

〈論 文〉

반복 계산법 및 계산 가속기법에 의한 다물체 동역학 해법

이기수* · 임철호**

(1992년 1월 27일 접수)

An Accelerated Iterative Method for the Dynamic Analysis of Multibody Systems

Kisu Lee and Cheol-Ho Lim

Key Words : Differential Algebraic Equation(대수 미분 방정식), Constraints(제한조건), Iterative Scheme(반복 계산법), Acceleration Technique(계산 가속 기법), Time Integration(시간적분)

Abstract

An iterative solution technique is presented to analyze the dynamic systems of rigid bodies subjected to kinematic constraints. Lagrange multipliers associated with the constraints are iteratively computed by monotonically reducing an appropriately defined constraint error vector, and the resulting equation of motion is solved by a well-established ODE technique. Constraints on the velocity and acceleration as well as the position are made to be satisfied at joints at each time step. Time integration is efficiently performed because decomposition or orthonormalization of the large matrix is not required at all. An acceleration technique is suggested for the faster convergence of the iterative scheme.

1. 서 론

여러개의 물체가 연결점을 통하여 연결된 다물체 동역학은 우주선의 동력학적 해석, 기계 동역학 및 차량 동역학 등에서 광범위하게 응용되고 있다. 다물체 동역학에서 각각의 물체는 모두 운동 방정식을 만족시켜야 하며 또 연결점에서 각 물체는 기구학적 제한 조건을 만족시켜야 한다. 이러한 문제의 경우에는 미분 방정식(운동 방정식)과 대수 방정식(기구학적 제한조건)이 연립 방정식을 구성하게 되므로 상미분 방정식의 시간 적분법이 그대로 적용될 수는 없으므로 특수한 해법이 필요하며⁽¹⁻²⁾ 수많은 방법이 소개되었다. Orlandea 등⁽³⁾ 및 Car-

dona 등⁽⁴⁾처럼 특수한 시간 적분법을 사용하여 전체 연립 방정식을 푸는 경우 경제적인 계산은 가능하나 각종 제한 조건으로 인한 오차의 누적을 방지할 필요가 있다. Predictor-corrector 시간 적분법과 같은 잘 알려진 상미분 방정식 해법을 적용하기 위하여는 제한 조건으로 인하여 발생하는 미지수의 힘을 운동 방정식에서 소거하여 순수한 미분 방정식을 유도하는 것이 편리하다. 이러한 목적으로 미지수인 Lagrange multiplier를 운동방정식에서 소거하거나 독립 좌표를 구하기 위하여 Kamman과 Huston의 zero eigenvalue 기법⁽⁵⁾, Sing과 Likins의 singular value decomposition⁽⁶⁾, Kim과 Vanderploeg의 QR decomposition⁽⁷⁾, Lance의 differential null space⁽⁸⁾, Ider와 Amirouche의 Gaussian row operation⁽⁹⁾, Wehage와 Haug의 LU decomposition⁽¹⁰⁾들이 제안되었다. 그러나 제한조

*정회원, 한국항공우주연구소 위성분체실

**정회원, 한국과학기술연구원 시스템공학연구소

건의 Jacobian 행렬은 시시각각 변하므로, 매 time step마다 이러한 특수한 기법이 사용되어야 하는 경우 계산 시간은 대단히 증가하게 되며 계산의 정확도는 독립 좌표를 구하는 방법에 따라서 달라질 수 있다. 또 위치 제한조건을 제대로 만족시키기 위하여는 일반적으로 Newton Raphson 반복법이 추가로 필요하다. Park과 Haug⁽¹¹⁾는 시간이 많이 걸리는 N-R 반복법등을 필요한 time step에서만 수행함으로써 계산 시간을 절감하였다.

본 작업에서는 위에서 언급한 대수 미분 방정식을 풀기위한 새로운 방법을 소개한다. 본 작업에서는 Lagrange multiplier의 값이 사전에 주어졌다고 생각하여, 즉 대수 미분 방정식을 순수한 상미분 방정식으로 변환하여, 잘 알려진 시간 적분법을 적용한다. 또 정확한 Lagrange multiplier값은 반복 계산법(iterative scheme)에 의하여 계산한다. 시간 적분의 정확도와 제한 조건의 정확도는 모두 보장된다. 특히 제한 조건의 경우, 위치, 속도 및 가속도의 제한 조건이 모두 만족된다. 또 정확한 Lagrange multiplier의 값을 계산 가속 기법(acceleration technique)에 의하여 대단히 빨리 계산한다. 독립 좌표를 구할 필요가 없으므로 거대한 행렬을 decomposition하는 등의 복잡한 절차가 불필요하며 N-R 반복법 역시 불필요하다. 이러한 사항들 및 Jacobian 행렬의 sparsity로 인하여 경제적인 계산이 가능하게 된다.

2. 기본 방정식

2.1 운동 방정식 및 제한조건

본 작업에서는 모든 물체를 강체(rigid body)로 취급한다. 기구학적 제한 조건이 부과된 강체들의 운동 방정식은 다음과 같이 쓸 수 있다.

$$M\ddot{q} + \Phi_i^* \lambda = f \tag{1}$$

위에서 M 은 질량행렬, Φ_i^* 는 제한 조건식의 전치 Jacobian 행렬, q 는 일반화 좌표계(위치), λ 는 Lagrange multiplier, 그리고 f 는 힘을 나타낸다. 본 작업에서 사용하는 기구학적 제한조건 Φ 는 다음과 같이 표시되는 holonomic constraint이다.

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

위에서 t 는 시간을 나타낸다. 동역학적 해석을 위하여 미분 방정식 (1)과 대수 방정식(2a)는 동시

에 풀려야 한다. 만약 식(1)에서 정확한 Lagrange multiplier의 값이 시간 적분 이전에 주어진다면, 식 (1)은 상미분 방정식이 되며, 수많은 참고문헌에 소개되어 있는 시간 적분법을 그대로 이용할 수 있다. Lagrange multiplier의 값이 정확히 주어지면 제한 조건식 (2a)는 자동적으로 만족된다. 제한 조건식 (2a)로 인하여 다음의 미분된 제한 조건식들도 만족되어야 한다.

$$\Phi_q \dot{q} + \Phi_t = 0 \tag{2b}$$

$$\Phi_q \ddot{q} + (\Phi_{qq} \dot{q})_q \dot{q} + 2\Phi_{qt} \dot{q} + \Phi_{tt} = 0 \tag{2c}$$

현실적인 계산에서는, 식 (2a)~(2c)들은 그중의 하나가 만족되었다고 하여 나머지 2식이 저절로 만족되지는 않으므로 특별한 기법에 의하여 이들 3식을 모두 만족시키는 것이 필요하다. 이 목적을 위하여 본 작업에서는 다음과 같은 세가지 형태의 제한 조건 오차를 정의한다.

(1) 위치 제한 조건의 경우, 제한 조건 오차 v 는

$$v = -\Phi \tag{3a}$$

(2) 속도 제한 조건의 경우, 제한 조건 오차 v 는

$$v = -\Phi_q \dot{q} - \Phi_t \tag{3b}$$

(3) 가속도 제한 조건의 경우, 제한 조건 오차 v 는

$$v = -\Phi_q^* \ddot{q} - (\Phi_{qq} \dot{q})_q \dot{q} - 2\Phi_{qt} \dot{q} - \Phi_{tt} \tag{3c}$$

실제 계산시 제한 조건 오차 v 는, 계산 과정의 필요에 의하여, 식 (3a)~(3c)중의 어느 하나가 선택된다.

2.2 계산 속도 가속을 위한 행렬

본 작업에서는 참고 문헌(12)에서 사용된 계산 가속기법(acceleration technique)을 사용하여 계산 속도를 크게 증진시킨다. 계산 가속 기법에 대한 자세한 내용은 참고문헌(12)에 소개되어 있으므로 여기에서는 당장 필요한 행렬에 대해서만 언급하기로 한다. 대칭이고 positive definite인 기본 행렬 C 로부터 다음과 같은 일군의 행렬이 정의된다.

$$\begin{aligned} C_n &= C / \|C\|_\infty \text{ if } n=1 \\ &= C_{n-1}(b_n I - C_{n-1}) \text{ if } n \geq 2 \end{aligned} \tag{4}$$

위에서

$$b_n = \varepsilon_{n-1} + \omega_{n-1} \quad (n \geq 2) \quad (5)$$

$$\begin{aligned} \omega_n &= 1 \text{ if } n=1 \\ &= (b_n)^2/4 \text{ if } n \geq 2 \end{aligned} \quad (6)$$

$$\begin{aligned} \varepsilon_n &= \bar{\varepsilon}_1 \text{ if } n=1 \\ &= \varepsilon_{n-1} \omega_{n-1} \text{ if } n \geq 2 \end{aligned} \quad (7)$$

위에서 n 은 일군의 행렬을 표시하기 위한 정수이며 $\bar{\varepsilon}_1$ 은 행렬 C_1 의 최소 eigenvalue이다. 위 식들에 의해서 다음의 관계가 성립한다.

$$\frac{\omega_n}{\varepsilon_n} = \frac{1}{4} \left(1 + \frac{\omega_{n-1}}{\varepsilon_{n-1}} \right)^2 \frac{\omega_{n-1}}{\varepsilon_{n-1}} \quad (8)$$

따라서 ω_n/ε_n 의 값은 n 이 증가함에 따라서 급격히 감소한다. 예로써 ω_1/ε_1 이 10^7 인 경우 $\omega_{14}/\varepsilon_{14}$ 은 1.02274에 불과하다. ω_n/ε_n 은 행렬 C_n 의 최대 eigenvalue와 최소 eigenvalue의 비보다 작으며 결국 n 이 증가하면 행렬 C_n 의 최대 eigenvalue와 최소 eigenvalue는 거의 같아진다.⁽¹²⁾ 따라서 다음장에서 사용되는 반복 행렬(iteration matrix)의 spectral radius를 급격히 감소시킬 수 있으며 계산 속도를 크게 증가시킬 수 있다. 또한 행렬 A_n 을 다음과 같이 정의한다.

$$\begin{aligned} A_n &= I \text{ if } n=1 \\ &= (b_n I - C_{n-1}) A_{n-1} \text{ if } n \geq 2 \end{aligned} \quad (9)$$

식 (4)와 (9)에 의하여 다음과 같은 관계가 성립한다.

$$C_n = C_1 A_n \quad (10)$$

3. 방정식 해법

3.1 일반적인 고찰

본 작업에서는 Lagrange multiplier λ 를 순수한 시간의 함수로 취급한다(즉 $\lambda = \lambda(t)$). 매 time step에서 Lagrange multiplier의 정확한 값은 반복 계산법(iterative scheme)에 의하여 계산되며 각 time step 사이의 Lagrange multiplier 값은 interpolation에 의하여 계산된다. 이와같은 방법에 의하여 Lagrange multiplier의 값은 시간 적분 이전에 주어진다.

본 작업의 주요 과제는 매 time step에서 운동 방정식 및 제한 조건 (2a)~(2c)를 만족시키는 λ , q , \dot{q} 와 \ddot{q} 를 계산하는 것이다. 현실적으로 디지털 컴퓨터를 이용한 계산에서는, 수치해석에 수반되는

각종 오차로 인하여, 제한조건 (2a)~(2c)중 어느 하나가 만족되어도 나머지 2조건이 반드시 만족되지 않는다는, 만약 제한조건 (2a)~(2c) 중 어느 하나라도 제대로 만족되지 않는 경우 계산의 안정성은 상실된다. 따라서 제한조건(2a)~(2c)를 모두 만족시키기 위한 특별한 대책이 필요하다. 이러한 목적을 위하여, 본 작업에서는 q 만 방정식 (1) 및 (2a)를 풀어서 구하고, \dot{q} 는 다음에 정의된 방정식(11)과 (2b)를 풀어서 구하며 \ddot{q} 는 방정식(11)과 (2c)를 풀어서 구한다.

$$M\ddot{q} + \bar{\Phi}'_q \lambda = f \quad (11)$$

위에서 $\bar{\Phi}'_q$ 는 방정식 (1) 및 (2a)를 풀어서 구한 q 를 사용한 경우의 Φ'_q 를 의미한다. 이와같은 방법을 사용하면 제한 조건(2a)~(2c)는 모두 만족이 된다. 특히 $\bar{\Phi}'_q$ 는 순수한 시간의 함수가 되어 다시 계산할 필요가 없으므로 \dot{q} 및 \ddot{q} 은 대단히 경제적으로 계산할 수 있다. 편의상 매 time step에서의 계산 기본 과정을 다음에 제시한다.

(1) 방정식 (1)과 (2a)를 풀어서 q 를 구한다.

(2) 방정식 (1)의 시간 적분 오차를 구한다(다음장에서 설명).

(3) 방정식 (11)과 (2b)를 풀어서 \dot{q} 를 구한다.

(4) 방정식 (11)과 (2c)를 풀어서 \ddot{q} 와 λ 를 구한다.

3.2 시간 적분

전술한 바와 같이, 매 time step마다, Lagrange multiplier의 값은 시간 적분 이전에 주어지며 Lagrange multiplier는 순수한 시간의 함수로 취급된다. 따라서 방정식 (1)과 (11)은 상미분 방정식이 되며 잘 알려진 시간 적분 방법을 이용하여 정확한 해를 구할 수 있다. 제한조건 Jacobian 행렬은 일반적으로 대단히 sparse 하며 본 작업의 경우 decomposition이나 orthonormalization 등의 복잡한 과정이 불필요하므로 대단히 경제적인 시간 적분이 가능하다. 시간 적분에서는 one step method의 채택이 가능하며, 이 경우 물체간의 충돌 등 시간에 대하여 극심하게 변하는 운동을 정확히 풀수있다.⁽¹³⁻¹⁴⁾

본 장에서는 time step i 와 $i+1$ 사이의 구간에서 방정식(1)의 시간 적분에 대하여 생각한다. 이미 time step i 까지의 해가 구해졌으며 현재 time step $i+1$ 의 해를 구하고 있다고 생각한다. 즉 시간 적

분이 t_i 와 t_{i+1} 사이에서 행하여 지고 있다고 생각 한다. (t_{i+1} 에서의 Lagrange multiplier 값은 주어 져 있다). 이 구간에서 Lagrange multiplier의 값 은 interpolation에 의하여 결정한다. 그러면 방정 식 (1)을 적분할 때 위치 q 에 포함되는 오차는 Lagrange multiplier의 interpolation으로 인하여 포함되는 오차와 순수한 시간 적분(예로써 Runge Kutta method)으로 인한 오차의 2가지가 된다. 순수한 시간 적분 오차는 잘 알려진 방법들(예로써 참고 문헌(15))에 의하여 쉽게 계산된다. Lagrange multiplier $\lambda(t)$ 가 k 회의 time step을 이용하여 interpolation되는 경우 그 interpolation 오차 $e(t)$ 는 다음과 같이 표현된다.⁽¹⁶⁾

$$e_j(t) = \frac{(t-t_{i-k+1}) \cdots (t-t_{i-1})}{(k+1)!} L_j^{(k+1)}(\xi_j) \tag{12}$$

위에서 ξ_j 는 interpolation 되는 시간 구간 내부에 존재하며, L 은 정확한 Lagrange multiplier 값이다. 일반적으로 interpolation 오차는 interpolation point의 수가 하나 더 증가한 경우의 interpolation 과 비교하여 계산하게된다. 이 목적을 위하여, time step size를 절반으로 줄인후, 방정식 (1)과 (2a)를 time step $i+1/2$ 에서 추가로 푼다. 이 추가점을 이용한 경우의 interpolation과 원래의 interpolation의 차이로써 interpolation 오차의 대부분을 계산할 수 있다. λ 의 interpolation 오차를 e 라고 하면, e 로 인하여 q 에 포함되는 위치 오차 s 는 다음 식을 적분하게 구하여 된다.

$$Ms + \Phi_s^0 e = 0 \tag{13}$$

한편 순수한 시간 적분 오차(예로써, Runge Kutta method로 인한 오차)를 PTI라고 하면, 방정식 (1)을 시간 적분할 때 위치 q 에 포함되는 전체 오차 ET는 다음과 같이 된다.

$$ET \leq \|s\| + PTI \tag{14}$$

Interpolation이, t_i 와 t_{i+1} 사이에서는, time step i 와 $i+1$ 의 값에 절대적인 영향을 받는다. 따라서 이 구간에서 interpolation 오차는 다음과 같이 근사적으로 표현할 수 있다.

$$e_j(t) \approx \frac{(t-t_i)(t-t_{i+1})}{2} L_j^{(2)}(\xi_j) \tag{15}$$

상기 근사적인 오차는 time step size 예측에 편리

하게 이용할 수 있다. 방정식 (13)과 (15)로부터, 짧은 구간 h 에서 Φ_s^0 와 $L^{(2)}(t)$ 의 변화는 무시할 수 있으므로, 위치 오차 s 는 h^4 에 비례하게된다. 따라서 방정식 (14)는 다음과 같이 쓸 수 있다.

$$ET \leq ah^4 + bh^N \tag{16}$$

위에서 a 와 b 는 상수이며 N 은 순수한 시간 적분 오차와 관계되는 지수이다(N 은 참고문헌 (15) 등에 상세히 설명되어 있으며, 참고문헌 (15)에 있는 program DOPRIN의 경우 $N=7$ 이다). 따라서 다음 time step에서의 step size인 h_{new} 는 다음 방정식을 풀어서 결정할 수 있다.

$$(TOL)_{int} = \|s\|(\text{fac})^4 + \text{PTI}(\text{fac})^N \tag{17}$$

$$\text{fac} = h_{new}/h \tag{18}$$

위에서 $(TOL)_{int}$ 는 사용자가 지정하는 시간 적분의 tolerance이다.

3.3 제한 조건을 만족시키기 위한 반복 계산법

Time step $i+1$ 에서의 정확한 Lagrange multiplier 값 λ_{i+1} 는 다음의 반복 계산법(iterative scheme)에 의하여 구해진다.

$$\lambda_{i+1}^m = \lambda_{i+1}^{m-1} - \alpha A_n v_{i+1}^{m-1} / \|C\|_\infty \tag{19}$$

위의 반복 계산법은 참고 문헌(12)에 소개된 반복 계산법과 본질적으로 동일하다. 위에서 $i+1$ 은 time step을, m 과 $m-1$ 은 반복 회수를 나타내며, α 는 상수이고 A_n 과 C 는 계산 가속을 위한 행렬이다. (α , A_n 과 C 는 다음 장에서 자세히 설명한다). v 는 제한 조건 오차이며 (3a)~(3c) 중 어느 하나에 의하여 정의된다. λ_{i+1}^{m-1} 가 주어지면 q_{i+1}^{m-1} , \dot{q}_{i+1}^{m-1} 과 \ddot{q}_{i+1}^{m-1} 은 방정식 (1)과 (11)의 해에 의하여 계산되며, 따라서 v_{i+1}^{m-1} 을 계산할 수 있다. 매 time step에서, 방정식 (1), (11)과 (2a)~(2c)를 풀기 위하여, 세가지 종류의 제한 조건 오차가 사용된다. 즉 방정식 (1)과 (2a)를 풀 때에는 위치 오차 (3a)가 사용되고, 방정식 (11)과 (2b)를 풀 때에는 속도 오차 (3b)가 사용되고, 방정식 (11)과 (2c)를 풀 때에는 가속도 오차 (3c)가 사용된다. 반복 계산법 (19)는 각각의 제한 조건 오차가 사전 지정된 tolerance보다 작게될 때까지 반복된다.

3.4 반복 계산법의 수렴에 대한 분석

본 장에서는 time step $i+1$ 에서의 반복 계산법

(19)에 대하여 분석한다. 편의상 하첨자 $i+1$ 은 생략한다. Tiem step size h 와 상수 α 는 상당히 작은 값이라고 생각한다. (실제 계산시, 수렴에 필요한 경우, h 와 α 는 충분히 작게 만들어 진다.) 여기에서는 반복 계산법(19)가 (3a)-(3c) 중의 어느 제한 조건 오차도 모두 0으로 축소 시킴을 증명하고자 한다.

먼저, 반복 계산법(19)에서 위치 오차(3a)를 사용한 경우에 대하여 분석한다. 이 경우는 방정식(1)과 (2a)를 푸는 경우이다. α 값이 작은 경우, (3a)로부터 다음의 근사식이 얻어진다.

$$v^m - v^{m-1} = -\Phi^m + \Phi^{m-1} = -\Phi_q(q^m - q^{m-1}) \quad (20)$$

짧은 구간 h 에서 비선형의 고차 항을 생략하면 가속도는 거의 선형적으로 변화하므로

$$q^m - q^{m-1} = \frac{h^2}{6}(\ddot{q}^m - \ddot{q}^{m-1}) \quad (21)$$

식(1)로부터

$$\ddot{q}^m - \ddot{q}^{m-1} = -M^{-1}\Phi_q^t(\lambda^m - \lambda^{m-1}) \quad (22)$$

식(20)와 (22)로부터

$$v^m - v^{m-1} = \frac{h^2}{6}\Phi_q M^{-1}\Phi_q^t(\lambda^m - \lambda^{m-1}) \quad (23)$$

식(19)와 (23)으로부터

$$v^m - v^{m-1} = \frac{-\alpha h^2}{6}\Phi_q M^{-1}\Phi_q^t A_n v^{m-1} / \|C\|_\infty \quad (24)$$

행렬 C_n 을 다음과 같이 정의하면

$$C_n = \frac{h^2}{6\|C\|_\infty}\Phi_q M^{-1}\Phi_q^t A_n \quad (25)$$

식(24)는 다음과 같이 변환된다.

$$v^m = (I - \alpha C_n)v^{m-1} \quad (26)$$

식(4), (9)와 (10)으로부터, 식(4)의 행렬 C 가 다음과 같이 정의된다면, (25)의 행렬 C_n 은 식(4)에 의하여 얻어진다.

$$C = \frac{h^2}{6}\Phi_q M^{-1}\Phi_q^t \quad (27)$$

그러면 행렬 C_n 은 행렬 C 가 singular가 아닌한 대칭이고 positive definite가 된다. 행렬 C 가 singular인 경우는 별도의 추가 대책이 필요하므로 본 작업에서는 행렬 C 가 nonsingular인 경우만 생각한다. 그러면 α 가 다음 조건을 만족시키면

$$0 < \alpha < \frac{2}{(\Lambda_n)_{max}} \quad (28)$$

다음과 같은 부등식이 성립한다.

$$\|v^m\|_2 < \|v^{m-1}\|_2 \quad (29)$$

위에서 $(\Lambda_n)_{max}$ 는 행렬 C_n 의 최대 eigenvalue이다. 따라서 위치 오차(3a)는 반복 계산법(19)에 의하여 0을 향하여 단조 감소하게 된다.

이제 반복 계산법(19)에 속도 오차 (3b)를 사용하는 경우에 대하여 검토한다. 이 경우에는 방정식(11)과 (2b)를 풀게되며 Φ_q 은 $\bar{\Phi}_q$ 가 된다. (3b)로부터, 짧은 구간 h 에서는 q 의 변화가 \dot{q} 의 변화보다 훨씬 작으므로, 다음 식이 얻어진다.

$$v^m - v^{m-1} = \bar{\Phi}_q(\dot{q}^m - \dot{q}^{m-1}) \quad (30)$$

짧은 구간 h 에서 비선형의 고차 항을 생략하면 가속도는 거의 선형적으로 변화하므로

$$\dot{q}^m - \dot{q}^{m-1} = \frac{h}{2}(\ddot{q}^m - \ddot{q}^{m-1}) \quad (31)$$

식(11), (19), (30)과 (31)로부터

$$v^m - v^{m-1} = \frac{-\alpha h}{2}\bar{\Phi}_q M^{-1}\bar{\Phi}_q^t A_n v^{m-1} / \|C\|_\infty \quad (32)$$

식(4)의 행렬 C 가 다음과 같이 정의된다면

$$C = \frac{h}{2}\bar{\Phi}_q M^{-1}\bar{\Phi}_q^t \quad (33)$$

식(32)는 식(26)의 형태로 변환되며, 따라서 속도 오차(3b)는 반복 계산법(19)에 의하여 0을 향하여 단조 감소하게 된다.

마지막으로, 반복 계산법(19)에 가속도 오차(3c)를 사용하는 경우에 대하여 검토한다. 이 경우에는 방정식(11)과 (2c)를 풀게되며, Φ_q 는 $\bar{\Phi}_q$ 가 된다. 짧은 구간 h 에서 q 와 \dot{q} 의 변화는 \ddot{q} 의 변화보다 훨씬 작으므로, (3c)로부터 다음과 같은 근사식이 얻어진다.

$$v^m - v^{m-1} = -\bar{\Phi}_q(\ddot{q}^m - \ddot{q}^{m-1}) \quad (34)$$

식(11)과 (34)로부터

$$v^m - v^{m-1} = \alpha \bar{\Phi}_q M^{-1} \bar{\Phi}_q^t (\lambda^m - \lambda^{m-1}) \quad (35)$$

식(19)와 (35)로부터

$$v^m - v^{m-1} = \alpha \bar{\Phi}_q M^{-1} \bar{\Phi}_q^t A_n v^{m-1} / \|C\|_\infty \quad (36)$$

식(4)의 행렬 C 가 다음과 같이 정의된다면

$$C = \bar{\Phi}_q M^{-1} \bar{\Phi}_q^t \quad (37)$$

식(36)은 식(26)의 형태로 변환되며, 가속도 오차(3c)는 반복 계산법(19)에 의하여 0을 향하여 단

조 감소하게 된다. 따라서 (3a)~(3c)에서 정의된 제한조건 오차는 모두 반복 계산법(19)에 의하여 지정된 tolerance 이하로 줄일 수 있으며 기구학적 제한조건(2a)~(2c)를 모두 만족시킬 수 있다.

3.5 계산 가속 기법

α 의 값이 다음과 같이 주어질 경우에

$$\alpha = \frac{2}{(\Lambda_n)_{max} + (\Lambda_n)_{min}} \quad (38)$$

식 (26)의 반복 행렬 (iteration matrix)은 다음과 같은 최소의 spectral radius를 갖게된다.

$$\rho(I - \alpha C_n) = \frac{(\Lambda_n)_{max} - (\Lambda_n)_{min}}{(\Lambda_n)_{max} + (\Lambda_n)_{min}} \quad (39)$$

위에서 Λ_n 은 행렬 C_n 의 eigenvalue를 나타낸다. 2.2장에서 논의된 바와 같이 $(\Lambda_n)_{max}/(\Lambda_n)_{min}$ 은 n 이 증가함에 따라서 1을 향하여 감소한다. 그러면 식 (39)에 의하여 제한 조건 오차의 감소 속도는 n 이 증가함에 따라서 급속히 증가하게 되며 계산 가속 효과를 얻을 수 있다. 비록 비선형 방정식의 경우에는 선형 방정식에 비하여 계산가속 효과가 떨어지기는하나, 작은 구간 h 에서 행렬 C 의 변화가 크지 않는 이상 계산 시간을 현저히 단축시킬 수 있다.

참고문헌(12)에 의하면, 행렬 A_n 을 직접 계산하지 않고 행렬과 벡터의 연속적인 곱에 의하여 간접적으로 계산하는 경우에, 본 작업에 소개된 계산 가속 기법은 Chebyshev acceleration technique과 동등한 수준이 된다. 본 작업의 경우에는, 행렬 C 의 0이 아닌 행이 고정되어 있으므로 행렬 A_n 을 반복 계산(iteration) 초기에 직접 계산하는 것(즉 2.2장에서 정의되대로 행렬 A_n 을 직접 계산)이 가능하다. (참고문헌 (12)에서는 점축 조건으로 인하여 0이 아닌 행열 자체가 미지수이므로 행렬 A_n 을 계산 전에 직접 계산하는 것이 불가능하다). 그러면 행렬 A_n 은 그 time step에서 다시 계산할 필요가 없으며 계산 시간은 크게 절감된다. 행렬 A_n 은 conjugate gradient method의 preconditioning 행렬과 비슷한 역할을 하고 있으므로 다음과 같은 방법에 의하여 계산 시간을 추가로 단축시킬 수 있다.

(가) 식 (4)에 의하여, 행렬 C 가 C_1 으로 normalize된 경우, 행렬 C_1 은 행렬 C 가 식 (27), (33)과 (37)중 어느 것에 의하여 정의되더라도 모

두 실질적으로 동일한 행렬이 된다. 따라서 행렬 C 는 식 (27)에 의하여만 계산한다.

(나) 전 time step에서 계산한 행렬 C_1 을 현 time step에서 이용하여도 계산 속도가 크게 떨어지지 않는 경우, 행렬 C_1 을 추가로 계산하지 않는다.

4. 실제적인 계산 방법

실제 계산시 계산 가속을 위하여 α 의 값은, 식 (38) 대신에, 다음과 같이 주어진다.

$$\alpha = \frac{2}{\omega_n + \varepsilon_n} \quad (40)$$

위에서 ω_n 과 ε_n 은 2.2장에서 정의된 식들에 의하여 주어진다. 효율적인 계산을 위하여는 식 (7)의 ε_1 값은 행렬 C_1 의 최소 eigenvalue와 상당히 가까운 값이어야 한다. ε_1 을 계산할 필요가 있는 경우에는, time step 초기의 j 회 반복 계산(iteration)을 이용하여, n 과 α 의 값을 각각 1과 1.9로 놓고서 다음 식에 의하여 계산한다.

$$\varepsilon_1 = \min_{1 \leq m \leq j} \frac{1}{\alpha} \left(1 - \frac{\|v^m\|_2}{\|v^{m-1}\|_2} \right) \quad (41)$$

현실적인 계산시 $j=10$ 정도면 충분히 정확한 값을 구할 수 있다. 위에서는 편의상 α 를 1.9로 고정하였으며 대부분의 모델에서 만족할만한 수렴속도가 얻어진다. 만약 ε_1 의 상당히 정확한 근사치인 ε_1^* 을 알고 있다면 (예로써 전 time step의 ε_1 값 이용) $1 \leq m \leq j$ 인 구간에서는 α 를 $2/(1 + \varepsilon_1^*)$ 로 고정하면 수렴 속도가 약간 증대하나 전체 계산 시간에 미치는 영향은 무시할 수준이다. 계산 가속 기법이 적용되는 경우, 대부분의 모델에 대하여, n 의 값이 대략 10부근이면 충분히 빠른 계산 속도를 얻을 수 있으므로 n 의 값을 특별히 조정할 필요는 없다. (필요한 경우에는, n 을 증가 또는 감소시켜가며 계산 시간을 비교하여 n 의 값을 조정할 수 있다.) Time step $i+1$ 에서의 계산 과정을 요약하면 다음과 같다.

(1) Time step $i+1$ 에서 방정식 (1)과 (2a)를 푼다.

1.1. m 을 0으로 한다.

1.2. ε_1 을 계산할 필요가 있는 경우, n 과 α 값을

각각 1과 1.9로 놓는다. 그렇지 않은 경우 $n=10$ 으로 한다. $m=0$ 이면 step 1.4로 가고 (매 time step의 초기마다 λ 는 전 time step에서 결정한 λ 값 이용. 특히 $t=0$ 에서는 식(1) 및 식(2a)를 두 번 미분한 식의 연립 방정식에 초기조건을 적용하여 $\lambda(0)$ 를 결정한다), 그렇지 않으면 step 1.3으로 간다.

1.3. 반복 계산법(19)를 이용하여 λ^m 을 계산한다.

1.4. 방정식 (1)을 적분하여 q^m 을 계산한다.

1.5. (3a)에 의하여 위치 제한 오차 v^m 을 계산한다.

1.6. 만약 $\|v^m\|_2 < \|v^{m-1}\|_2$ 또는 $m=0$ 이면 step 1.7로 간다. 그렇지 않은 경우, $\alpha = \alpha/2$ 로 하여 step 1.3으로 다시 가서 m 번째의 계산을 반복한다. 만약 α 가 상당히 작은 값으로 감소된 뒤에도 $\|v^m\|_2$ 이 $\|v^{m-1}\|_2$ 보다 작지 않으면 time step size를 절반으로 줄여 가지고 step 1.1로 가서 이 time step의 계산을 처음부터 반복한다.

1.7. $\|v^m\|_2 \geq (\text{TOL})_{im}$ 이면 step 1.8로 간다. $\|v^m\|_2 < (\text{TOL})_{im}$ 이면 순수 시간 적분 오차 PTI를 계산한후 step 2로 간다.

1.8. 만약 $\bar{\epsilon}_1$ 을 계산할 필요가 있으면 (예로서, 필요한 time step에서의 $m=10$), $\bar{\epsilon}_1$, α 와 A_n 을 (41), (40)과 (9)에 의하여 계산한다. Step 1.9로 간다.

1.9. m 이 너무 크지 않으면 (예로써 $m < 50$) $m = m+1$ 으로 하여 step 1.2로 간다. m 이 너무 크면(즉 수렴 속도가 너무 느리면) time step size를 절반으로 하여 step 1.1로 가서 이 time step의 계산을 처음부터 반복한다.

(2) Time step $i+1/2$ 에서의 (즉 time step size를 절반으로 줄여서) 방정식 (1)과 (2a)를 풀 후, interpolation으로 인한 시간 적분 오차 s 를 식(13)에 의하여 계산한다. 방정식 (1)과 (2a)를 풀기 위하여는 step 1과 비슷한 방법을 사용한다. Step 1에서 구한 PTI와 위의 s 에 의하여 시간 적분에 포함된 전체 오차 ET와 다음 번의 time step size인 h_{new} 를 (16)~(18)에 의하여 구한다. 만약 ET의 상한치가 사용자가 지정한 tolerance 값보다 작으면 step 3으로 가고, 그렇지 않은 경우에는 step 1로 가서 이 time step에서의 계산을 새로운 time step size인 h_{new} 를 가지고 처음부터 반복한다.

(3) Step 1과 비슷한 방법에 의하여, time step $i+1$ 에서의 방정식 (11)과 (2b)를 풀어서 속도 \dot{q} 를 계산한다. 제한조건 오차 (3b)를 사용한다.

(4) Step 1과 비슷한 방법이 의하여, time step $i+1$ 에서의 방정식 (11)과 (2c)를 풀어서 가속도 \ddot{q} 를 계산한다. 제한 조건 오차 (3c)를 사용한다.

5. 예제 계산

5.1 예제 1

Fig. 1은 3개의 강체로 구성된 크랭크-슬라이더 기구이다. 관련 데이터는 $M_A=100$, $J_A=33.33$, $L_A=2$, $M_B=100$, $J_B=10$, $M_C=300$, $J_C=25$, $L_C=1$ 이고, $F=\min(1000, 10000t)$ 이다. 여기에서 M , J , L 과 F 는 각각 질량, 관성 모멘트, 길이와 힘을 나타낸다. $t=0$ 에서 물체 A는 수평선과 30° 각도를 이루고 있다. 방정식 (1)을 적분하기 위해서는 참고문헌 (15)에 소개된 프로그램 DOPRIN을 약간의 수정을 거쳐 사용하였다. DOPRIN은 explicit Runge-Kutta Nystrom method에 의하여

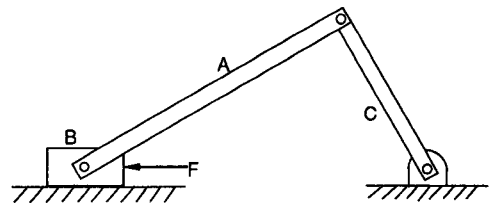


Fig. 1 Crank-slider mechanism

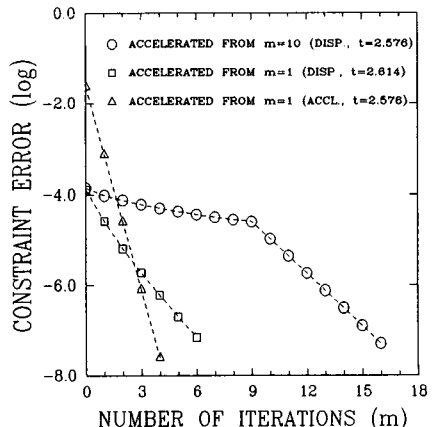


Fig. 2 Typical trend of constraint error reduction with the iteration numbers

시간 적분을 행한다. 시간 적분 tolerance와 제한 조건 tolerance는 각각 10^{-5} 와 10^{-7} 으로 주어졌다. 계산시 세가지 (즉 위치, 속도, 가속도) 종류의 제한 조건이 모두 만족되도록 하였다. 계산 가속을 위한 행렬 A_n 은, 전 time step에서 반복 계산법 (19)가 가속 기능을 사용하여 10회 이상 반복된 경우, $m=9$ 에서 계산하였다. 계산 가속 기법이 적용될 경우에는 n 을 15로 고정하였다. Fig. 2에 예시된 바와 같이, 제한 조건의 오차는 계산 가속 기법에 의하여 급격히 감소하게 된다. 전체 운동 에너지의 변화가 Fig. 3에 표시되어 있으며 물체 A의 각가속도의 변화가 Fig. 4에 표시되어 있다. 본 계산시 운동 에너지 최고치의 변화는 0.01% 정도이다. 그런데 속도 제한 오차 (3b)와 가속도 제한 오차(3c)의 경우에는 그 tolerance를 각각 $10^{-7}/h$ 와 $10^{-7}/h^2$ 으로 낮추어도 거의 동일한 결과가 얻어진다(여기에서 h 는 time step size).

(2a)-(2c)의 세 가지 형태의 제한 조건을 모두 만족시키는 것이 필요한가를 점검하기 위하여, 가속도 제한 조건 (2c)만을 적용하여 문제를 풀어

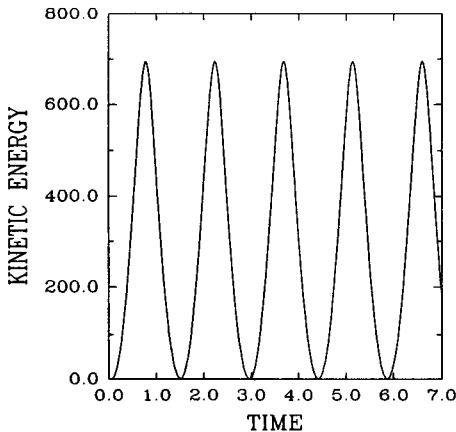


Fig. 3 Variation of the total kinetic energy of the crank-slider mechanism

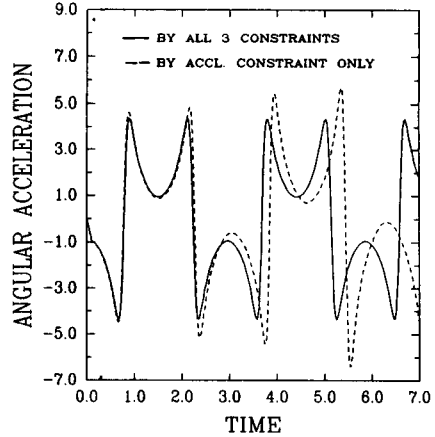


Fig. 4 Comparison of the angular acceleration of body A: (1) obtained by applying all the three constraints, and (2) obtained by applying only the acceleration constraint at each time step

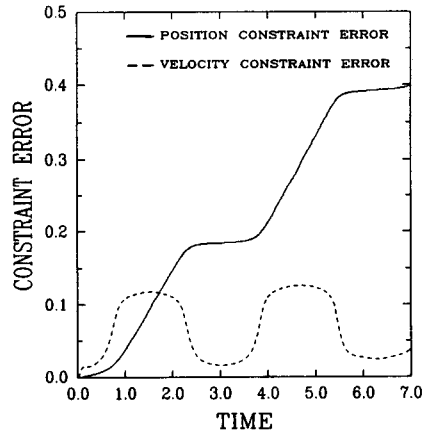


Fig. 5 Variation of the constraint errors when the solution is obtained by applying only the acceleration constraint

보았다. 즉 q, \dot{q} 와 \ddot{q} 를 모두 방정식 (1)과 (2c)를 풀어서 구하였다. 이렇게 하여 구한 가속도가 Fig. 4에 표시되어 있는데, 시간이 갈수록 오차는 점점

Table 1 Comparison of CPU time (sec) with the different values of n for $0 \leq t \leq 7$ ($(TOL)_{int}=10^{-5}$, $(TOL)_{cons}=10^{-7}$)

Cases	$n=1$	$n=6$	$n=9$	$n=15$
$M_A=100, J_A=33.33$	53.2	7.6	8.2	8.6 (194 steps)
$M_A=10, J_A=3.333$	545.1	18.1	19.0	22.1 (350 steps)
$M_A=1, J_A=0.3333$	30분 이상	98.7	40.5	45.3 (769 steps)

증가하고 있다. 이러한 오차는 위치 및 속도의 제한 조건 (2a)~(2b)가 Fig. 5에 보이는 바와 같이 만족되지 않기 때문이다. 만약 제한 조건 (2a) 또는 (2b)만 만족이 되고 다른 조건은 만족이 되지 않는 경우에는 보다 더 큰 오차가 발생하게 된다.

계산 가속 기법의 효율성을 점검하기 위하여 몇 가지 경우에 대하여 n 을 변화시켜 가며 계산 시간을 비교하였다. 여기에서는 물체 A의 질량만을 변화시켰으며 나머지 데이터는 변하지 않았다. 계산 가속을 위한 행렬 A_n 은 필요한 time step의 $m=9$ 에서 계산하였으며, 계산 가속시 n 값은 고정되었다. 시간 적분 및 제한 조건의 tolerance는 전과 같이 10^{-6} 와 10^{-7} 으로 하였다. $0 \leq t \leq 7$ 동안의 운동을 위한 계산을 Cyber 910 workstation을 이용하여 double precision으로 수행하였다. 소요 계산 시간은 Table 1에 표시되어 있는데 계산 가속 기법에 의하여 계산 시간이 대폭 감소됨을 알 수 있다. 본 작업의 계산 가속 기법은 수렴 속도가 느린 경우에 특히 효율적이다.

5.2 예제 2

Fig. 6은 토션바를 현수장치로 장비한 육상 차량이며 3개의 강체로 구성되어 있다. 샤시의 질량은 1,000kg이며 피치 관성 모우먼트는 $2,000\text{kg}\cdot\text{m}^2$ 이다. 토션바암의 질량은 100kg, 관성 모우먼트는 $10\text{kg}\cdot\text{m}^2$, 길이는 0.8m이며 질량 중심은 윗 회전 연결점 (즉 토션바)으로부터 0.5333m되는 지점에 있다. 샤시의 질량 중심에 대한 회전 연결점 (즉 토션바)의 상대 위치는 각각 $(-0.15\text{ m}, -0.5\text{ m})$ 와 $(0.28197\text{ m}, -0.15\text{ m})$ 이다. 토션바의 스프링 상수는 $6922.9\text{ N}\cdot\text{m}$ 이며, 초기 평형 상태에서 토션바는 수평과 45° 를 이룬다. 타이어와 지면의 접촉력은 Fig. 7에서 보인 바와 같이 스프링을 모델로 하여 계산하며 접촉력은 지면 또는 장애물의 표면에 수직으로 작용한다고 생각한다. 타이어 하나에는 최대 2곳 (즉 지면과 장애물)의 접촉점이 존재한다. 접촉점에서의 타이어의 변형을 δ 라고 하면,

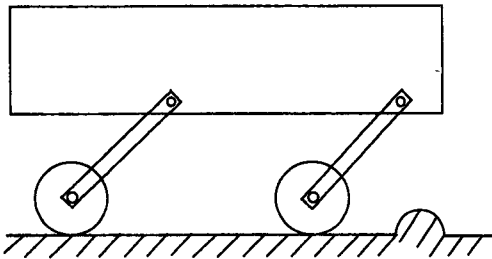


Fig. 6 Vehicle with the torsion bar suspension system

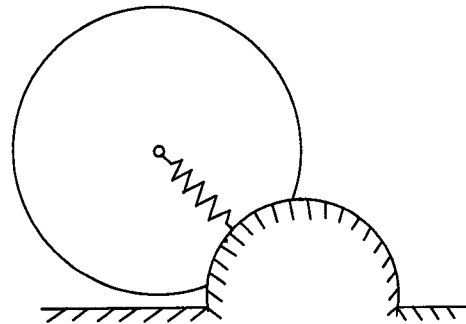


Fig. 7 Tire model

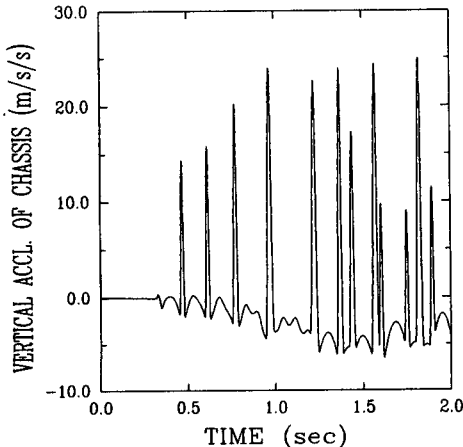


Fig. 8 Vertical acceleration of chassis

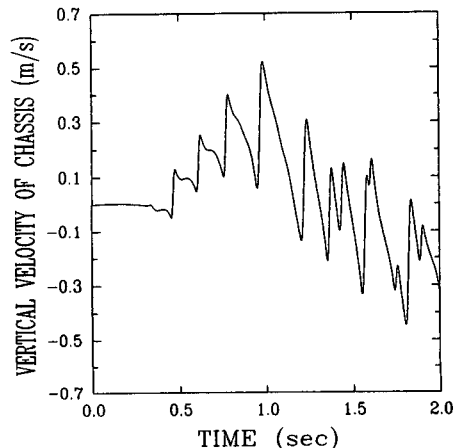


Fig. 9 Vertical velocity of chassis

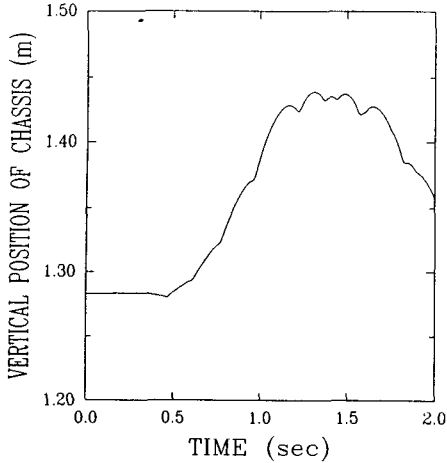


Fig. 10 Vertical position of chassis

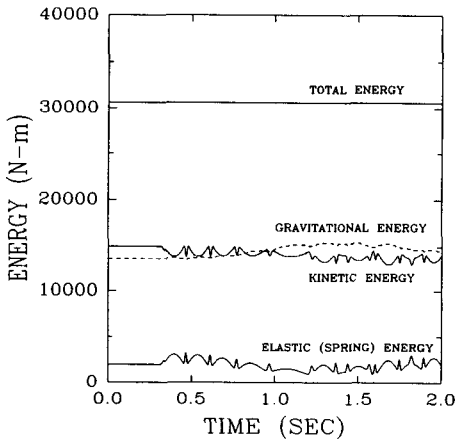


Fig. 11 Variation of the energy of the vehicle

타이어의 접촉력 p 는 $p = k\delta^2$ 에 의하여 계산하기로 하며, 여기에서 k 는 $2 \cdot 10^7$ 로 한다. 타이어의 원래 반경은 0.4m이며 장애물은 반경 0.1 m의 반원이다. 차량의 초기 속도는 5 m/sec이며 $t = 0.315$ sec에서 장애물과 충돌하기 시작한다. 프로그램 DOPRIN을 약간 수정하여 시간 적분에 이용하였으며, 시간 적분과 제한 조건의 tolerance는 각각 10^{-8} 과 10^{-10} 으로 하였다. 사시 질량 중심의 수직 방향 가속도, 속도 및 위치가 Fig. 8~10에 표시되어 있다. 본 예제에는 댐핑이 포함되어 있지 않으며, 타이어는 장애물과 충돌한 후 끝없이 지면과 충돌-분리의 동작을 반복한다. 토션바의 반력에 의하여 타이어가 지면을 칠 때마다 가속도는 심한 변화를 보인다. 에너지의 변화가 Fig. 11에 표시되어

있는데, 전체 에너지 (30563 N-m)는 계산 중 0.005 % 이내의 변화를 보인다.

6. 결론

기구학적 구속 조건을 가진 다물체 동역학을 풀기 위한 새로운 방법을 소개하였다. 정확한 Lagrange multiplier의 값을 반복 계산법에 의하여 구했으며, 운동방정식은 상미분 방정식으로 변화시켜 잘 알려진 시간 적분 방법을 적용하였다. 제한 조건 오차는 반복 계산법에 의하여 단조 감소하며 계산 가속 기법이 적용되는 경우에는 급격히 감소함을 이론 및 예제 계산을 통하여 입증하였다. 위치, 속도 및 가속도의 세가지 종류의 제한 조건을 모두 매 time step에서 만족시킴으로써 계산의 안정성을 유지하였다. 예제 계산을 통하여 본 방법의 정확성을 입증하였다. 특히 계산시 Newton-Raphson method나 행렬의 decomposition을 사용하지 않았다.

본 작업에서는, 다른 대부분의 강체 동역학 참고 문헌과 마찬가지로, 두 강체의 충돌로 인한 접촉력을 계산할 경우에 질량이 없는 스프링이 두 물체 사이에서 압축 된다고 가정하여 문제를 풀었다. 그런데 질량을 가진 변형체(deformable body)들이 충돌하는 경우에는 이러한 스프링을 사용하여 단순화 시킬 수 없으며 또한 실제의 접촉점들이 시간 적분 이전에 결정되어야 한다. 변형체가 충돌하는 경우에 접촉력은 Lagrange multiplier와 비슷한 역할을 하게되며 접촉점에서는 세가지 종류의 기하학적 적합성(즉 변위, 속도, 가속도)이 만족되어야 한다. 현재까지의 예비적 계산에 의하면, 변형체로 구성된 다물체의 충돌 현상은 본 작업과 참고 문헌(12)의 기법을 적용하여 정확히 풀 수 있다고 판단되며 추후에 계속적인 작업을 할 예정이다. 또 본 작업에서 제시한 계산 방법을 이용하여 추후에 차량 동역학 및 통신위성과 같은 구체적인 공학 문제를 푸는데 응용할 예정이다.

계산 시간은 사용되는 시간 적분법의 종류 및 전산 프로그램 기법에 의하여 추가로 단축될 수 있으며 이에 대한 연구 검토가 필요하다고 판단된다. 한편 본 계산 방법의 경우, 대부분의 계산 시간은 행렬과 벡터를 곱하는데 소요되며 따라서 병렬 계산 방법이 효율적으로 사용되리라고 판단된다.

참고문헌

- (1) Petzold, L., 1982, "Defferential Equations Are Not ODEs," SIAM J. Sci. Stat. Comput., Vol. 3, pp. 367~384.
- (2) Petzold, L., 1989, "Recent Developments in the Numerical Solution of Differential/Algebraic Equations," Comput. Methods Appl. Mech. Engrg., Vol. 75, pp. 77~89.
- (3) Orlandea, N., Chace, M.A. and Calahan, D.A., 1977, "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems-Part I and II," ASME J. Engrg. Ind., Vol. 99, pp. 773~784.
- (4) Cardona, A. and Geradin, M., 1989, "Time Integration of Equations of Motion in Mechanism Analysis," Comput. Struct., Vol. 33, pp. 801~820.
- (5) Kamman, J.W. and Huston, R.L., 1984, "Dynamics of Constrained Multibody Systems," ASME J. Appl. Mech., Vol. 51, pp. 899~903.
- (6) Singh, R.P. and Likins, P.W., 1985, "Singular Value Decomposition for Constrained Dynamical Systems," ASME J. Appl. Mech., Vol. 52, pp. 943~948.
- (7) Kim, S.S. and Vanderploeg, M.J., 1986, "QR Decomposition for State Space Representation of Constrained Mechanical Dynamic Systems," ASME J. Mech. Trans. Auto. Des., Vol. 108, pp. 183~188.
- (8) Liang, C.G. and Lance, G.M., 1987, "A Differentiable Null Space Method for Constrained Dynamic Analysis," ASME, J. Mech. Trans. Auto. Des., Vol. 109, pp. 405~411.
- (9) Ider, S.K. and Amirouche, F.M.L., 1988, "Coordinate Reduction in the Dynamics of Constrained Multibody Systems-A New Approach," ASME J. Appl. Mech., Vol. 55, pp. 899~904.
- (10) Wehage, R.A. and Haug, E.J., 1982, "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," ASME J. Mech. Des., Vol. 104, pp. 247~255.
- (11) Park, T.W. and Haug, E.J., 1986, "a Hybrid Numerical Integration Mehtod for Machine Dynamic Simulation," ASME J. Mech. Trans. Auto. Des., Vol. 108, pp. 211~216.
- (12) Lee, K., 1989, "An Accelerated Iterative Method for Contact Analysis," Int. J. Numer. Methods Eng., Vol. 28, pp. 289~293.
- (13) Cash, J.R. and Karp, A.H., 1990, "A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides," ACM Trans. Math. Softw., Vol. 16, pp. 201~222.
- (14) Addison, C.A., Enright, W.H., Gaffney, P.W., Gladwell, I. and Hanson, P.M., 1991, "Algorithm 687 A Decision Tree for the Numerical Solution of Initial Value Ordinary Differential Equations," ACM Trans. Math. Softw., Vol. 17, pp. 1~10.
- (15) Hairer, H., Norsett, S.P. and Wanner, G., 1987, "Solving Ordinary Differential Equations I," Springer-Verlag, Berlin.
- (16) Atkinson, K.E., 1989, "An Introduction to Numerical Analysis," John Wiley & Sons, New York.