Computational Method for Dynamic Analysis of Constrained Mechanical Systems Using Partial Velocity Matrix Transformation

Jung Hun Park, Hong Hee Yoo*
Department of Mechanical Engineering, Hanyang University
Yoha Hwang
Department of Tribology Research Center, KIST

A computational method for the dynamic analysis of a constrained mechanical system is presented in this paper. The partial velocity matrix, which is the null space of the Jacobian of the constraint equations, is used as the key ingredient for the derivation of reduced equations of motion. The acceleration constraint equations are solved simultaneously with the equations of motion. Thus, the total number of equations to be integrated is equivalent to that of the pseudo generalized coordinates, which denote all the variables employed to describe the configuration of the system of concern. Two well-known conventional methods are briefly introduced and compared with the present method. Three numerical examples are solved to demonstrate the solution accuracy, the computational efficiency, and the numerical stability of the present method.

Key Words : Partial Velocity Matrix, Differential and Algebraic Equations, Constrained Multibody Systems, Kane's Method

1. Introduction

A mixed set of differential and algebraic equations (abbreviated hereafter as DAE) often needs to be solved simultaneously to obtain the transient response of a constrained mechanical system. If the number of equations of motion (abbreviated hereafter as EOM) of the system is greater than the degree of freedom, one or more algebraic equations are involved in the EOM. Solving a set of DAE is much more challenging than solving a set of pure differential equations.

Several computational methods (Mani and Haug, 1984 ; 1986 ; Brenan et al., 1996 ; Baumgarte, 1972 ; Chang and Nikravesh, 1985 ; Haug and Yen, 1992 ; Wehage and Haug, 1982 ; Lee et al., 1994a ; 1994b) to solve DAE have been suggested in the literature. Among them, constraint stabilization method (abbreviated hereafter as CS method) and generalized coordinates partitioning method (abbreviated hereafter as GCP method) are two well-known methods. The CS method (Baumgarte, 1972 ; Chang and Nikravesh, 1985) employs a feedback control logic to solve the EOM. This method seems to be computationally efficient since it does not perform position and velocity analysis while solving the EOM. However, due to the same reason, this method often loses the solution accuracy when the system is near a kinematically singular position or takes abrupt external disturbances. Thus, the CS method often fails to maintain the numerical stability. In the GCP method (Haug and Yen, 1992 ; Wehage and Haug, 1982), the pseudo generalized coordinates (which are often called generalized coordinates, used for different meaning in this paper) are partitioned into independent and dependent coordinates. Two methods are available for the GCP method after the par-

* Corresponding Author,
E-mail : yhh@hydyn. hanyang. ac. kr
TEL : +82-2-2290-0446 ; **FAX** : +82-2-2293-5070
Department of Mechanical engineering, Hanyang University, 17 Haengdang-dong, Sungdong-ku, Seoul 133-791, Korea. (Manuscript **Received** April 9, 1999 ; **Revised** October 26, 1999)

titioning. In the first method, the EOM corresponding to the independent coordinates only are integrated, and the dependent coordinates are calculated through a complementary procedure. The independent coordinates, however, often need to be re-selected to obtain accurate numerical solutions and integration must be re-started. In the second method, all the equations are integrated to avoid re-starting of integration. Even though the integrated dependent coordinates are available, they are adjusted to satisfy the algebraic constraint equations through a complementary procedure. The second method has been found to be numerically robust. So, it is implemented in some commercial programs (Haug, 1989).

In Kane's method (Kane and Levison, 1985 ; Kamman and Huston, 1984 ; Huston et al., 1978 ; Huston and Passerello, 1979 ; 1980 ; Amirouche, 1992 ; Park et al., 1997), generalized coordinates (which mean the necessary and sufficient coordinates to describe the system of concern in this study) are usually employed to derive the EOM. Since the generalized coordinates are the fewest number of coordinates that are necessary to describe the system, the equation formulation using the generalized coordinates is computationally more efficient than any other formulations. However, the generalized coordinates are often not suitable for describing constrained mechanical systems (especially when the system has kinematic loops). For constrained mechanical systems, employing six (when Euler angles are employed) or seven (when Euler parameters are employed) coordinates for each rigid body is convenient. Since these coordinates are generally not independent, they are called the pseudo generalized coordinates. Based on the GCP method that employs the pseudo generalized coordinates, the computational method presented in this study utilizes favorable aspects of Kane's method for the dynamic analysis of constrained mechanical systems. Especially, the concept of partial velocities in Kane's method is utilized to introduce the partial velocity matrix, which plays the key role in the efficiency of this new method. Three numerical examples are presented and solved to demonstrate the numerical efficiency and stability of this new method. The CS method and the GCP method are briefly summarized in the following section and their numerical results and performance are compared with those of the proposed method in the penultimate section.

2. Conventional Methods

For constrained mechanical systems, the EOM and constraint equations are conventionally written as follows:

$$\boldsymbol{M}\ddot{\boldsymbol{q}}+\boldsymbol{\Phi}_q^T\boldsymbol{\lambda}=\boldsymbol{Q}, \tag{1}$$

$$\boldsymbol{\Phi}(\boldsymbol{q},\ t)=\boldsymbol{0} \tag{2}$$

where $\boldsymbol{M}$ of dimension $n\times n$ denotes the mass matrix which depends on the pseudo generalized coordinates vector $\boldsymbol{q}$ of dimension n, $\boldsymbol{\Phi}$ of dimension m denotes the vector of algebraic constraint equations, $\boldsymbol{\lambda}$ of dimension m denotes the Lagrange multiplier vector, and $\boldsymbol{Q}$ of dimension n denotes the general force vector which depends on $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and t.

$\boldsymbol{q}$ is sometimes called the generalized coordinates vector in the literature. However, the terminology (generalized coordinates) is used to mean differently in Kane's method (Kane and Levinson, 1985). Therefore, to avoid unnecessary confusion, $\boldsymbol{q}$ is named as the pseudo generalized coordinates vector in this study. A dot over a symbol denotes the time differentiation of the symbol, and the subscript in a symbol denotes partial derivative of the symbol with respect to the subscript.

Both sides of Eq. (2) may be differentiated with respect to time and rearranged to obtain the velocity constraint equations.

$$\boldsymbol{\Phi}_q\dot{\boldsymbol{q}}=-\boldsymbol{\Phi}_t \tag{3}$$

Just as the velocity constraint equations of Eq. (3) were obtained by differentiating the constraint equations of Eq. (2), differentiating both sides of Eq. (3) yields the acceleration constraint equations.

$$\boldsymbol{\Phi}_q\ddot{\boldsymbol{q}}=\boldsymbol{\gamma}, \tag{4}$$

$$\boldsymbol{\gamma}=-(\boldsymbol{\Phi}_q\dot{\boldsymbol{q}})_q\dot{\boldsymbol{q}}-2\boldsymbol{\Phi}_{qt}\dot{\boldsymbol{q}}-\boldsymbol{\Phi}_{tt} \tag{5}$$

where $\boldsymbol{\Phi}_q$ of dimension $m\times n$ denotes the

Jacobian matrix of the constraint equations. In this section, the CS method and the GCP method are briefly summarized first. Later, the computational efficiency and accuracy of these methods will be compared with those of the new method proposed in this study.

2.1 CS method

In CS method, the following modified acceleration constraint equations are utilized to solve the EOM.

$$\ddot{\boldsymbol{\Phi}}+2\alpha\dot{\boldsymbol{\Phi}}+\beta^2\boldsymbol{\Phi}=\mathbf{0} \tag{6}$$

where α and β are appropriately chosen constants. The general solution for this differential equation can be written as follows.

$$\boldsymbol{\Phi}=\boldsymbol{a}_1e^{s_1t}+\boldsymbol{a}_2e^{s_2t} \tag{7}$$

where $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are constant vectors that depend on the initial conditions, and s_1 and s_2 are the roots of the characteristic equation of Eq. (6) that can be expressed as follows:

$$s_1, s_2=-\alpha\pm\sqrt{\alpha^2-\beta^2} \tag{8}$$

If α and β are positive constants, the two roots have negative real part. This guarantees the satisfaction of the constraint equations in Eqs. (2) and (3). The initial position and velocity conditions of the mechanical system should guarantee that the vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ are zero. To implement the CS method, Eq. (6) is written explicitly in the form

$$\boldsymbol{\Phi}_q\ddot{q}=\boldsymbol{\gamma}-2\alpha\dot{\boldsymbol{\Phi}}-\beta^2\boldsymbol{\Phi}\equiv\bar{\boldsymbol{\gamma}} \tag{9}$$

By using Eqs. (1) and (9), the EOM can be written as follows.

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{\Phi}_q^T \\ \boldsymbol{\Phi}_q & 0 \end{bmatrix}\begin{Bmatrix} \ddot{q} \\ \boldsymbol{\lambda} \end{Bmatrix}=\begin{Bmatrix} \boldsymbol{Q} \\ \bar{\boldsymbol{\gamma}} \end{Bmatrix} \tag{10}$$

If appropriate values of α and β are chosen, constraint violation will be reduced during the time integration.

Since the CS method does not involve position or velocity analysis, the computational cost of the CS method is smaller than that of the GCP method. With the CS method, however, instability often occurs near a kinematically singular configuration. Once the instability occurs, the efficiency of the method will be endangered. Furthermore, it is often computationally expensive to determine the appropriate values of α and β.

2.2 GCP method

By differentiating the constraint equations with respect to the pseudo generalized coordinates vector $\boldsymbol{q}$, a Jacobian matrix can be obtained. With the Jacobian matrix, full row and column pivoting is performed to automatically partition the pseudo generalized coordinates $\boldsymbol{q}$ into independent coordinates $\boldsymbol{q}_i$ of dimension $n-m$ and dependent coordinates $\boldsymbol{q}_d$ of dimension m. So the pseudo generalized coordinates can be written as

$$\boldsymbol{q}=[\boldsymbol{q}_d^T\ \boldsymbol{q}_i^T]^T \tag{11}$$

Incidentally, redundant constraints can be automatically eliminated at this stage by checking the rank of the Jacobian matrix. Now, if $\boldsymbol{q}_i$ is given, $\boldsymbol{q}_d$ can be obtained by the iterative procedure that employs the following matrix equation.

$$\boldsymbol{\Phi}_{q_d}^k\Delta\boldsymbol{q}_d^k=-\boldsymbol{\Phi}^k \tag{12}$$

The symbols with superscript k in Eq. (12) represent the value of the symbols at the k-th iteration. By using the above equation, the improved solution of $\boldsymbol{q}_d$ for the $(k+1)$-th iteration can be obtained as follows:

$$\boldsymbol{q}_d^{k+1}=\boldsymbol{q}_d^k+\Delta\boldsymbol{q}_d^k \tag{13}$$

By using Eqs. (12) and (13), the iteration continues until the solution variance remains within a specified allowable error tolerance. The procedure mentioned so far is usually called the position analysis.

Once the position analysis solution $\boldsymbol{q}_d$ is found, the velocity analysis solution can be obtained by solving the following velocity constraint equations that are equivalent to Eq. (3).

$$\boldsymbol{\Phi}_{q_d}\dot{\boldsymbol{q}}_d=-\boldsymbol{\Phi}_{q_i}\dot{\boldsymbol{q}}_i-\boldsymbol{\Phi}_t \tag{14}$$

Different from the position analysis, the solution of Eq. (14) can be obtained without the iterative procedure.

Once the position and velocity analysis solutions are found, the following EOM are to be solved.

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{\Phi}_q^T \\ \boldsymbol{\Phi}_q & 0 \end{bmatrix}\begin{Bmatrix} \ddot{q} \\ \boldsymbol{\lambda} \end{Bmatrix}=\begin{Bmatrix} \boldsymbol{Q} \\ \boldsymbol{\gamma} \end{Bmatrix} \tag{15}$$

This method is numerically robust compared with the CS method since it forces the pseudo generalized coordinates to satisfy the constraint equations during the time integration. However, in order for the EOM in Eq. (15) to be solved, $n+m$ linear equations need to be solved during the each step of the time integration.

A new method that is proposed in the following section solves only n linear equations instead of $n+m$ equations. That will make a difference to the computational efficiency of the proposed method.

3. Proposed Method

A new method will be proposed in this section. This method is devised based on the GCP method explained in the previous section. So, the method of choosing independent coordinates $\boldsymbol{q}_i$ from the pseudo generalized coordinates $\boldsymbol{q}$, and the method of performing position and velocity analysis are the same as those of the GCP method.

Following the convention of Kane's method, the generalized speeds vector $\boldsymbol{u}$ is defined as follows:

$$\boldsymbol{u} \equiv \dot{\boldsymbol{q}}_i \tag{16}$$

Now, using Eq. (16) along with the velocity constraint equation, $\dot{\boldsymbol{q}}$ can be expressed as a function of $\boldsymbol{u}$ as follows: (Park et al., 1997)

$$\dot{\boldsymbol{q}} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{c} \tag{17}$$

where $\boldsymbol{B}$ (named the partial velocity matrix) and $\boldsymbol{c}$ are given as follows:

$$\boldsymbol{B} = \begin{bmatrix} -\boldsymbol{\Phi}_{q_d}^{-1}\boldsymbol{\Phi}_{q_i} \\ \boldsymbol{I} \end{bmatrix}, \tag{18}$$

$$\boldsymbol{c} = \begin{bmatrix} -\boldsymbol{\Phi}_{q_d}^{-1}\boldsymbol{\Phi}_t \\ 0 \end{bmatrix}. \tag{19}$$

In the above equations, $\boldsymbol{\Phi}_{q_d}^{-1}\boldsymbol{\Phi}_{q_i}$ and $\boldsymbol{\Phi}_{q_d}^{-1}\boldsymbol{\Phi}_t$ can be calculated efficiently by performing only $n-m+1$ times back-substitutions (without LU decomposition, which was previously performed during the velocity analysis).

Now, by multiplying the EOM in Eq. (1) by the transposed partial velocity matrix obtained in Eq. (18), the following EOM can be obtained.

$$-\boldsymbol{B}^T\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{B}^T\boldsymbol{Q} = 0. \tag{20}$$

The reaction forces or torques corresponding to the constraint equations are eliminated during the multiplication since the partial velocity matrix is the null space of the Jacobian of the constraint equations. Thus, in the above equations, only two terms remain after the multiplication. The first and second terms are called the generalized inertia forces and the generalized active forces, respectively. Since Eq. (20) consists of $n-m$ equations with n unknown variables $\ddot{\boldsymbol{q}}$, the unknown variables cannot be determined uniquely. Thus, m acceleration constraint equations given in Eq. (4) are supplemented to find the solution of $\ddot{\boldsymbol{q}}$. In other words, Eq. (4) and Eq. (20) are to be solved simultaneously. The two sets of equations can be written in a matrix form as follows:

$$\begin{bmatrix} \boldsymbol{B}^T\boldsymbol{M} \\ \boldsymbol{\Phi}_q \end{bmatrix} \ddot{\boldsymbol{q}} = \begin{Bmatrix} \boldsymbol{B}^T\boldsymbol{Q} \\ \boldsymbol{\gamma} \end{Bmatrix}. \tag{21}$$

That is, only n equations instead of $n+m$ equations (see Eq. (10) and Eq. (15)), are to be solved in this method. And consequently, compared with the two previous methods, better computational efficiency results from this. In this method, numerical instability can be also avoided since the position and velocity analyses are performed during the integration.

To calculate reaction forces or torques at the kinematic constraints, it is necessary to calculate the vector of Lagrange multipliers $\boldsymbol{\lambda}$. For this method, the Lagrange multipliers can be calculated in the following equation that can be obtained from Eq. (1).

$$\begin{aligned} \boldsymbol{\Phi}_q^T\boldsymbol{\lambda} &= \boldsymbol{b} \\ &= \boldsymbol{Q} - \boldsymbol{M}\ddot{\boldsymbol{q}}. \end{aligned} \tag{22}$$

$\boldsymbol{\Phi}_q$, $\boldsymbol{Q}$, and $\boldsymbol{M}\ddot{\boldsymbol{q}}$ are calculated during the formulation procedure of EOM. Eq. (22) cannot be solved straightforward since there are n equations with m unknown variables $\boldsymbol{\lambda}$. To remove redundant equations from Eq. (22), the equation is partitioned as follows.

$$\begin{bmatrix} \boldsymbol{\Phi}_{q_d}^T \\ \boldsymbol{\Phi}_{q_i}^T \end{bmatrix} \boldsymbol{\lambda} = \begin{Bmatrix} \boldsymbol{b}_d \\ \boldsymbol{b}_i \end{Bmatrix} \tag{23}$$

in which $\boldsymbol{\Phi}_{q_d}$ has full rank with m. Thus, the independent equations to calculate Lagrange multipliers can be written as follows:

$$\boldsymbol{\Phi}_{q_d}^T \boldsymbol{\lambda} = \boldsymbol{b}_d. \qquad (24)$$

Since LU decomposition for $\boldsymbol{\Phi}_{q_d}$ is previously performed in the velocity analysis, it is not necessary to repeat the decomposition to obtain the Lagrange multipliers.

4. Numerical Examples

To examine the computational efficiency, accuracy, and reliability of the method proposed in this paper, three numerical examples are given in this section. The computational efficiency of the new method is evaluated through the computation time (compared with those of two other conventional methods explained in section 2). The computation time includes only the minimum time required to obtain the solution. That is, the time required for the extra calculation such as Lagrange multipliers at the proposed method is excluded. The accuracy and reliability are checked with a problem that often causes numerical instability. All the numerical calculations were performed by a desk-top personal computer.

4.1 Double bar pendulum

A double bar pendulum is shown in Fig. 1. This is a typical open loop system that undergoes planar motion. The properties of the system are given as follows. ; $a_1 = a_2 = b_1 = b_2 = 1.0[m]$, $m_1 = 5.0[kg]$, $m_2 = 7.0[kg]$, $J_1 = 10.0[kg \cdot m^2]$, $J_2 = 20.0[kg \cdot m^2]$, A constant M^1 (magnitude of 10.0 $[N \cdot m]$) is applied to the first body, and a constant force $F^p = 8.0\hat{n}_1 + 7.0\hat{n}_2[N]$ acts on the point p shown in the figure. Initially, θ_1, θ_2, $\dot{\theta}_1$, and $\dot{\theta}_2$ are 0.524[rad], 0.785[rad], 0.000[rad/sec], and 0.000[rad/sec], respectively. Figure 2 shows the results of orientation angle θ_2 of body 2 with in 10 seconds of simulation time. It shows that the numerical results obtained by three methods (CS method, GCP method, and the proposed method in this paper) are almost identical. Table 1 contains CPU time ratio (the CPU time consumed by the proposed method is used as the reference), number of function evaluation, and average integration step size for three cases of simulation time interval. It shows that the new method proposed in this paper is computationally more efficient than other two methods for the three cases of simulation time interval. The significant reduction of simulation time results from the

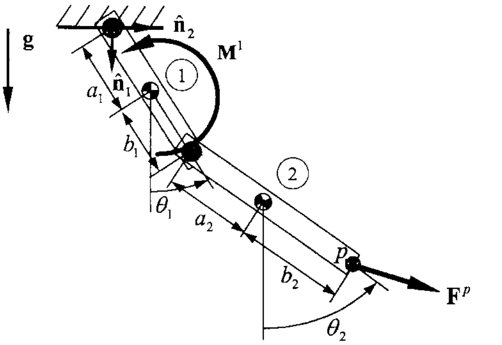

Fig. 1 A planar double pendulum

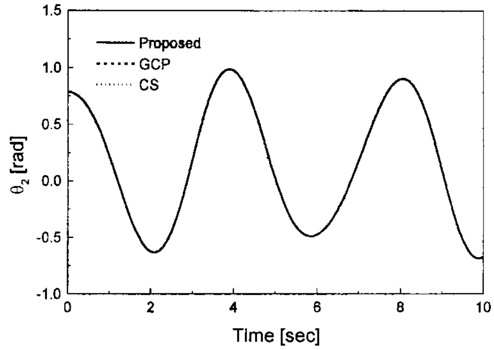

Fig. 2 Angular displacement θ_2 of body

Table 1 Number of function evaluations, CPU time ratio, and average integration step size for example 1

Simulation time [sec]	Method	CPU time ratio	No. of function evaluation	Average integration step size
5.0	Proposed	1.00	96	0.052
	GCP	1.70	96	0.052
	CS	1.10	140	0.035
10.0	Proposed	1.00	151	0.066
	GCP	1.65	152	0.066
	CS	1.65	283	0.035
50.0	Proposed	1.00	625	0.080
	GCP	1.86	607	0.082
	CS	2.17	1313	0.038

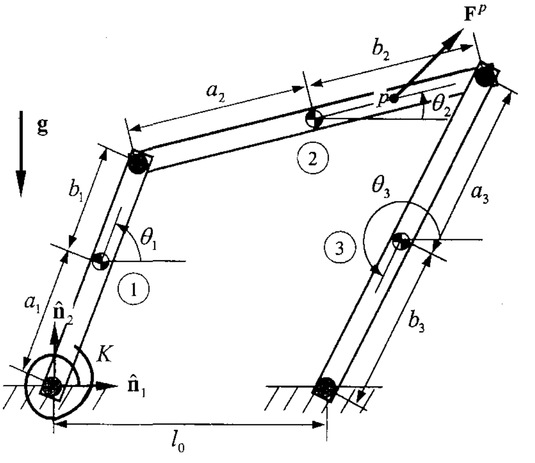

Fig. 3 A planar four bar mechanism

smaller matrix sizes (in the EOM) of the proposed method. Incidentally, in the case of the short interval (5.0 seconds) simulation, the CPU time of the CS method (in spite of its smaller step size) is smaller than that of the GCP method. This results from the fact that the position and velocity analyses are not performed in the CS method. However, as the simulation time interval increases, the CPU time of the CS method increases faster than that of GCP method. Since position and velocity analyses are not performed in the CS method, the fast increasing numerical error eventually deteriorates the computational efficiency of the CS method.

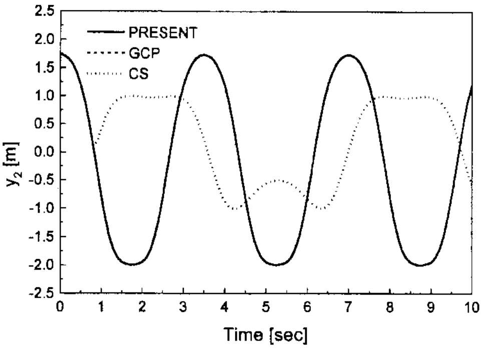

Fig. 4 Displacement y_2 of link 2 mass center

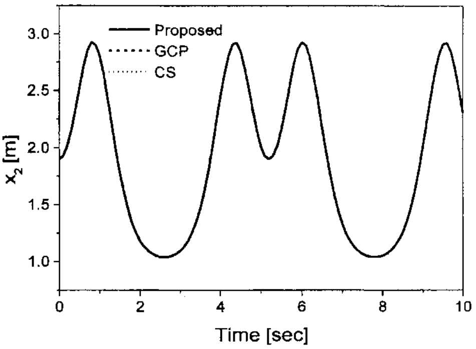

Fig. 5 Displacement x_2 of link 2 mass center

Table 2 Number of function evaluations, CPU time ratio, and average integration step size for example 2

	Method	CPU time ratio	No. of function evaluation	Average integration step size
Case Ⅰ	Proposed	1.00	303	0.033
	GCP	1.60	295	0.034
	CS	2.31	699	0.014
Case Ⅱ	Proposed	1.00	311	0.032
	GCP	1.60	311	0.032
	CS	1.80	707	0.014

4.2 Planar four bar mechanism

A planar four bar mechanism is shown in Fig. 3. This is a typical closed loop system that undergoes planar motion. The properties of the system are given as follows; $m_1=5.0[kg]$, $m_2=7.0[kg]$, $m_3=9.0[kg]$, $J_1=10.0[kg \cdot m^2]$, $J_2=20.0[kg \cdot m^2]$, $J_3=25.0[kg \cdot m^2]$, $a_1=a_2=a_3=1.0[m]$, and $b_1=b_2=b_3=1.0[m]$, where m_i and J_i denote the mass and the moment of inertia (about the mass center) of the i-th link, respectively. The stiffness of the rotational spring (K) is $100[N \cdot m/rad]$. The spring is not deformed when $\theta_1=0.524$ $[rad]$. An external force $F^p=80.0\,\hat{n}_1+70.0\,\hat{n}_2$ $[N]$ is applied to the point p on the connecting link. The first simulation is carried out with $l_0=2.0[m]$. Initial conditions are given as follows; $\theta_1=1.074[rad]$ and $\dot{\theta}_1=0.000[rad/\text{sec}]$. The kinematic singularity occurs when $y_2=0.0[m]$. Figure 4 shows that the CS method produces spurious results while the GCP and the proposed methods produce accurate results (well matched to each other). Since the CS method does not perform position and velocity analyses, it usually encounters trouble near the singular position. The

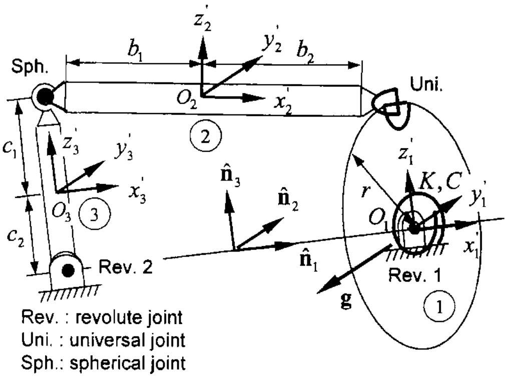

Fig. 6 Spatial four bar mechanism

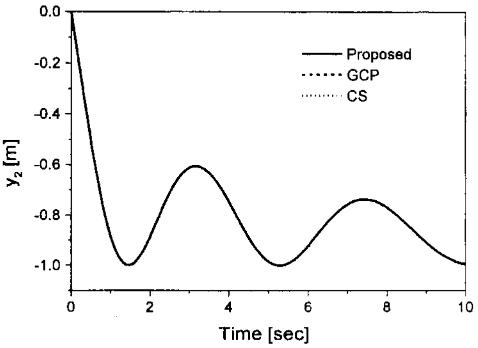

Fig. 7 Displacement y_2 of link 2 mass center

second simulation is carried out with $l_0 = 1.9[m]$. Initial conditions are given as follows. ; $\theta_1 = 1.104$ $[rad]$ and $\dot{\theta}_1 = 0.000[rad/\text{sec}]$. Figure 5 shows the results of the displacement x_2 of the second link. Since kinematic singularity does not occur in this case, the numerical results obtained by three methods are almost identically matched. Table 2, however, shows that the computational efficiency of the proposed method is superior to that of the other two methods in both cases.

4.3 Spatial four bar mechanism

A spatial four bar mechanism is shown in Fig. 6. The properties are given as follows; $m_1 = 2.0$ $[kg]$, $m_2 = 1.0[kg]$, $m_3 = 1.0[kg]$, $J_{1xx} = 4.0[kg \cdot m^2]$, $J_{1yy} = J_{1zz} = 2.0[kg \cdot m^2]$, $J_{2xx} = 0.01[kg \cdot m^2]$, $J_{2yy} = J_{2zz} = 12.4[kg \cdot m^2]$, $J_{3xx} = J_{3yy} = 4.54$ $[kg \cdot m^2]$, $J_{3zz} = 0.01[kg \cdot m^2]$, $r = 2.0[m]$, $b_1 = b_2 = 6.0[m]$, $c_1 = c_2 = 1.0[m]$, where J_{ixx}, J_{iyy}, and J_{izz} represent the principal moments of inertia of the i-th link in x, y, and z directions (that is, principal directions), respectively. Ground and body 1, body 1 and body 2, body 2 and body 3, and body 3 and ground are connected by a revolute joint, universal joint, spherical joint, and revolute joint, respectively. Active forces by rotational spring, damper, and gravitation are given for the system. The rotational spring has stiffness K of $1.0[N \cdot m/rad]$ and the damper has damping constant C of $5.0[N \cdot m \cdot \text{sec}/rad]$. The spring is not deformed when $\theta_1 = 0.0[rad]$. Simulation is carried out with the initial conditions of $\theta_1 = 0.0[rad]$ and $\dot{\theta}_1 = 0.0[rad/\text{sec}]$. Figure 7 shows the results of the displacement y_2 of link 2 with in 10 seconds of simulation time,

Table 3 CPU time ratio and the number of function evaluations for example 3 with the integration error tolerance ε_{int} and the position level error tolerance ε_{pos}

ε_{int}	ε_{pos}	Method	CPU time ratio	No. of function evaluation
$1.0e^{-5}$	$1.0e^{-3}$	Proposed	1.00	210
		GCP	1.68	210
		CS	3.59	662
$1.0e^{-3}$	$1.0e^{-3}$	Proposed	1.00	157
		GCP	1.58	157
		CS	2.76	399
$1.0e^{-3}$	$1.0e^{-5}$	Proposed	1.00	154
		GCP	1.33	152
		CS	1.80	399

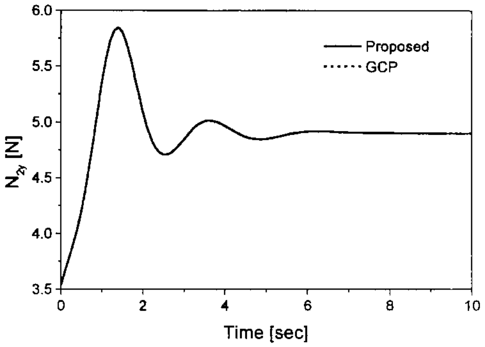

Fig. 8 Reaction force on the spherical joint N_{2y} of link 2

when the integration error tolerance ε_{int} and the position analysis error tolerance ε_{int} are 0.001. The analysis results obtained by three methods are almost identical. Table 3 contains the simulation results with three cases of position and integration error tolerance. Since the proposed method performs the same position analysis as the GCP method, the advantage of the proposed method over the GCP method relatively decreases as the position error tolerance decreases. On the other hand, when the integration error tolerance decreases, the relative advantage of the proposed method increases. In any cases, however, the proposed method provides the best computational efficiency. Figure 8 shows the results of the reaction force N_{2y} on the spherical joint in Fig. 6. It shows that the numerical results obtained by the proposed method are almost identical to those of the GCP method.

5. Conclusion

A new formulation for the dynamic analysis of a constrained mechanical system is proposed in this study. A partial velocity matrix is introduced for the formulation to reduce the number of equations to be solved. Even if the Lagrange multipliers (representing reaction forces and torques at the joints) are not appeared in the EOM, they can be efficiently obtained by this method. The proposed method is computationally more efficient than the other two conventional methods (remarked in this paper for the comparison study) since it employs smaller size EOM for the time integration. This method is also numerically stable since it performs position and velocity analyses during the time integration. The computational efficiency as well as the stability is confirmed through three numerical examples.

References

Amirouche, F. M. L., 1992, *Computational Methods in Multibody Dynamics*, Prentice-Hall.

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

Brenan, K. E., Campbell, S. L., and Petzold, L. R., 1996, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM.

Chang, C. O. and Nikravesh, P. E., 1985, "An Adaptive Constraint Violation Stabilization Method for Dynamic Analysis of Mechanical Systems," *J. Mechanisms, Transmissions, and Automation in Design*, Vol. 107, pp. 488～492.

Haug, E. J. and Yen, J., 1992, "Implicit Numerical Integration of Constrained Equations of Motion via Generalized Coordinate Partitioning," *J. Mechanical Design*, Vol. 114, pp. 296～304.

Haug, E. J., 1989, *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, ALLYN AND BACON.

Huston, R. L. and Passerello, C. E., 1979, "On Multi-rigid Body System Dynamics," *Computers & Structures*, Vol. 10, pp. 439～446.

Huston, R. L. and Passerello, C. E., 1980, "Multibody Structural Dynamics including Translation between the Bodies," *Computers & Structures*, Vol. 12, pp. 713～720.

Huston, R. L., Passerello, C. E., and Harlow, M. W., 1978, "Dynamics of Multirigid-Body Systems," *J. Applied Mechanics*, Vol. 45, pp. 889～894.

Kamman, J. W. and Huston, R. L., 1984, "Dynamics of Constrained Multibody Systems," *J. Applied Mechanics*, Vol. 51, pp. 899～903.

Kane, T. R. and Levinson, D. A., 1985, *DYNAMICS: Theory and Application*, Mcgraw-Hill.

Lee, D. C., Bae, D. S., Han, C. S., Seo, M. S., and Kim, J. Y., 1994a, "A Study on the Dynamic Analysis of Multibody System by the Relative Joint Coordinate Method," *Transactions of the KSME*, Vol. 18, No. 7, pp. 1974～1984.

Lee, S. H., Bae, D. S., and Han, C. S., 1994b, "A Dynamic Analysis of Constrained Multibody Systems," *Transactions of the KSME*, Vol. 18, No. 9, pp. 2239～2348.

Mani, N. K. and Haug, E. J., 1984, "Use of Singular Value Decomposition for Analysis and Optimization of Mechanical System Dynamics," *Technical Report 84-13*, University of Iowa, Iowa city.

Park, J. H., Yoo, H. H., Hwnag, Y, and Bae, D. S., 1997, "Dynamic Analysis of Constrained Multibody Systems Using Kane's Method," *Transactions of the KSME*, Vol. 21, No. 12, pp. 2156~2164.

Park, T. W. and Haug, E. J., 1986, "Hybrid Numerical Integration Method for Machine Dynamic Simulation," *J. Mechanisms, Transmissions, and Automation in Design*, Vol. 108, pp. 211~216.

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