

차량 실시간 시뮬레이션을 위한 암시적 수치 알고리즘

Implicit Numerical Algorithm for Real-time simulation of a Vehicle

박 민 영*, 이 정 근*, 송 창 섭*, 배 대 성**
M. Y. Park, J. K. Lee, C. S. Song, D. S. Bae

ABSTRACT

In this reaserch, a program for real time simulation of a vehicle is developed. This program uses relative coordinates to save the computation time and BDF(Backward Difference Formula) to integrate system variables. Numerical tests were performed for J-turn and Lane change steering, respectively. The validity of the program is proved by the ADAMS package. Numerical results showed that the proposed implicit method is more stable in carrying out the numerical integration for vehicle dynamics than the explicit method. Hardware requirements for real time simulation are suggested.

주요기술용어 : Implicit numerical integration(암시적수치적분), BDF(Backward Difference Formula), Real-time-simulation(실시간 시뮬레이션)

1. 서 론

컴퓨터 기술의 눈부신 발달에 힘입어 복잡한 기계시스템의 실시간 시뮬레이션이 가능해지면서 최근에는 자동차 분야에서도 실시간 시뮬레이터가 개발되어 극한 상황 대체시험이나 차량주행시 안정성 확보시험 등에 확대 응용되어지고 있

다. 컴퓨터 프로세서가 빨라짐에 따라서 더욱 복잡하고 실제시스템에 가까운 모델을 채택하고 있으나 이는 다시 시뮬레이션 시간을 증가시켜 실시간내의 해석을 어렵게 한다. 지금까지 개발된 실시간 차량 시뮬레이터에는 차량동역학 시뮬레이션시 간략화된 차량모델을 사용하여 차량 시뮬레이션 정확도가 떨어지거나, 대단히 계산량이 많은 차량 모델을 사용하므로써 초대형 컴퓨터를 필요로 하는 문제점이 있다. 즉, 실시간 시뮬레이션에서는 시뮬레이션의 정확도와 속도라는

* 한양대학교 정밀기계공학과

** 정회원, 한양대학교 정밀기계공학과

서로 상반되는 요구를 동시에 만족시켜야 하기 때문에, 비실시간 시뮬레이션과는 다른 상황이 고려 되어져야한다. 본 연구에서는 일반 좌표로써 상대좌표를 채택하고, 적분방법으로 암시적 수치 적분방법 중의 하나인 BDF를 이용했다. 계산시간의 대부분을 차지하는 자코비안(Jacobian), 레지듀얼(Residual), 역치환(Back-Substitution)의 계산 횟수에 따른 한 스텝당 평균 시간을 계산하여 실시간에 필요한 하드웨어 기종을 제시하며 그 가능성을 보이고자 한다.

2. 운동방정식(Motion of Equation)

차량시스템의 뉴턴-오일러(Newton-Euler) 운동방정식을 변분형태로 쓰면 다음과 같이 나타낼 수 있다.¹¹⁾

$$\delta Z^T \cdot (M\dot{Y} - Q + \phi_Z^T \lambda) = 0 \quad (2.1)$$

여기서 δZ 는($\in R^{nc}$, nc는 직교좌표수) 모든 기구학적 구속조건과 부합하여야 하며 ϕ 는 절단조인트 구속조건식이고, λ 는($\in R^m$, m은 cut 구속조건수) Lagrange 승수이며, M 은 질량행렬이고, Q 는 일반힘이다. 식(2.1)에 직교좌표와 상대좌표의 관계식 $\delta Z = B \delta q$ 를 대입하면 상대좌표계에서의 운동방정식이 얻어진다.¹⁰⁾

$$\delta q^T [B^T (M\dot{Y} + \phi_Z^T \lambda - Q)] = 0 \quad (2.2)$$

여기서 δq 는 임의가 되므로 운동방정식은 다음과 같이 얻어진다.

$$F(q, v, \dot{v}, \lambda) = B^T (M\dot{Y} + \phi_Z^T \lambda - Q) = 0 \quad (2.3)$$

여기서 q 와 v 는 각각 상대좌표, 상대속도를 나타낸다.

3. BDF를 이용한 이산화

이 장에서는 운동방정식과 기구학적 구속조건에서 발생하는 지배방정식이 어떻게 ODAE

(Overdetermined Differential Algebraic Equation)로 정의되는지 서술하며, 정의된 ODAE의 풀이 방법에 대하여 논의하고 그 해법을 설명한다.

3.1 ODAE의 정의

2장에서 정의된 기구학적 구속조건이 있는 운동방정식은 다음과 같이 암시적으로 쓸 수 있다.

$$v - \dot{q} = 0 \quad (3.1a)$$

$$F(q, v, \dot{v}, \lambda) = 0 \quad (3.1b)$$

$$\phi(q) = 0 \quad (3.1c)$$

식 (3.1c)를 연속적으로 시간에 대하여 미분하면,

$$\dot{\phi}(q, v) = \phi_q v - \dot{v} = 0 \quad (3.2a)$$

$$\ddot{\phi}(q, v, \dot{v}) = \phi_q \dot{v} - \ddot{v} = 0 \quad (3.2b)$$

여기서 식 (3.2a)는 속도 구속조건이고, 식 (3.2b)는 가속도 구속조건이다. 식 (3.1a)부터 식 (3.2b)까지를 모두 결합하면 다음과 같은 ODAE를 만들 수 있다.

$$H(x) = \begin{vmatrix} F(x) \\ \phi \\ \phi \\ \phi \\ U_0^T \left(\frac{h}{b_0} R_1 \right) \\ U_0^T \left(\frac{h}{b_0} R_2 \right) \end{vmatrix} = \begin{vmatrix} F(q, v, \dot{v}, \lambda) \\ \phi_q v - \dot{v} \\ \phi_q v - \dot{v} \\ \phi(q) \\ U_0^T \left(\frac{h}{b_0} \dot{v} - v - \xi_1 \right) \\ U_0^T \left(\frac{h}{b_0} \dot{q} - q - \xi_2 \right) \end{vmatrix} = 0 \quad (3.3)$$

여기서 h 는 적분구간이며 $\xi_1 = \frac{1}{b_0} \sum_{i=1}^k b_i v_{(k-i)}$,

$\xi_2 = \frac{1}{b_0} \sum_{i=1}^k b_i q_{(k-i)}$ 이며 k 는 적분차수, b_i 는

BDF 계수이다. 또 U_0 ($\in R^{nr \times (nr-m)}$, nr 은

상대좌표수의 열은 위치 구속조건 파라미터 영역(parameter space)에서의 기저(bases)로 이루어지는데

$\begin{bmatrix} \phi_q \\ U_0^T \end{bmatrix}$ 의 역행렬이 존재하도록 선택

되어진다. 그러므로 U_0 의 열로 확장되어진 파라

미터영역과 ϕ_q^T 의 열로 확장되어진 부영역(subspace)은 R^{nr} 영역의 전체를 구성한다.⁷⁾

3.2 ODAE의 해법

식 (3.3)의 해를 구하기 위해 뉴턴 방법을 적용하면 다음 식을 얻을 수 있다.

$$F_q \Delta q + F_v \Delta v + F_a \Delta a + F_\lambda \Delta \lambda + F(x) = 0 \quad (3.4a)$$

$$\Phi_q \Delta v + \Phi_v \Delta v + \Phi_q \Delta q + \Phi(x) = 0 \quad (3.4b)$$

$$\Phi_q \Delta v + \Phi_q \Delta q + \Phi(x) = 0 \quad (3.4c)$$

$$\Phi_q \Delta q + \Phi(x) = 0 \quad (3.4d)$$

$$U_0^T (h \Delta a - \Delta v + h R_1) = 0 \quad (3.4e)$$

$$U_0^T (h \Delta v - \Delta q + h R_2) = 0 \quad (3.4f)$$

여기서 $h = \frac{\Delta t}{b_0}$ 이다.

먼저 식 (3.4)의 부분집합인 아래의 식을 고려해 보자.

$$F_q \Delta q + F_v \Delta v + F_a \Delta a + F_\lambda \Delta \lambda + F(x) = 0 \quad (3.5a)$$

$$\Phi_q \Delta q + \Phi(x) = 0 \quad (3.5b)$$

$$h \Delta a - \Delta v + h R_1(x) = 0 \quad (3.5c)$$

$$h \Delta v - \Delta q + h R_2(x) = 0 \quad (3.5d)$$

식 (3.5c)와 (3.5d)를 Δv 와 Δa 에 대하여 풀면 다음을 얻을 수 있다.

$$\Delta v = \frac{1}{h} \Delta q - R_2(x) \quad (3.6a)$$

$$\Delta a = \frac{1}{h^2} \Delta q - \frac{1}{h} R_2(x) - R_1(x) \quad (3.6b)$$

식 (3.6a)와 (3.6b)를 식 (3.5a)에 대입하고 h^2 을 곱하면 아래의 식을 얻을 수 있다.

$$K^* \Delta q + h^2 \Phi_q^T \Delta \lambda = R_3 \quad (3.7)$$

여기서 Φ_q^T 는 F_λ 이고, K^* 는 $h^2 F_q + h F_v + F_a$

로 나타낼 수 있으며, R_3 는 $-h^2 F + h (h F_v + F_a) R_2 + h^2 F_a R_1$ 로 나타낼 수 있다.

식 (3.5b)와 식 (3.7)로부터 다음 행렬 방정식을 형성할 수 있다.

$$\begin{bmatrix} K^* & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{bmatrix} \Delta q \\ h^2 \Delta \lambda \end{bmatrix} = \begin{bmatrix} R_3 \\ -\Phi \end{bmatrix} \quad (3.8)$$

윗식에서 Δq 와 $\Delta \lambda$ 를 구한다.

Δv 를 구하는 방법은 다음과 같다. 식 (3.4f)를 정리하면 다음의 식을 얻을 수 있다.

$$U_0^T (h \Delta v) = U_0^T (\Delta q - h R_2) \quad (3.9)$$

윗식에서 U_0^T 를 $N^T K^*$ (N 은 $\Phi_q \cdot N = 0$ 를 만족시키는 null-space 행렬)로 선정하면 식 (3.9)는 다음과 같다.

$$N^T K^* (\Delta v) = N^T K^* (cj \Delta q - R_2) \quad (3.10)$$

식 (3.10)은 다음식과 동가이다.

$$K^* (\Delta v) + \Phi_q^T \tau_1 = K^* (cj \Delta q - R_2) \quad (3.11)$$

그러므로 식 (3.11)과 식 (3.4c)로부터 다음의 행렬 방정식이 형성된다.

$$\begin{bmatrix} K^* & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{bmatrix} \Delta v \\ \tau_1 \end{bmatrix} = \begin{bmatrix} K^* (cj \Delta q - R_2) \\ -\Phi_q \Delta q - \Phi(x) \end{bmatrix} \quad (3.12)$$

여기서 τ_1 과 식 (3.11)에 나올 τ_2 는 식 (3.11)과 같이 등가식을 형성하므로써 미지수와 방정식수를 같게 만들기 위한 임의벡터이며 이렇게 함으로써 자코비안 계산시 형성된 식 (3.8)의 좌변계수 행렬을 그대로 이용할 수 있으므로 계산상의 효율을 높일 수 있다. 식 (3.12)로부터 Δv 를 구한다.

식 (3.4e)를 정리하면 다음과 같다.

$$U_0^T (\Delta a) = U_0^T (cj \Delta v - h R_1) \quad (3.13)$$

식 (3.13)와 (3.4b)를 Δv 와 마찬가지로 Δa 에 대하여 정리하면 다음과 같다.

$$\begin{bmatrix} K^* & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{bmatrix} \Delta a \\ \tau_2 \end{bmatrix} = \begin{bmatrix} K^* (cj \Delta q - R_1) \\ -\Phi_v - \Phi_q \Delta q - \Phi \end{bmatrix} \quad (3.14)$$

식 (3.14)로부터 Δa 를 구한다.

4. 실시간 차량 모델

4.1 전 차량 모델과 시스템 자유도

차량이나 기구학적 모델을 표현하는 방법으로서 위상학적 표현을 사용할 수 있다. 기계시스템은 크게 개회로(open loop)와 폐회로(closed loop)로 나누어진다. 개회로 시스템은 loop를 포함하지 않고 폐회로 시스템은 이를 포함하는 경우를 말한다. 본 연구에서는 조인트에 의해 연속적으로 연결되어진 바디(body)들의 체인을 따라 순차적인 계산을 수행하므로 폐회로를 잘라서 회로를 포함하지 않는 트리로 변환해야 한다. 이러한 과정을 신장(spanning)이라 하며 이 때 조인트(joint)와 절단 조인트(cut joint)가 구별되어진다. 본 연구에서 개발된 시뮬레이션 프로그램은 위의 조인트와 절단 조인트에 따라 상대 좌표와 구속조건이 결정되는데 전 차량 모델의 상대좌표수와 구속조건식은 도표 1과 같이 설정되었다. 도표 1에서 base body는 지면과 전혀 구속되어 있지 않으므로 6자유도를 갖는다. 이에 따라 시스템 자유도는 11자유도를 갖는다. 전체 차량은 그에 구속된 독립적인 현가장치로 구성되어 있다. 해석을 위한 전 차량모델은 총 12개의 강체로 구성되어 있고 전륜에 사용된 현가장치는 맥퍼슨 스트러트 2조를 가지는 것으로 구성되어 있고 후륜에 사용된 현가장치는 멀티링크 2조를 사용하였다. 차체에 대한 조향입력은 랙 앤드 피니언(rack and pinion)을 구성하여 사용하였다. 랙(Rack)은 맥퍼슨 스트러트의 tie rod와 구면 조인트로 구속된다.

4.2 맥퍼슨 스트러트 현가장치

맥퍼슨 스트러트 현가장치는 적은 부속을 가지고 현가장치의 하중을 넓은 면적에 전달하는 특성을 가지고 있다. 그 구조도가 그림 1에 보여진다. Lower control arm은 차체와 회전조인트로 연결, 구속되고 piston rod는 차체와 구면 조인트로 구속되어진다. strut와 knuckle은 병진조인트로 연결되어지고, tie rod 및 lower control arm은 구면조인트로 구속된다.

4.3 멀티링크 현가장치

멀티링크 현가장치는 링크지(Linkages)의 끝에 구면 조인트를 연결하여 굽힘모멘트를 받지 않게 된다. 그 구조도가 그림 2에 보여진다. Toe control arm (front, rear)와 camber control arm, trail link는 차체와 부형으로 구속되어진다. 기구학적 해석에서는 회전 조인트나 구면 조

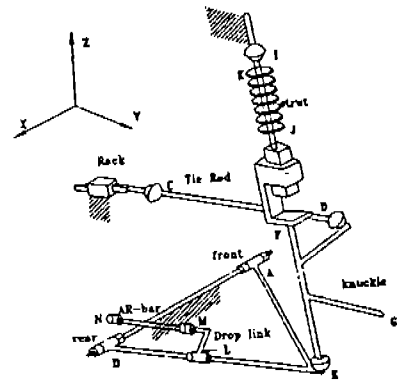


Fig.1 맥퍼슨 스트러트 현가장치의 구조

Table 1 조인트와 절단 조인트에 따른 상대좌표와 구속조건수

	상대 좌표		절단 조인트 구속 조건		자유도
	종류	base body(Chassis)	$6 \times 1 = 6$	구면 절단 조인트 2개	
병진 조인트 3개		$1 \times 3 = 3$			
회전 조인트 4개		$1 \times 4 = 4$	distance 절단 조인트 8개	$1 \times 8 = 8$	
구면 조인트 4개		$3 \times 4 = 12$			
합계	25개		14개		

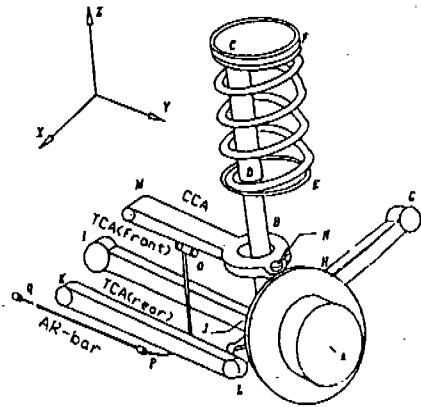


Fig.2 멀티링크 현가장치의 구조

인트로 연결한다. Knuckle에 연결된 toe control arm(front, rear)와 camber control arm, trail link는 구면조인트로 구속된다.

4.4 전 차량 모델의 위상학적 표현

본 연구에서 사용되어진 차량모델의 위상학적 표현은 그림 3과 같다. 그림에서 원은 바디를, 각 실선은 조인트를, 그리고 점선은 절단 조인트를 의미하며 T, R, sph는 각각 병진 조인트, 회전 조인트, 그리고 구면 조인트이다. 각 바디 기호와 distance 절단 조인트번호가 4.2, 4.3절에 소개한 현가장치와 이루는 관계는 표 2에 표시하였다.

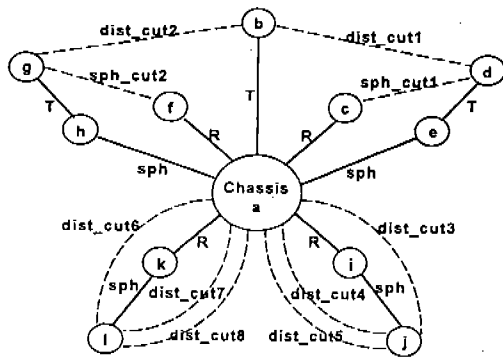


Fig.3 전차량 모델의 위상학적 표현

Table 2 현가 장치의 바디 및 distance 절단 조인트 기호

현가 장치 종류	명 칭	해당기호
맥퍼슨 스트러트 현가장치 (전 륜)	rack	b
	lower control arm	c, f
	knuckle	d, g
	strut	e, h
	tie rod	1, 2
멀티 링크 현가 장치 (후 륜)	camber control arm	i, k
	knuckle	j, l
	trail link	3, 6
	toe control arm(front)	4, 7
	toe control arm(rear)	5, 8

5. 수치해석 결과와 실시간 해석을 위한 Hardware 요구사항

5.1 재안된 시뮬레이션 프로그램의 검증

본 연구에서 개발된 시뮬레이션 프로그램의 정확도 검증을 위하여 범용 다물체 동역학 S/W인 ADAMS의 해석결과와 비교 검토하였다. 주행 시뮬레이션 조건을 두가지로 나누어 수행하였는데 차량을 80km/hr의 속도로 1초간 직진주행시킨 후 rack and pinion 조인트를 그림 4와 같이 조향하여 J-turn 시뮬레이션을 실행하고, 다른 하나는 같은 속도로 2초간 직진 주행시킨 후 1초동안 rack and pinion 조인트를 그림 5와 같이 조향하여 Lane Change 시뮬레이션을 실행하였다. stepsize를 1msec로 작게 설정하고 자코비안과 레지듀얼을 매 스텝 계산하여 구한 결과와 ADAMS결과를 그림 6과 7에 비교 도시하였다.(이하에서 자코비안, 레지듀얼, 역치환 계산은 다음과 같은 의미로 쓰기로 한다. 자코비안 계산은 식 (3.8)의 좌변 계수행렬을 구성하여 LU 분해법을 적용한 단계까지의 계산을, 레지듀얼 계산은 식 (3.8)의 H(x)를 계산한 것을, 그리고 역치환 계산은 식 (3.8), (3.12) 그리고 (3.14)의 우변 벡터를 구성하여 Δq , Δv , Δa , 그리고 $\Delta \lambda$ 를 구하는 계산을 의미한다) 그림 6과 7

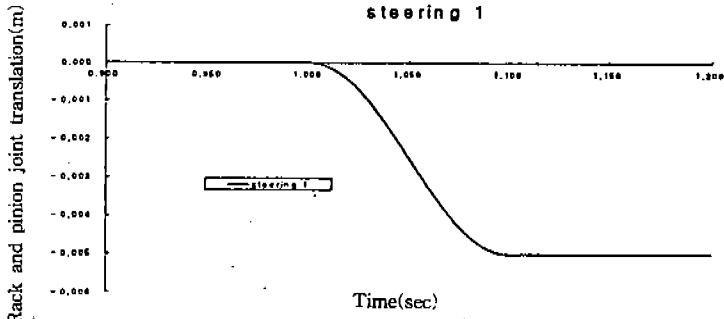


Fig.4 J-turn steering

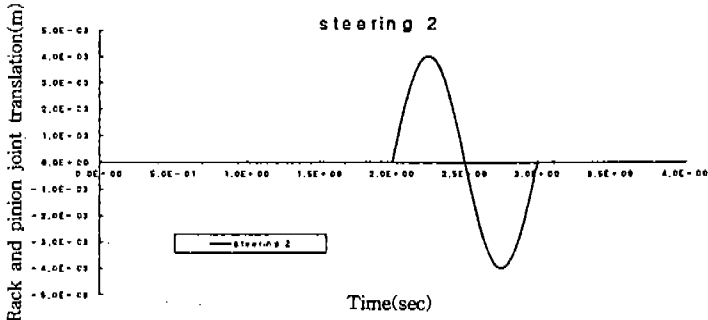


Fig.5 Lane change steering

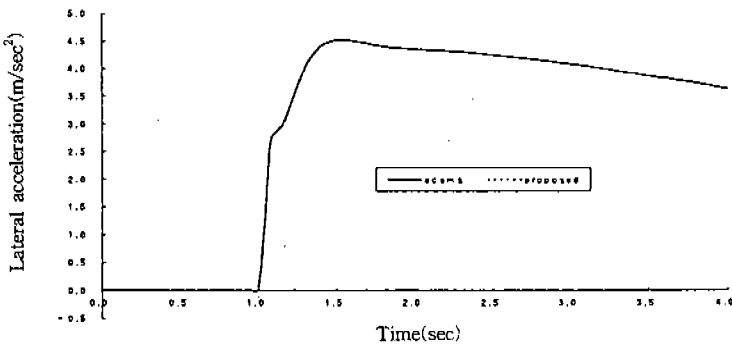
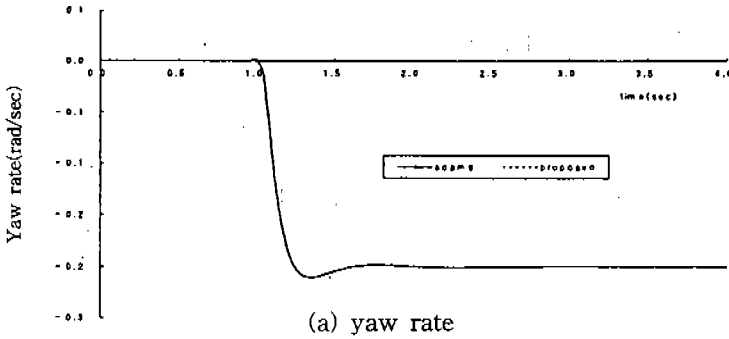
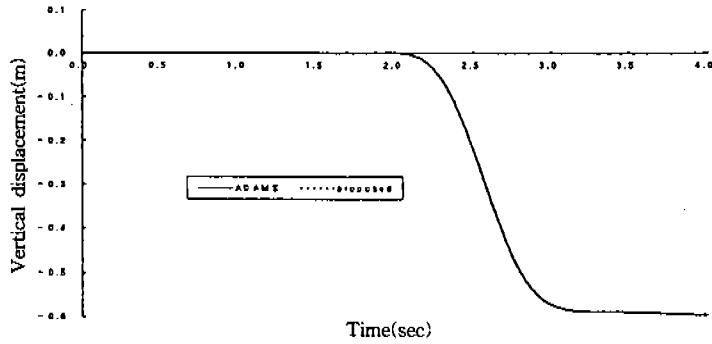
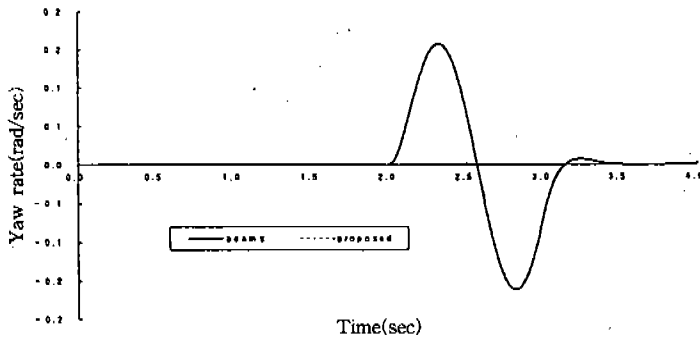


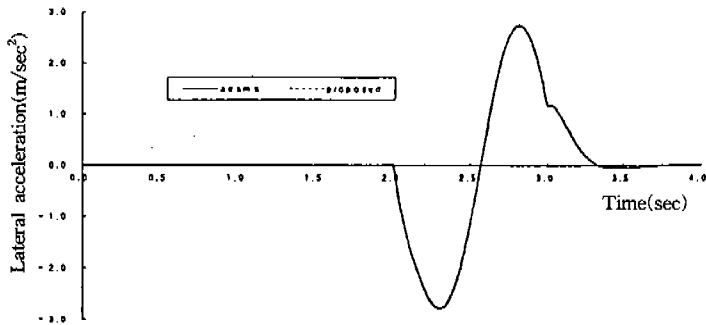
Fig.6 J-turn(stepsize=1msec)



(a) vertical displacement



(b) yaw rate



(c) lateral acceleration

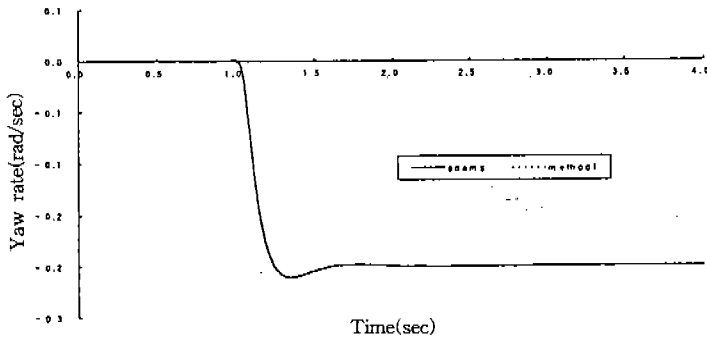
Fig.7 Lane change(stepsize=1msec)

에서 보듯이 동일한 결과가 나올 것을 확인하였고 이로써 본 연구에서 개발된 프로그램의 타당성을 검증하였다.

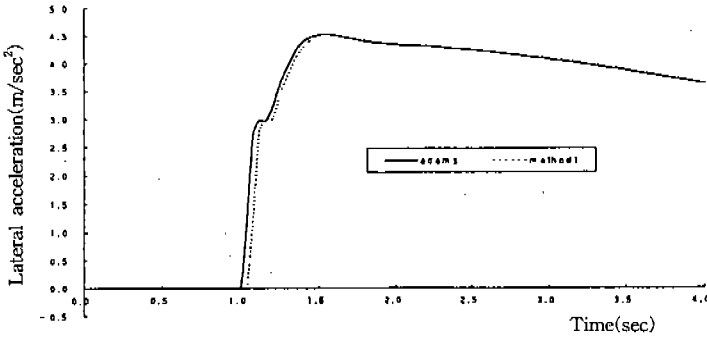
5.2 10msec의 수치 적분구간을 이용한 시뮬레이션 결과 고찰

참고문헌 3은 차량 실시간 시뮬레이션시 운전자의 입력을 외부로부터 10msec마다 입력받아

매 순간의 차량 상황을 계산하고 있다. 이는 실제 주행상황과 유사한 시뮬레이션을 수행하기 위해서는 외부로부터 입력을 넣어주는 간격이 최소한 10msec 정도로 판단되어졌기 때문이다. 본 연구에서 개발된 시뮬레이션 프로그램의 적분방법인 BDF는 10msec 이상의 적분구간도 가능하나 위에 설명한 이유에 따라 10msec를 기준 적분구간으로 정하여 연구를 수행하였다.



(a) yaw rate

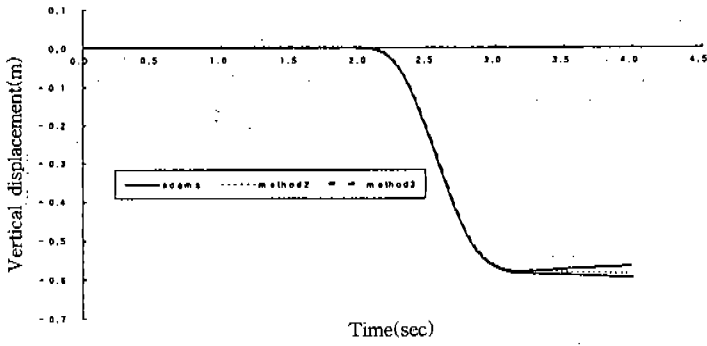


(b) lateral acceleration

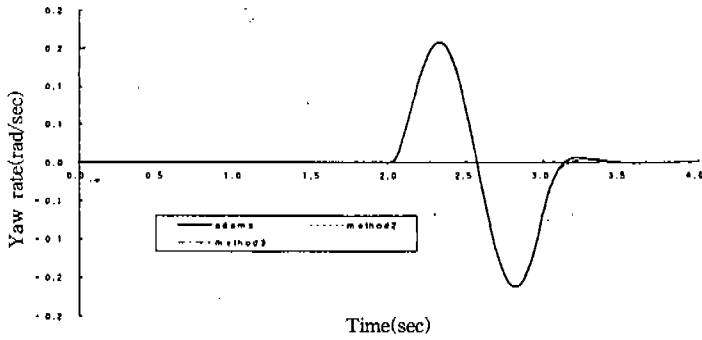
Fig.8 J-turn(stepsize=10msec)

BDF 적분은 자코비안을 계산해야 하는데 이는 매우 많은 계산시간을 요구한다. 그러므로 매 스텝 자코비안을 계산하지 않고 여러 스텝중 한 번씩만 계산해주므로써 적분시간을 줄여줄 수 있다. 또 자코비안 계산이 독립적인 프로세서에 의해 행해진다면 즉, parallel processing으로 시뮬레이션이 이뤄진다면 한 스텝적분에 소요되는 계산시간을 일정하게 하므로써 실시간과의 차이를 파악할 수 있다. 그러므로 이번 절에서는 일정한 적분구간(10msec)을 가지고 시뮬레이션시 자코비안 계산 횟수와 매 스텝당 뉴턴 램슨 반복 횟수를 변경시켜가며 해석 결과를 관찰하였다. 이에 대한 프로그램 순서도는 그림 10에 보여진다. 또 (1)절에서 언급한 두가지 경우의 주행모델을 ADAMS 결과와 비교하였다. 도표 3과 4는 각각 J-turn조향과 Lane Change조향에 대하여 가로축을 자코비안 계산 스텝에 따라서, 세로축을 뉴턴 램슨 반복횟수에 따라서 작성되었

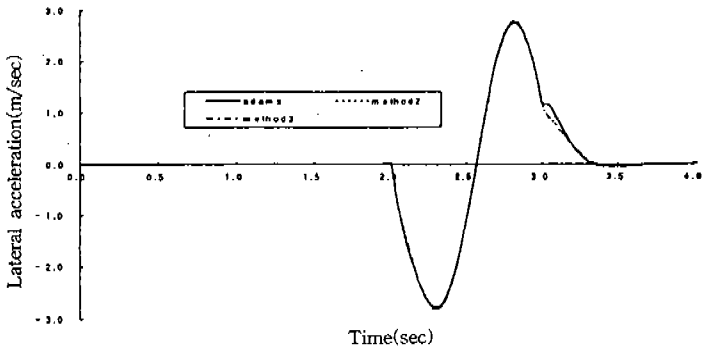
다. 도표 3과 4는 4장에서 설명된 특정 차량 모델과 실차 파라미터에 의한 결과이므로 모든 차량 모델의 파라미터에 대해서 같은 결과가 나온다고 단정지을순 없다. 그러나 차량의 시스템 진동수는 차량설계시 실험치에 의해 어느 일정 범위내에 들어오도록 결정되어져야 하며 이러한 시스템 진동수에 가장 큰 영향을 미치는 현가장치와 파라미터들은 차종에 따라 큰 차이를 갖지 않는다. 그러므로 본 연구를 통해 얻은 결과는 대부분의 차량 모델 파라미터에 대해서 큰 차이 없이 적용될 수 있으리라 사료된다. 그림 8은 J-turn 조향에 대하여 뉴턴 램슨 반복횟수를 2회로 설정하고 200스텝마다 자코비안을 계산하여 구한 결과(method1)를 ADAMS 결과와 비교도시하였다. 도표 3에서 보듯이 뉴턴 램슨 반복 횟수를 1회로 설정하고 매 스텝 자코비안을 계산하는 경우에도 총 400스텝중 대부분의 스텝에서 제대로 된 해를 구하지 못한채 적분을 해나가서



(a) vertical displacement



(b) yaw rate



(c) lateral acceleration

Fig.9 Lane change(stepsize=10msec)

Table 3 J-turn(stepsize=10msec)

	only 1st step	every 200step	every 100step	every 2step	every step
N-R iteration 1	발산	발산	발산	발산	발산
N-R iteration 2	발산	성공	성공	성공	성공

Table 4 Lane change(stepsize=10msec)

	only 1st step	every 200step	every 100step	every 2step	every step
N-R iteration 1	발산	발산	발산	발산	발산
N-R iteration 2	성공	성공	성공	성공	성공

결국 국부오차(local error)가 누적되어 발산하였다. 반면에 뉴턴 램슨 반복 횟수를 2번으로 설정하고 200스텝마다 한번씩 자코비안을 계산해 주는 경우는 자코비안을 계산해주는 횟수가 매우 적지만 뉴턴 램슨 반복을 통해 정확한 해를 구하며 적분이 수행되었으므로 발산하지 않고 실제해와 근사한 해를 구할 수 있음을 그림 8을 통해 확인할 수 있다. 그림 9는 Lane Change 조향에 대하여 뉴턴 램슨 반복 횟수를 2회로 설정하고 200스텝마다 자코비안을 계산하여 구한 결과(method2)와 마찬가지로 뉴턴 램슨 반복횟수를 2회로 설정하고 첫 스텝에서 한 번만 자코비안을 계산하여 구한 결과(method3)를 ADAMS 결과와 비교 도시하였다. 두 경우 모두 연직방향 변위와 lateral acceleration에서 다소 차이를 보였지만 대체로 비슷한 경향을 보임을 확인할 수 있다.

5.3 실시간 해석을 위한 hardware 요구사항

실제해와 근사한 해를 구하기 위해 가장 많은 적분 시간을 요하는 뉴턴 램슨 반복횟수를 2번으로 하고 200스텝마다 한번씩 자코비안을 계산하는 경우에 대하여 한 스텝당(10msec) 계산소요시간을 평균내어 실시간과의 차이를 파악하고 이를 통해 실시간에 필요한 Hardware 요구사항을 예측해 보고자 한다. 시뮬레이션에 사용된 컴퓨터기종은 pentium 166MHz이며, 이 기종에서 자코비안(Jacobian), 레지듀얼(Residual), 그리고 역치환(Back-Substitution) 계산시간은 각각 68.99msec, 9.47msec, 21.16msec이다. 4초 시뮬레이션시 총 적분 스텝은 400회이며 자코비안 계산횟수는 2회, 레지듀얼과 역치환 계산횟수는 각각 800회이다. 그러므로 이를 한 스텝 당의 적분 계산시간으로 환산해 보면 61.60msec

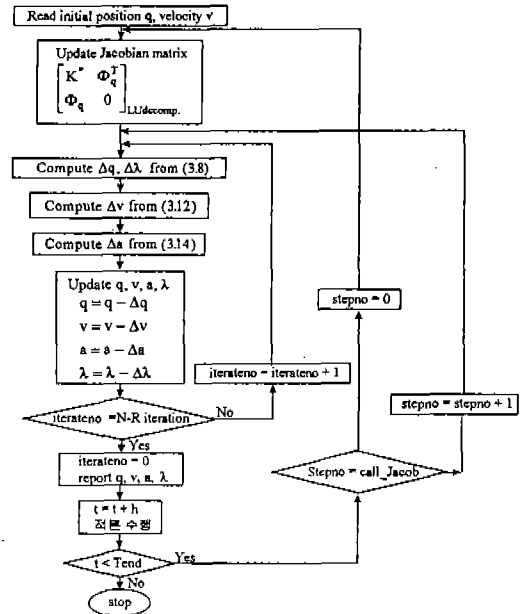


Fig.10 프로그램 순서도

가 나온다. 즉, 차량 전용 프로그램에 대해서 pentium 166MHz 컴퓨터보다 6배 가량 처리속도가 빠른 기종이 실시간 차량 시뮬레이션을 위해 요구되어짐을 알 수 있다.

6. 결 론

본 연구를 통하여 차량 시뮬레이션 프로그램을 개발하고 실제 차량에 사용되는 현가장치를 장착한 차량을 모델링하여 그 결과를 상용프로그램인 ADAMS를 이용하여 검증하였다. 위 시뮬레이션 프로그램은 절대 좌표 대신에 상대 좌표를 이용하여 운동방정식 및 기구학적 구속조건식을 표현하므로써 시스템 행렬의 크기를 줄

여 계산시간을 절약하였다. 또 일반적으로 실시간 시뮬레이션시 적분방법으로 사용되는 명시적 적분 방법 대신에 암시적 적분 방법인 BDF를 사용하여 시뮬레이션 하므로써 실시간 해석에서의 그 이용가능성을 제시하였다. 이에 대한 구체적인 사항으로 J-turn 주행과 Lane Change 주행에 대해서 10msec의 적분구간으로 적분하여 실제해와 근사한 해를 구하기 위하여 필요한 최소의 자코비안, 레지듀얼, 역치환 계산횟수가 찾아졌다. 이의 실행을 실시간에 얻는 데는 pentium pc 166MHz보다 6배 가량 빠른 프로세서를 가진 컴퓨터가 필요하리라는 예측을 할 수 있었다.

후 기

이 논문은 교육부의 후원을 받아 작성되었습니다.

참 고 문 헌

1. J. Wittenburg, Dynamics of Wystems of Rigid Bodies, B. G. Teubner, Stuttgart, 1977.
2. Thomas D. Gillespie, Fundamentals of Vehicle Dynamics, SAE, pp. 7~19, 275~305.
3. Drosdol. J, and Panik. F., The Daimler-Benz Driving Simulator : A tool for Vehicle Development, SAE paper 850334, 1985.
4. 이운성, J. Samson, W. S. Lee, "실시간

시뮬레이션을 위한 병렬 적분", 자동차공학회 논문집 제2권 제1호, pp. 106~115, 1994.

5. 최규재, 이관호, 유영면, "실시간 차량동력학 시뮬레이션 S/W 개발", 자동차공학회 논문집 제3권 제5호, pp. 30~37, 1995.
6. C. W. Gear, "The simultaneous numerical solution of differential-algebraic equations", IEEE Trans. Circuit Theory, Vol. 18, pp. 89~95, 1971.
7. K. E. Brenan, S. L. Campbell and L. R. Petzold, "Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations", SIAM Press, 1995.
8. F. A. Potra, "Implementation of linear multistep methods for solving constrained equations of motion", SIAM J. Numer. Anal., Vol. 30, pp. 74~789, 1993.
9. Ming-Gong Lee and Edward J. Haug, "Stability and Convergence for Difference Approximations of Differential-Algebraic Equations of Mechanical system Dynamics", Technical Report R-157, August, 1992.
10. D.S.BAE. and Edward J. Haug, "A Recursive Formulation for Constrained Mechanical System Dynamics : Part 2. Closed Loop Systems", Mech. struct. and Machines, Vol. 15, pp. 481~506.
11. Edward J. Haug, Computer-Aided Kinematics and Dynamics of Mechanical Systems Volume 1 : Basic Methods pp. 440~454.