韓國精密工學會 '97年度 秋季學術大會論文集 pp.497~500

실시간 차량 시뮬레이터 개발을 위한 암시적 적분기법을 이용한 병렬처리 알고리즘에 관한 연구

Study on the parallel processing algorithms with implicit integration method for real-time vehicle simulator development

박 민 영(한양대 대학원), 이 정 근,(한양대 대학원) 송 창 섭,(한양대 공대) 배 대 성(한양대 공대)
M.Y.Park(Graduate School Hanyang Univ.), J.K.Lee(Graduate School Hanyang Univ.)
C.S.Song(Hanyang Univ.), D.S.Bae(Hanyang Univ.)

ABSTRACT

In this paper, a program for real time simulation of a vehicle is developed. The program uses relative coordinates and BDF(Backward Difference Formula) numerical integration method. Numerical tests 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. Algorithms of parallel processing is developed with DSP(digital signal processor).

Key Words : Parallel Processing Algorithm(병렬처리 알고리즘), Real-time-simulation(실시간 시뮬레이션) Implicit numerical integration(암시적 수치적분), BDF(Backward Difference Formula)

1. 서 론

지금까지 개발된 실시간 차량 시뮬레이터에는 차량 동역학 시뮬레이션시 간략화된 차량모델을 사용하여 차량 시뮬레이션 정확도가 떨어지거나, 대단히 계산량이 많은 차량 모델을 사용하므로써 초대형 컴퓨터를 필요로 하는 문제점이 있다. 본 연구에서는 일반좌표로써 상대좌표를 채택하고, 적분 방법으로 암시적 수치 적분방법 중의 하나인 BDF를 이용했다. 계산시간의 대부분을 차지하는 자코비안(Jacobian),레지듀얼(Residual),역치환(Back-Substitution)의 계산횟수에 따른 한 스텝 당 평균시간을 계산하여 실시간에 필요한 하드웨어 기종을 제시하며 그 가능성을 보이고자 한다.또 DSP를 사용하여 실시간 모의실험이 가능함을 입증하려고 한다.

2. 차량 시뮬레이터 개발

(1) 제안된 시뮬레이션 프로그램의 검증
본 연구에서 개발된 시뮬레이션 프로그램의 정확도 검증을 위하여 범용 다물체 동역학 S/W인 ADAMS의 해석결과와 비교 검토하였다.
J-turn 모의실험과Lane Change 모의실험을 실행하였다. 이로써 본 연구에서 개발된 프로그램의 타당성을 검증하였다.

(2) 10msec의 수치 적분구간을 이용한 모의실험
BDF 적분은 자코비안을 계산해야하는데 이는 매우 많은 계산시간을 요구한다. 그러므로 매 스텝 자코비안을 계산하지 않고 여러 스텝중 한번씩만 계산해 주므로써 적분시간을 줄여줄 수 있다. 일정한 적분구간(10msec)을 가지고 시뮬레이션시 자코비안 계산횟수와 매 스텝당 뉴튼 랩슨 반복횟수를 변경시켜가며 해석 결과를 관찰하였다. 이를 위해 (1)절에서 언급한 두가지 경우의 주행모델을 ADAMS결과와 비교하였다. 도표 1와 2는 각각 J-turn조향과 Lane Change조향에 대하여 가로축을 자코비안 계산 스텝에 따라서, 세로축을 뉴튼 랩슨 반복횟수에 따라서 작성되었다. 그림 1과 그림 2는 J-turn 조향에 대하

여 뉴튼 랩슨 반복횟수를 2회로 설정하고 200스텝마다 자코비안을 계산하여 구한 결과(method1)를 ADAMS 결과와 비교 도시하였다. 도표 1에서 보듯이 뉴튼 랩슨 반복 횟수를 1회로 설정하고 매 스텝 자코비안을 계산하는 경우에도 총 400 스텝중 대부분의 스텝에서 제대로 된 해를 구하지 못한 채 적분을 해나가서 결국 국부오차(local error)가 누적되어 발산하였다. 반면에 뉴튼 랩슨 반복횟수를 2번으로 설정하고 200스텝마다 한번씩 자코비안을 계산해 주는 경우는 자코비안을 계산해주는 횟수가 매우 적지만 뉴튼 랩슨 반복을 통해 정확한 해를 구하며 적분이 수행되었으므로 발산하지 않고 실제해와 근사한 해를 구할 수 있음을 그림 1과 그림 2를 통해 확인할수 있다. 그림 3, 그림 4 그리고 그림 5는 Lane Change 조향에 대하여 뉴튼 랩슨 반복횟수를 2회로 설정하고 200스텝마다 자코비안을 계산하여 구한 결과와 뉴튼 랩슨 반복횟수를 2회로 설정하고 첫 스텝에서 한번만 자코비안을 계산하여 구한 결과를 ADAMS 결과와 비교 도시하였다. 두 경우 모두 연직방향 변위와 lateral accelaration에서 다소 차이를 보였지만 대체로 비슷한 경향을 보임을 확인할수 있다.

도표 1. J-turn (stepsize = 10msec)

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

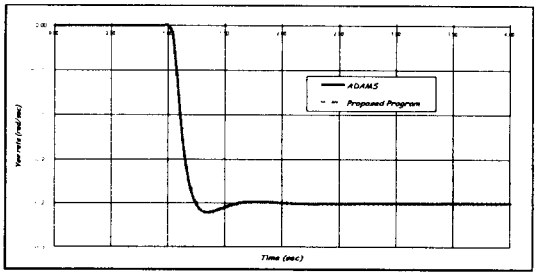

그림.1 Yaw rate in case of J-turn

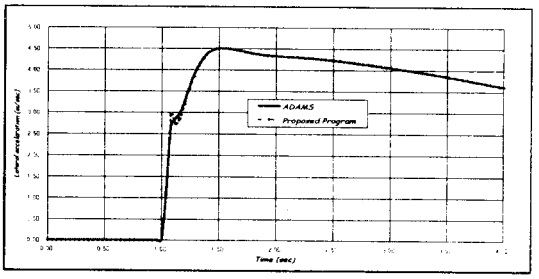

그림 2. Lateral acceleration in case of J-turn

도표 2. Lane Change (stepsize = 10msec)

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

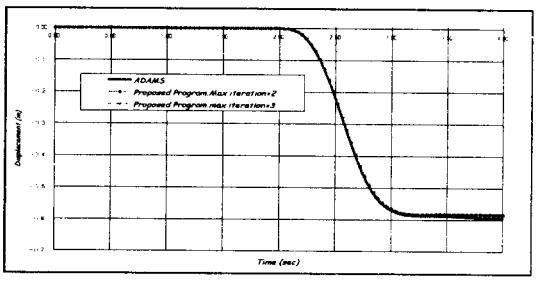

그림 3. Vertical displacement in case of Lane Change

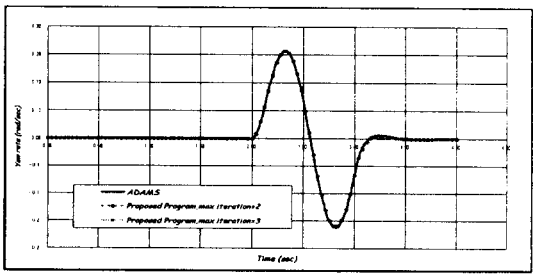

그림 4. Yaw rate in case of Lane Change

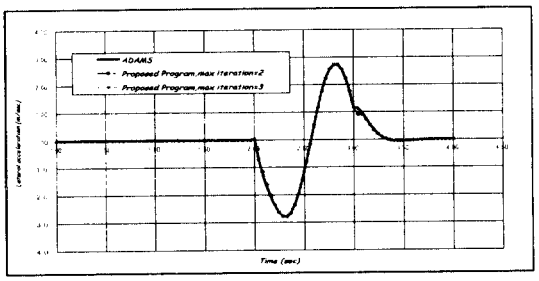

그림 5. Lateral acceleration in case of lane change

(3) 실시간 해석을 위한 hardware 요구사항

시뮬레이션에 사용된 컴퓨터기종은 pentium 166MHz이며, 이 기종에서 자코비안(Jacobian),레지듀얼(Residual), 그리고 역치환(Back-Substitution) 계산시간은 각각 68.99msec, 9.47msec, 21.16msec 이다. 4초 시뮬레이션시 총 적분 스텝은 400회 이며 자코비안 계산횟수는 2회, 레지듀얼과 역치환 계산횟수는 각각 800회이다. 그러므로 이를 한 스텝 당의 적분 계산시간으로 환산해 보면 61.60msec가 나온다. 즉, 차량 전용 프로그램에 대해서 pentium 166MHz 컴퓨터보다 6배 가량 처리속도가 빠른 기종이 실시간 차량 시뮬레이션을 위해 요구되어짐을 알수 있다.

3. 병렬처리 알고리즘

(1) 수행과정

직렬처리 알고리즘에서는 모든 프로그램을 Pentium CPU에서 모의실험을 수행하였고 DSP는 사용하지 않았다.직렬처리 알고리즘을 사용할 때의 문제점은 모의실험시간이 길어지고, 앞 subroutine에서 넘어오는 데이터를 필요로 할 때 대기시간이 길어지면 sampling time이 제한되고 제한된 sampling time이상으로 대기할 경우에는 전체 프로그램의 결과가 발산할 가능성이 높아진다.

직렬 처리(Serial processing)의 문제점에 의해 DSP 프로세서 2개를 사용한 병렬 처리(Parallel processing)를 사용했으며 그 구성도는 그림 II.3.6와 같다.중요한 점은 직렬처리 알고리즘에서 사용했던 sampling time을 사용하는 것이 아니고 참고문헌에서 사용한 일반적인 실시간 차량 시뮬레이터에서 사용하는 sampling time을 사용한다. 이러한 sampling time을 적용하였을 때 응답이 발산하지 않거나 정확한 결과가 나오는 지에 대해서 조사하고 그렇지 않다면 sampling time을 변경하여 모의실험을 수행한다.

1) 알고리즘 1

첫 번째 시도한 알고리즘은 가장 시간이 많이 드는 모듈 2가지를 선정하여 그 하나는 site A에, 다른 하나는 site B에 download하여 연산하는 알고리즘이다. 암시적 적분기법의 절의 결과를 볼 때 가장 많은 시간을 요하는 subroutine은 자코비안을 연산하는 subroutine과 LU 분해와 back-substitution을 연산하는 subroutine이다. 그리고 차량동역학을 구성하는 자코비안과 레지듀얼을 서로 분리하였을 경우에는 더 많은 CPU time의 손실을 초래하므로 하나의 프로세서에서 연산을 하고 다른 하나의 프로세서에서는 LU decomposition, back-substitution 그리고 그 밖의 다른 subroutine을 연산하도록 한다. 그림 4는 그러한 알고리즘의 flow chart이다.

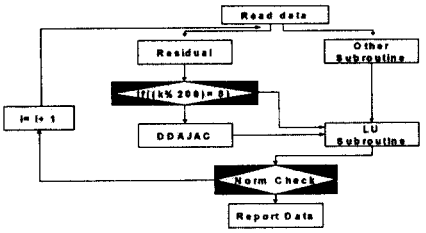

그림 6. Flow chart of algorithm 1

전체 시뮬레이션 시간을 10초라고 하였을 때 모의실험 시간은 10.87초가 걸리고 그 응답도 직렬처리 알고리즘을 적용한 응답과 동일하였다. 따라서 이 첫 번째 알고리즘은 실시간 모의실험에 적합함을 증명할 수 있다.

2) 알고리즘 2

두 번째 시도한 알고리즘은 프로그램 순서에 따라 앞 모듈을 site A에, 그 다음 모듈을 site B에 download하여 계산하는 알고리즘이다

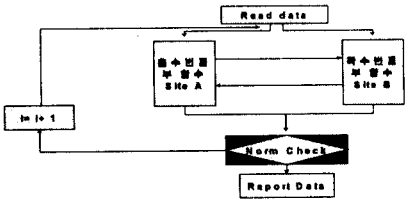

그림 7. Flow chart of algorithm 2

전체의 프로그램 흐름도에서 연산의 우선순위를 직렬처리 알고리즘에서 생각할 때 제일 처음 연산되는 subroutine을 site A에 load하여 연산하고 그 다음은 순차적으로 연산되는 subroutine을 site B, site A순으로 load시킨다. 알고리즘 2를 적용하는데 있어서의

먼저 고려해야 할 점은 subroutine을 어디까지 적용하는 것이다. 개발된 프로그램은 수많은 subroutine으로 구성이 되어있어 트리구조의 호출구조를 띄고 있지 않고 서로가 얽혀있는 그물구조를 띄고 있다. 그래서 이 범위를 정의하는 것이 매우 중요하다. 본 연구에서는 이 범위를 시행착오법으로 결정하였다. 즉 다시 말해서 각각의 subroutine을 모두 하나의 객체로 보고 모의실험을 수행하여보고 올바른 결과가 나오지 않을 경우 조금씩 묶어나가는 방향을 선정하였다. 그리하여 대개의 경우 개발된 프로그램의 한 library인 reclib.c에 있는 대부분의 subroutine들은 그 함수가 호출되는 부분에 종속시키고 그리고 짧은 subroutine들의 경우에도 대개 상위 subroutine에 종속시켰다.위의 결과에서 보듯이 실시간 모의실험이 불가능함을 보여주고 있다.마찬가지로 전체 모의실험시간은 10초를 실행하였고 sampling time을 10ms으로 선정하였을 때 모의실험시간은 297.68초이었고, 10ms의 경우에는 더 큰 sampling time에 대하여 실시간 모의실험을 수행하지 못하였으므로 실행하지 않았다.따라서 알고리즘 2는 적합하지 않음이 증명되었다.

3) 알고리즘 3

마지막으로 시도한 알고리즘은 차량 동역학 모듈 전체를 site A에, 드라이브 트레인 모듈 전체를 site B에 download하여 계산하는 알고리즘이다.

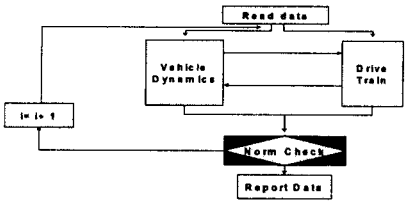

그림 8. Flow chart of algorithm 3

그림 8.에서 Module B는 차량 동역학(Vehicle dynamics)을 담당하게 된다. Module B는 연산된 결과를 Module A로 보내 주어야 하고 이 데이타는 Module A가 드라이브 트레인에 필요한 입력 값들이 되므로 Module A의 Interrupt 주기보다 빨리 연산을 마쳐 Module A에 데이타를 전송해야 한다. Module A는 Module B에서 연산된 결과를 사용하고 필요한 데이타를 PC쪽으로 보내 주며 Interrupt frequency에 의해 DPRAM 혹은 LIA를 사용하여 데이타를 Module B에 보내 주기도 하며 PC에 저장하기도 한다. 따라서 Module A가 본 연구에서 병렬처리를 하는데 있어서 Master의 역할을 하게 된다.

3. 결 론

본 연구에서 앞서 언급한 많은 결과들을 총괄적으로 정리하여 볼 때 다음의 결론을 얻을 수 있었다.

(1) 암시적 적분기법을 이용한 전차량 모델을 개발하였다. 이 모델의 특징은 수치적분구간의 크기에 거의 영향없이 안정함을 알 수 있었다.

(2) 상대좌표와 recursive 다물체 동역학을 이용한 전차량 모델을 개발하였다. 이 모델의 특징은 명시적 적분기법을 이용한 전차량 모델에 비하여 적분속도에 차이가 없음을 알 수 있었다.

(3) 상기의 전차량 모델을 이용하여 차량의 거동해석을 특히 현가장치의 거동해석을 검증할 수 있었다.

(4) DSP를 사용하여 병렬처리 알고리즘을 적용한 경우 2개의 DSP를 사용하고도 실시간 차량 시뮬레이터의 개발가능성을 입증하고 이에 대한 알고리즘을 제안하였다.

(5) 제안된 알고리즘의 실용성을 입증하였다.

4. 참고문헌

1. Thomas D. Gillespie, "Fundamentals of Vehicle Dynamics" , SAE, pp7-19, 275-305
2. 이 운 성, J. Samson, "W. S. Lee, 실시간 시뮬레이션을 위한 병렬 적분", 자동차공학회논문집 제2권 제 1호, pp. 106 - 115, 1994
3. 최 규 재, 이 관 호, 유 영 면, "실시간 차량동력학 시뮬레이션 S/W 개발",자동차공학회논문집 제3권 제 5호, pp. 30 - 37, 1995
4. 김 범 수, "실시간 차량 시뮬레이터를 위한 병렬처리 알고리즘에 관한 연구", 한양대학교 석사논문, 1995