \boldsymbol{f}(\boldsymbol{x}^{(k)}) \tag{8}$$

is the Jacobian of for $\boldsymbol{f}(\boldsymbol{x})$ for $\boldsymbol{x}^{(k)}$.

It is more convenient to write $\lambda^{(k+1)} = \lambda^{(k)} + \delta\lambda$ and to solve the equivalent system

$$\begin{bmatrix} \boldsymbol{W}^{(k)} & \boldsymbol{A}^{(k)} \\ \boldsymbol{A}^{(k)T} & \boldsymbol{0} \end{bmatrix}\begin{pmatrix} \delta\boldsymbol{x}^{(k)} \\ \lambda^{(k+1)} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{g}^{(k)} \\ \boldsymbol{c}^{(k)} \end{pmatrix} \tag{9}$$

to determine $\delta\boldsymbol{x}^{(k)}$ and $\lambda^{(k+1)}$. Then $\boldsymbol{x}^{(k+1)}$ is given by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \delta\boldsymbol{x}^{(k)} \tag{10}$$

An important feature of the method is that ultimately the convergence is of second order. If the second order sufficient conditions for the equality constraint problem hold at $\boldsymbol{x}^*$, λ^*, and if rank of $\boldsymbol{A}^* = m$, then the Lagrangian matrix

$$\nabla^2 \boldsymbol{L}^* = \begin{bmatrix} \boldsymbol{W}^* & \boldsymbol{A}^* \\ \boldsymbol{A}^{*T} & \boldsymbol{0} \end{bmatrix} \tag{11}$$

is non-singular. The second order convergence of iteration Eqs. (9) and (10) follows by virtue of the convergence theorem applied to the system of equations $\nabla \boldsymbol{L}(\boldsymbol{x}, \lambda) = \boldsymbol{0}$. This requires both $\boldsymbol{x}^{(k)}$ and $\lambda^{(k)}$ to be sufficiently close to $\boldsymbol{x}^*$ and λ^* for some k.

2.1.1 Convergence theorem

If $\boldsymbol{x}^{(1)}$ is sufficiently close to $\boldsymbol{x}^*$, and the Lagrangian matrix

$$\begin{bmatrix} \boldsymbol{W}^{(1)} & \boldsymbol{A}^{(1)} \\ \boldsymbol{A}^{(1)T} & \boldsymbol{0} \end{bmatrix}$$

is non-singular, and if the second order sufficient conditions hold at $\boldsymbol{x}^*$, λ^* with rank of $\boldsymbol{A}^* = m$, then the Lagrange-Newton iteration Eqs. (9) and (10) converges and it is of second order.

2.1.2 Proof

Define errors $\boldsymbol{h}^{(k)} = \boldsymbol{x}^* - \boldsymbol{x}^{(k)}$ and $\Delta^{(k)} = \lambda^* - \lambda^{(k)}$. The Taylor series of $\boldsymbol{g}^*$, $\boldsymbol{c}^*$, and $\boldsymbol{a}^*$ about $\boldsymbol{x}^{(k)}$

$$\begin{aligned} \boldsymbol{c}^* &= \boldsymbol{c}^{(k)} + \boldsymbol{A}^{(k)T}\boldsymbol{h}^{(k)} + \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \\ \boldsymbol{g}^* &= \boldsymbol{g}^{(k)} + \nabla^2 \boldsymbol{f}^{(k)}\boldsymbol{h}^{(k)} + \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \\ \boldsymbol{a}^* &= \boldsymbol{a}^{(k)} + \nabla^2 \boldsymbol{c}_i^{(k)}\boldsymbol{h}^{(k)} + \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \\ &(i = 1, 2, \cdots\cdots, m) \end{aligned} \tag{12}$$

are valid. It follows from Eq. (9) that $\boldsymbol{h}^{(k+1)}$ and $\Delta^{(k+1)}$ satisfy the equations

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{W}^{(k)} & \boldsymbol{A}^{(k)} \\ \boldsymbol{A}^{(k)T} & \boldsymbol{0} \end{bmatrix} \begin{pmatrix} \boldsymbol{h}^{(k+1)} \\ \Delta^{(k+1)} \end{pmatrix} \\ &= -\begin{pmatrix} \sum_i \Delta_i^{(k)} \nabla^2 \boldsymbol{c}_i^{(k)} \boldsymbol{h}^{(k)} + \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \\ \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \end{pmatrix} \\ &= -\begin{pmatrix} \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) + \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2 \|\Delta^{(k)}\|) \\ \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \end{pmatrix} \end{aligned} \tag{13}$$

At $\boldsymbol{x}^*$ and λ^*, the Lagrangian matrix is non-singular so for $\boldsymbol{x}^{(k)}$ and $\lambda^{(k)}$ in some neighbourhood of $\boldsymbol{x}^*$ and λ^*,

$$\begin{pmatrix} \boldsymbol{h}^{(k+1)} \\ \Delta^{(k+1)} \end{pmatrix} = \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) + \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2 \|\Delta^{(k)}\|)$$

There exists a constant $\varepsilon > 0$ such that

$$\max(\|\boldsymbol{h}^{(k+1)}\|, \|\Delta^{(k+1)}\|) \le \varepsilon \|\boldsymbol{h}^{(k)}\| \cdot \max(\|\boldsymbol{h}^{(k)}\|, \|\Delta^{(k)}\|). \tag{14}$$

Thus, in a smaller neighbourhood, i.e., if $1 > \varepsilon$, $\max(\|\boldsymbol{h}^{(k)}\|, \|\Delta^{(k)}\|) = \alpha$, then

$$\max(\|\boldsymbol{h}^{(k+1)}\|, \|\Delta^{(k+1)}\|) \le \alpha \|\boldsymbol{h}^{(k)}\| \le \alpha \cdot \max(\|\boldsymbol{h}^{(k)}\|, \|\Delta^{(k)}\|).$$

So the iteration converges and the order is seen to be quadratic from Eq. (14).

Let only $\boldsymbol{x}^{(1)}$ be in a neighbourhood of $\boldsymbol{x}^*$, so that $\boldsymbol{A}^{(1)}$ has full rank, and let $\lambda^{(1)}$ be such that the Lagrangian matrix is non-singular. Then $\|\Delta^{(1)}\| \ge \|\boldsymbol{h}^{(1)}\|$ and so as above, there exists a constant, β, such that

$$\max(\|\boldsymbol{h}^{(2)}\|, \|\Delta^{(2)}\|) \le \beta \cdot \|\boldsymbol{h}^{(1)}\| \|\Delta^{(1)}\|$$

If $\boldsymbol{x}^{(1)}$ is sufficiently close to $\boldsymbol{x}^*$ in that $\|\boldsymbol{h}^{(1)}\| < \frac{1}{\varepsilon\beta\|\Delta^{(1)}\|}$, then $\max(\|\boldsymbol{h}^{(2)}\|, \|\Delta^{(2)}\|) < \frac{1}{\varepsilon}$ and so $\boldsymbol{x}^{(2)}$, $\lambda^{(2)}$ is in the neighbourhood for which convergence occurs.

2.2 Comparison of convergence rate with the previous algorithm

In order to compare the validity of numerical solutions with the previous algorithm that does update the Lagrange multipliers, this paper checks the convergence order. The right hand side of Eq. (13) shows that the convergence order of the presented algorithm is quadratic. The convergence order of previous algorithm can be derived from Eq. (6). It follows from Eq. (6) that $\boldsymbol{h}^{(k+1)}$, $\Delta^{(k+1)}$ satisfy the equations as follows;

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{W}^{(k)} & \boldsymbol{A}^{(k)} \\ \boldsymbol{A}^{(k)T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}^{(k+1)} \\ \Delta^{(k+1)} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{g}^* + \boldsymbol{A}^*\lambda^* - \sum_i \Delta_i^{(k)} \nabla^2 \boldsymbol{c}_i^{(k)} \cdot \boldsymbol{h}^{(k)} - \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \\ \boldsymbol{c}^{(k)} + \boldsymbol{A}^{(k)T}\boldsymbol{h}^{(k)} \end{bmatrix} \end{aligned}$$

where,

$$\begin{aligned} \boldsymbol{g}^* + \boldsymbol{A}^*\lambda^* = &(\boldsymbol{g}^{(k)} + \nabla^2 \boldsymbol{f}(\boldsymbol{x}^{(k)}) \cdot \boldsymbol{h}^{(k)}) \\ &+ (\boldsymbol{A}^{(k)} + \sum_i \nabla^2 \boldsymbol{c}_i^{(k)} \cdot \boldsymbol{h}^{(k)}) \cdot \lambda^* \\ &+ \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \end{aligned}$$

$\boldsymbol{g}^*$, $\boldsymbol{A}^*$ and λ^* of the right hand side in the above equation are $\boldsymbol{g}(\boldsymbol{x}^*)$, $\boldsymbol{A}(\boldsymbol{x}^*)$ and λ for the optimum point $\boldsymbol{x}^*$. Thus $\boldsymbol{g}^* + \boldsymbol{A}^*\lambda^*$ is $\nabla \boldsymbol{L}(\boldsymbol{x}^*, \lambda^*)$ and is zero by the stationary point condition. The above equation is changed by using Eq. (12) as follows;

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{W}^{(k)} & \boldsymbol{A}^{(k)} \\ \boldsymbol{A}^{(k)T} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{h}^{(k+1)} \\ \Delta^{(k+1)} \end{bmatrix} \\ &= -\begin{bmatrix} \sum_i \Delta_i^{(k)} \nabla^2 \boldsymbol{c}_i^{(k)} \cdot \boldsymbol{h}^{(k)} - \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \\ \boldsymbol{O}(\|\boldsymbol{h}^{(k)}\|^2) \end{bmatrix} \end{aligned}$$

The convergence order of the local parametrization method that does not update the Lagrange multipliers is similar to that of the presented algorithm. Thus, it is shown that even though the Lagrange multipliers are not updated, the numeri-

cal efficiency of solutions has no influence.

3. Numerical Analysis

Since the numerical solutions integrated by a mixed form of differential-algebraic equation, generally violate the constraint conditions, the numerical solutions must be corrected to displacements and velocities consistent with the constraint conditions. Thus, the following numerical analysis procedure is presented for correcting the state variables of multibody dynamic system.

3.1 Position analysis

Generally, the integrated generalized coordinates, $\boldsymbol{x}^0$, do not satisfy the constraint conditions. The fact that $\boldsymbol{x}^0$ does not satisfy the constraint conditions means the physical separation of mechanical system, and is the mathematical phenomenon different from the reality. In this paper, the initial guess $\boldsymbol{x}^0$ is projected on the constraint manifold by the Lagrange-Newton method.

The structure of the following constrained optimization problem is considered for the violated generalized coordinates.

$$\textbf{Minimize } \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^0)^T\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{x}^0), \quad \boldsymbol{x}\in\boldsymbol{R}^n$$
$$\textbf{Subject to } \phi(\boldsymbol{x}, t)=\boldsymbol{0}, \quad \phi\in\boldsymbol{R}^m \tag{15}$$

Where, $\boldsymbol{x}^0$ is a set of initial integrated displacements, $\boldsymbol{x}$ is a set of corrected displacements, $\boldsymbol{M}$ is the (n×n) mass matrix, and $\phi(\boldsymbol{x}, t)$ is a set of constraint equations.

We can make the Lagrangian function by introducing the Lagrange multipliers as follows:

$$\boldsymbol{L}(\boldsymbol{x}, \lambda_{pos})=\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^0)^T\boldsymbol{M}(\boldsymbol{x}-\boldsymbol{x}^0) + \phi^T\lambda_{pos} \tag{16}$$

By Eq. (9), the following equation is formulated for correcting the displacement coordinates.

$$\begin{bmatrix}\boldsymbol{M} & \phi_x^T\\ \phi_x & \boldsymbol{0}\end{bmatrix}^{(k)}\begin{pmatrix}\delta\boldsymbol{x}^{(k)}\\ \lambda_{pos}^{(k+1)}\end{pmatrix}=-\begin{pmatrix}\boldsymbol{M}(\boldsymbol{x}^{(k)}-\boldsymbol{x}^0)\\ \phi^{(k)}\end{pmatrix} \tag{17}$$

In the previous local parametrization, the correction equation of position is as follows:

$$\begin{bmatrix}\boldsymbol{M} & \phi_x^T\\ \phi_x & \boldsymbol{0}\end{bmatrix}^{(k)}\begin{pmatrix}\delta\boldsymbol{x}\\ \delta\lambda_{pos}\end{pmatrix}^{(k)}$$
$$=-\begin{pmatrix}\boldsymbol{M}(\boldsymbol{x}^{(k)}-\boldsymbol{x}^0)+(\phi_x^T)^{(k)}\lambda_{pos}^{(k)}\\ \phi^{(k)}\end{pmatrix} \tag{17-a}$$

From the above equation, the term of $(\phi_x^T)^{(k)}\lambda_{pos}^{(k)}$ is not concerned to the convergence order of numerical solutions by from Eq. (12) to Eq. (14). Thus, the abbreviated right hand side term is introduced in the right hand side of Eq. (17).

Error value of constraint equations is calculated by Eq. (18).

$$\pi=\sum_{i=1}^{m}|\phi_i| \tag{18}$$

The generalized displacement coordinates are calculated by Eq. (10).

3.2 Velocity analysis

As the same method with the position analysis, the structure of the following constrained optimization problem is considered for the violated generalized velocities.

$$\textbf{Minimize } \frac{1}{2}(\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}^0)^T\boldsymbol{M}(\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}^0), \quad \dot{\boldsymbol{x}}\in\boldsymbol{R}^n$$
$$\textbf{Subject to } \dot{\phi}(\dot{\boldsymbol{x}}, \boldsymbol{x}, t)=\boldsymbol{0}, \quad \dot{\phi}\in\boldsymbol{R}^m \tag{19}$$

Where, $\dot{\boldsymbol{x}}^0$ is a set of initial integrated velocities, $\dot{\boldsymbol{x}}$ is a set of corrected velocities, $\boldsymbol{M}$ is the (n×n) mass matrix, and $\dot{\phi}(\dot{\boldsymbol{x}}, \boldsymbol{x}, t)$ is a set of velocity constraint equations.

We can make the Lagrangian function by introducing the Lagrange multipliers as follows:

$$\boldsymbol{L}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \lambda_{vel})=\frac{1}{2}(\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}^0)^T\boldsymbol{M}(\dot{\boldsymbol{x}}-\dot{\boldsymbol{x}}^0) + \dot{\phi}^T\lambda_{vel} \tag{20}$$

By Eq. (9), the following equation is formulated for controlling the velocity coordinates.

$$\begin{bmatrix}\boldsymbol{M} & \phi_x^T\\ \phi_x & \boldsymbol{0}\end{bmatrix}^{(k)}\begin{pmatrix}\delta\dot{\boldsymbol{x}}^{(k)}\\ \lambda_{vel}^{(k+1)}\end{pmatrix}=-\begin{pmatrix}\boldsymbol{M}(\dot{\boldsymbol{x}}^{(k)}-\dot{\boldsymbol{x}}^0)\\ \dot{\phi}^{(k)}\end{pmatrix} \tag{21}$$

In the previous local parametrization, the correction equation of velocity is as follows:

$$\begin{bmatrix}\boldsymbol{M} & \phi_x^T\\ \phi_x & \boldsymbol{0}\end{bmatrix}^{(k)}\begin{pmatrix}\delta\dot{\boldsymbol{x}}\\ \delta\lambda_{vel}\end{pmatrix}^{(k)}$$
$$=-\begin{pmatrix}\boldsymbol{M}(\dot{\boldsymbol{x}}^{(k)}-\dot{\boldsymbol{x}}^0)+(\phi_x^T)^{(k)}\lambda_{vel}^{(k)}\\ \dot{\phi}^{(k)}\end{pmatrix} \tag{21-a}$$

But, from the above equation, the term of $(\phi_x^T)^{(k)}\lambda_{vel}^{(k)}$ is not concerned to the convergence order of numerical solutions by from Eq. (12) to Eq. (14).

Thus the abbreviated right hand side term is introduced in the right hand side of Eq. (21)

Error value of constraint equations is calculated by Eq. (22)

$$\pi=\sum_{i=1}^{m}|\dot{\phi}_i| \tag{22}$$

The generalized velocities are calculated by Eq. (10)

3.3 Acceleration analysis

If the admissible displacements and velocities that satisfy the constraint conditions exist, we can calculate the well-conditioned accelerations from the general differential-algebraic equations. The general differential-algebraic equations are a combined form of the Lagrange multiplier form of the equations of motions and algebraic constraint equations. That is as follows:

$$\begin{bmatrix} \boldsymbol{M} & \phi_x^T \\ \phi_x & \boldsymbol{0} \end{bmatrix}\begin{pmatrix} \ddot{\boldsymbol{x}} \\ \lambda_{acc} \end{pmatrix}=\begin{pmatrix} \boldsymbol{Q} \\ \gamma \end{pmatrix} \tag{23}$$

where $\boldsymbol{Q}$ is the external forces acting on the systems, and γ is defined from the twice differentiation of constraint equations.

$$\phi_x\ddot{\boldsymbol{x}}=-(\phi_x\dot{\boldsymbol{x}})_x\dot{\boldsymbol{x}}-2\phi_{xt}\dot{\boldsymbol{x}}-\phi_{tt}\equiv\gamma$$

4. Simulation

4.1 Computer program

In order to verify the efficiency of the numerical solutions by the presented algorithm, Fortran language is used for coding initial condition, inertial values, joint type, and solving method. This program is based on the previous program and is to update the previous method (Lee, S. H., et al., 1994). And the integrator is DE which is made by Shampine and Gordon (Shampine, L. F., and Gordon, M. K., 1975)

4.2 Vehicle modeling

The validity of multibody dynamic analysis algorithm developed in this paper, is verified through the full vehicle model analyses and comparison between the results of the presented algorithm and ADAMS. Joints in the suspension of the real vehicle system are interconnected through bushing, but in this analysis, bushing isn't used.

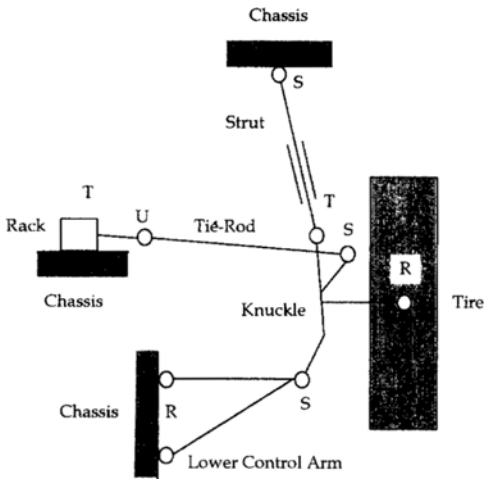


Fig. 1 Front suspension.

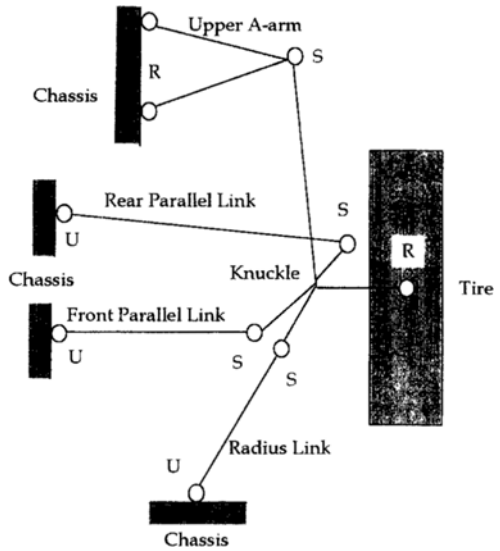


Fig. 2 Rear suspension.

The overall vehicle model in this paper uses McPherson strut type in the front wheels, and multilink independent suspension in the rear wheels. The total degrees of freedom of vehicle is 11 D.O.F ; 6 D.O.F in the chassis, vertical strokes, and steering system.

McPherson strut suspension consists of lower

Table 1 The kinematic model of a vehicle model.

Bodies	ngc=18×6=108
Eighteen bodies	
Constraints	
Translational joints	3×5=15
Revolute joints	4×5=20
Spherical joints	14×3=42
Universal joints	5×4=20
Distance constraints	1
	ncn=98
D.O.F=108−98=10	

Note) ngc : Number of generalized coordinates
ncn : Number of Constraints

control arm, knuckle, strut and tie rod as shown in Fig. 1. Rigid bodies are interconnected by joints denoted in Fig. 1, where, the mark S denotes spherical joint, R denotes revolute joint, T denotes translational joint, and U denotes universal joint. Multilink independent suspension consists of upper control arm, two lower parallel links in the lateral direction and radius link in the longitudinal direction. Figure 2 shows an intermediate type between double wishbone type and classical multilink type.

The kinematic model of a vehicle model is described at Table 1.

4.3 J-turn simulation

Figure 3 shows the full vehicle model with the previously described front and rear suspensions. This model is simulated by the proposed algorithm in this paper and the results are compared with the ADAMS solutions.

The riding simulation for evaluating the steering performances is a step steering of the full vehicle model. The vehicle runs at the 80 km/h and the 45° steering condition inputs as the steering input during 0.2 second from 2.0 second. Before the step steering simulation, the initial input positions use the value calculated from the equilibrium analysis.

4.4 Simulation remark

Figure 4 is the results of simulation for the step steering. (a), (b), and (c) in the Fig. 4 denote the

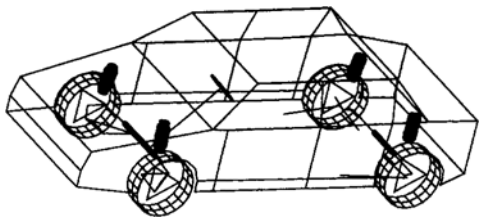
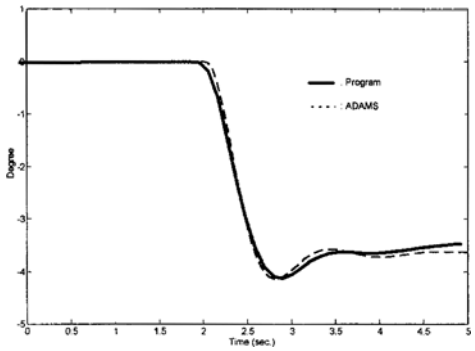
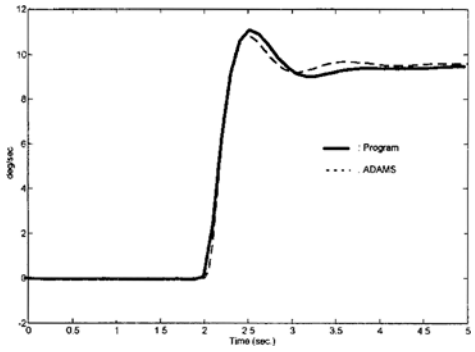


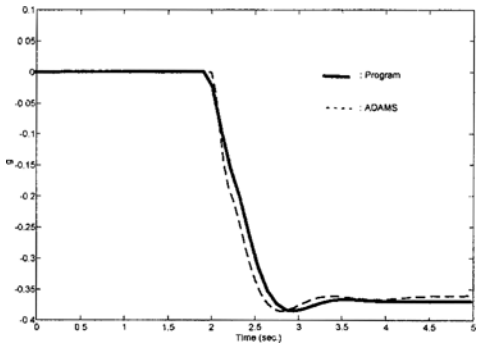
Fig. 3 Full vehicle model.



(a) Chassis roll angle



(b) Chassis yaw rate



(c) Chassis lateral acceleration

Fig. 4 Simulation results.

roll angle, yaw rate, and lateral acceleration, respectively. The solid and dashed lines denote the ADAMS results and the results of the proposed algorithm. The results of the previous local parameter method are not prerented, because they are identical to the ones with the proposed algorithm. The overall trends are consistent with the results of ADAMS, but the small difference occurs, in the transient region, which is caused by the differernt tire models. Tire model used in this program has a bilinear property and tire model of ADAMS is a UA tire model.

The number of arithmetic operation for additional term of Eqs. (17-a) and (21-a) is described at Table 2. In Eqs. (17) and (21), the total operational number decreases by 21580, in position and velocity analyses of every each iteration, as shown in Table 2.

5. Conclusion

In this paper, the modified local parametrization method is presented for the numerical analyses of the constrained multibody dynamic systems. For correcting the state variables, the Lagrangian function is derived from the state variables and constraint equations. The correction equations of state variables are derived by the Lagrange-Newton method, The presented algorithm does not update the Lagrange multipliers in the iteration formulations. Thus, the right hand side term of the correcting equations is abbreviated, in comparison with the previous local parameter method. Because the Lagrange multipliers is not updated in the iteration formulations, the computational burden in correcting the state variables can be reduced for each iteration in each position and velocity analyses. The convergence rate comparison between the abbreviated corrector algorithm and the previous algorithm is verified through the convergence theorem denoting the convergence order of numerical solutions. By using these well-conditioned state variables, the general differential-algebraic equations can directly be used for computing the accelerations.

To show the validity of this algorithm, the full vehicle model is simulated using the computer program and its results are compared with ADAMS results.

References

ADAMS Reference Manual, Mechanical Dynamics, 2301 Commonwealth Blvd, Ann Arbor, MI 48105.

Amirouche, F. M. L., Jyia, T., and Idler, S. K., 1988, "A Recursive Householder Transformation for Complex Dynamical Systems with Constraints," *Journal of Applied Mechanics*, Vol. 55, No. 3, pp. 729~734.

Table 2 Number of arithmetic operations for additional term of Eq. (17-a) and (21-a).

Additional term	Equation no. in paper	Number of arithmetic operation
$(\phi_x^T)^{(k)}\lambda_{pos}^{(k)}$	Eq. (17-a)	(ncn+1)×ngc =10692(in this example)
$\delta\lambda_{pos}$	Eq. (17-a)	ncn =98(in this example)
$(\phi_x^T)^{(k)}\lambda_{vel}^{(k)}$	Eq. (21-a)	(ncn+1)×ngc =10692(in this example)
$\delta\lambda_{vel}$	Eq. (21-a)	ncn =98(in this example)
Total operational number : $2\times[(ncn+1)\times ngc]+2\times ncn$ =21580(in this example)		

Note) ngc : Number of generalized coordinates
ncn : Number of Constraints

Baumgarte, J. W., 1972, "Stabilization of Constraints and Integrals of Motion," *Computer Methods in Applied Mechanics and Engineering*, Vol. 1, pp. 1~16.

Baumgarte, J. W., 1983, "A New Method of Stabilization for Holonomic Constraints," *Journal of Applied Mechanics*, Vol. 50, pp. 869~870.

Bayo, E. and Avello, A., 1993, "Singularity Free Augmented Lagrangian Algorithms for Constraint Multibody Dynamics," *Journal of Nonlinear Dynamics*.

Fletcher R., 1989, *Practical Methods of Optimization*, John Wiley & Sons.

Haug, E. J., 1989, *Computer Aided Kinematics and Dynamics, Vol 1 : Basic Methods*, Allyn and Bacon, Boston.

Ider, S. K., and Amirouche, F. M. L., 1988, "Coordinate Reduction in the Dynamics of Constrained Multibody Systems-A New Approach," *Journal of Applied Mechanics*, Vol. 55, pp. 899~904.

Kurdila, A. J. and Narcowich F. J., 1993, "Sufficient Conditions for Penalty Formulation Methods in Analytical Dynamics," *Computational Mechanics*.

Lee, S. H., Bae, D. S., Han, C. S., and Suh, M. S., 1994, "Dynamic Analysis for Constrained Multibody Systems," *KSME*, Vol. 18, No. 9, pp. 2339~2348.(in Korean)

Liang, C. G., and Lance, G. M., 1987, "A Differentiable Null Space Method for Constrained Dyanmic Analysis," *J. Mech. Tran. and Auto. in Design*, Vol. 109, pp. 405~411.

Garcia de Jalon, J., Unda, J., Avello, A., and Jimenez, J. M., 1987, "Dynamic Analysis of Three-dimensional Mechanisms in Natural Coordinates," *ASME J. Mech. Tran. and Auto. In Design*, Vol. 109, pp. 460~465.

Gear, C. W. And Petzold, L. R., 1984, "ODE Methods for the Solution of Differential/Algebraic Systems," *SIAM Journal of Numerical Analysis*, Vol. 21, No. 4, pp. 716~728.

Orlanda, N., M. A. Chace, and D. A. Calahan, 1977, "A Sparsity-Oriented Approach to Dynamic Analysis and Design of Mechanical Systems-Part 1 and 2," *ASME J. Engr.*, Vol. 99, pp. 773~784.

Parviz E. Nikravesh, 1988, *Computer-Aided Analysis of mechanical systems*, Prentice-Hall

Petzold, L. R., 1982, "Differential/Algebraic Equations Are Not ODE's," *SIAM Journal on Scientific and Statistical Computing*, Vol. 3, pp. 367~384.

Potra, F. A., 1991, "Multistep Method for Solving Constrained Equations of Motion," Univ. Of Iowa, *Ctr. For Simulation and Design Optim. Of Mech. System, Tech.* Rep. R-104, Feb.

Shampine, L. F., and Gordon, M. K., 1975, "Computer Solution of Ordinary Differential Equations: The Initial Value Problem," Freeman, San Francisco

Singh, R. P., and Likins, P. W., 1985, "Singular Values Decomposition for Constrained Dynamical Systems," *Journal of Applied Mechanics*, Vol. 52, pp. 943~948.

Strang, G., 1980. *Linear Algebraic and Its Application*, 2nd Edition, Academic Press, New York.

Unda, J., Gricia de Jalon, J., Losantos, F., and Enparantza, R., 1987, "A Comparative Study of Some Different Formulations of the Dynamic Equations of Constrained Mechanical Systems," *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 109, pp. 466~474.

Walton, W. C., and Steeves, E. C., 1969, "A New Matrix Theroem and Its Application for Establishing Independent Coordinates for Complex Dynamical Systems with Constraints," NASA TR-R326

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