

지능형 차량 HILS를 위한 실시간 차량 동역학 모델 개발

이 창 호¹⁾ · 김 성 수²⁾ · 정 원 희³⁾ · 이 선 호³⁾

넥센타이어 중앙연구소¹⁾ · 충남대학교 메카트로닉스공학과²⁾ · 충남대학교 메카트로닉스공학과 대학원³⁾

Development of Real Time Vehicle Dynamics Models for Intelligent Vehicle HILS

Changho Lee¹⁾ · Sung-Soo Kim²⁾ · Wanhee Jeong³⁾ · Sunho Lee³⁾

¹⁾R & D Center, Nexen Tire Co., 30 Yusan-dong, Yangsan-si, Gyeongnam 626-230, Korea

²⁾Department of Mechatronics Engineering, Chungnam National University, Daejeon 305-764, Korea

³⁾Graduate School of Mechatronics Engineering, Chungnam National University, Daejeon 305-764, Korea

(Received 7 February 2006 / Accepted 10 March 2006)

Abstract : Real time vehicle dynamics models have been developed with the subsystem synthesis method for intelligent vehicle HILS system. Three different models for solving subsystem equations are compared in order to find out the best suitable model for HILS applications. The first model is based on the generalized coordinate partitioning technique, and the second one is on the approximate function approach, and the last one is on the constraint stabilization method. To investigate the theoretical efficiency of three proposed methods, arithmetic operators used in the formulations of three models are counted. Bump run simulations with half-sine bump have also carried out with three different models to measure the actual CPU time to validate theoretical investigation.

Key words : Real time multibody vehicle dynamics (실시간 다물체 차량 동역학) Subsystem synthesis method(부분 시스템 합성법) HILS (하드웨어 인 더 루프 시뮬레이션)

1. 서론

차량의 안정성 및 편의성 향상을 위한 ABS (anti-lock brake system), TCS(traction control system), ESP(electronic stability program), ECS(electronically controlled suspension), CDC(continuous damper control), AFS(active front steering), UCC(Unified chassis control), EPS(electric power steering), 4WS(four wheel steering), X-By-Wire(brake-by-wire, steer-by-wire, chassis-by-wire) 시스템은 지능형 차량의 핵심 요소로 등장하고 있다. 이러한 사시 제어장치의 개발에는 설계 및 시제품 제작으로부터 차량 시험장에서

수행하는 다양한 조정안정성 평가를 위한 시험 검증까지 많은 비용과 시간, 그리고 위험성을 수반하게 되는데 이를 줄이기 위해서는 수학적 가상의 차량 모델(소프트웨어 모델)과 개발하고자 하는 제어 부분 시스템(하드웨어)이 결합하여 시뮬레이션을 수행하는 HILS(Hardware-in-the-loop Simulation) 기법이 효과적이다.

지능형 차량의 사시 제어 시스템의 효과적인 개발을 위한 HILS 시스템은 실시간 차량 동역학 모델과 사시 제어 시스템의 효과적인 연결을 위한 인터페이스, ECU 및 사시 제어 관련 하드웨어 시스템으로 구성된다. 구성요소 중 실시간 차량 동역학 모델은 HILS 시스템의 핵심적인 요소로서 제어 시스템

*Corresponding author, E-mail: sookiim@cnu.ac.kr

개발의 신뢰성을 위해서는 차량 시험장에서의 다양한 조정 안정성 시험에서의 실제 차량의 거동을 1ms의 매우 짧은 적분 간격에서도 실시간으로 재현할 수 있어야 한다.

현재 상용 HILS 도구로 개발된 차량 동역학 프로그램인 CarSim,¹⁾ Vedyna²⁾ 등은 차량의 현가장치, 조향장치 등을 구성하는 각종 링크 등을 바퀴의 수직 운동과 단순 스프링과 댐퍼로 간략화한 모델을 사용하기 때문에 현가장치의 정적 기구학 특성 실험 데이터를 이용한다. 따라서 실제의 차량의 거동을 정확하게 재현하지 못하나, 운동방정식이 매우 간단하여 HILS에서 요구하는 1ms의 매우 짧은 적분 간격내의 실시간 해석이 가능하다.

그러나 앞서 기술한 바와 같이 제어 시스템 개발의 신뢰성을 확보하기 위해서는 보다 정확한 거동을 재현하는 다물체 동역학에 기반을 둔 차량 모델이 필요하다. 다물체 동역학 모델은 차량을 구성하는 현가장치, 조향장치 등을 실제 각 구성품 그대로 모델링한 가상 차량모델을 사용하기 때문에 보다 정확한 차량의 거동 해석이 가능하다. 하지만 개개의 현가 링크의 운동을 반영하고 링크간의 구속조건들을 반영하므로, 미분-대수 방정식 형태의 복잡하고 변수의 개수가 많은 운동방정식을 다루어야 한다. 변수의 개수를 줄이기 위해서, 복합 조인트를 이용한 무질량 링크 개념의 다물체 동역학 모델이 개발³⁾되었지만 현가장치의 종류에 따라서는 현가의 스프링과 댐퍼의 작용점에 해당하는 물체가 없기 때문에 그 효과를 정확하게 반영하는 것이 힘들다. 이와는 달리, 최근 개발된 부분 시스템 합성방법은 무질량 링크를 사용하지 않고도, 차시 프레임과 각각의 현가 부분 시스템의 분리된 운동 방정식을 생성하기 때문에 효율성을 향상 시켰다.^{4,5)}

본 논문에서는 HILS 시스템에서 요구하는 1ms의 짧은 적분 간격의 해석을 수행하고 동시에 보다 정확하게 차량의 운동을 재현할 수 있는 실시간 차량 동역학 모델 개발을 위해 다물체 동역학을 해석 방법을 사용하고, 부분 시스템 합성방법에 기반을 둔 3가지 모델(일반좌표 분할법, 구속조건식의 안정화 방법, 근사함수방법)을 비교하여 HILS에 적합한 모델을 제시하고자 한다.

2. 부분시스템 합성법에 의한 다물체 차량 동역학 운동 방정식

Fig. 1은 맥퍼슨 스트럿과 멀티링크 현가장치의 차량 모델의 부분 시스템 해석을 위한 위상분석도이다. 부분 시스템 합성법을 적용하여 차량을 해석하는 경우 먼저 4개의 독립적인 부분 시스템이 차체에 미치는 효과를 효과 관성 행렬 \tilde{M}_i ($i=1\sim 4$)과 효과 힘 벡터 \tilde{P}_i ($i=1\sim 4$)를 통하여 구하고, 그 효과를 차체의 운동방정식에 합성하여 6자유도의 차체만의 운동방정식을 식 (1)과 같이 구한다.⁵⁾

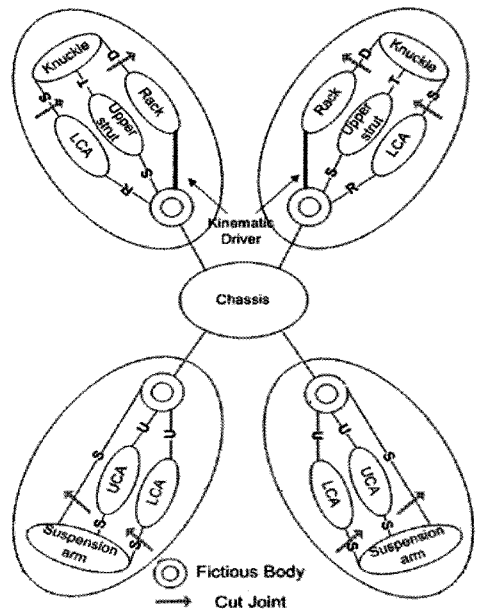


Fig. 1 Topology of 4 subsystem

$$(\hat{M}_0 + \sum_{i=1}^4 \tilde{M}_i) \dot{Y}_0 = (\hat{Q}_0 + \sum_{i=1}^4 \tilde{P}_i) \tag{1}$$

식 (1)을 이용하여 차체 가속도의 상태 벡터인 $\dot{Y}_0(t)$ 를 구하면 각각의 현가 부분 시스템에 대하여 식 (2)와 같은 독립적인 운동방정식을 얻을 수 있다.

$$\begin{bmatrix} \bar{M}_{qq} & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{P}_q \\ \bar{\gamma} \end{bmatrix} - \begin{bmatrix} \bar{M}_{yq}^T \\ 0 \end{bmatrix} \dot{Y}_0(t) \tag{2}$$

여기서 \bar{M}_w 는 부분시스템의 관성행렬, \bar{P}_v 는 부분 시스템의 힘 벡터이고, \bar{M}_w 는 부분 시스템과 샤시 프레임과 연성된 관성행렬이다. 또한 Φ_q 은 부분시스템의 구속조건식에 대한 자코비안 행렬이며, \bar{v} 는 가속도 구속조건식의 우변 항이 된다. 그리고 λ 는 라그랑지 승수를 나타낸다.⁵⁾

Fig. 2는 Fig. 1의 차량 모델에 대한 부분 시스템 합성법의 계산 순서의 흐름도를 나타내고 있다. 먼저 샤시 프레임의 위치, 자세, 속도, 각속도를 각각의 부분 시스템의 루틴으로 보내면, 각각의 부분 시스템은 이 정보로부터, 현가 부분 시스템의 위치, 속도 해석을 순환 다물체 공식⁶⁾에 의해서 수행한다. 그리고 각각의 현가 부분 시스템을 구성하고 있는 물체의 관성과 일반화된 힘 벡터를 구하고, 효과 관성행렬과 효과 힘 벡터를 구하게 된다. 구한 효과 관성행렬과 효과 힘 벡터를 식(1)과 같이 모두 합성하여 독립적인 6자유도의 샤시 프레임 운동방정식을 형성하고, 샤시 프레임의 가속도 상태 벡터를 구한다. 샤시 프레임의 가속도 상태 벡터 정보와 함께, 다시 각각의 현가 부분 시스템별로 식 (2)의 운동방정식을 풀어, 현가 부분 시스템 운동을 기술하는 변수의 가속도를 구할 수 있다.

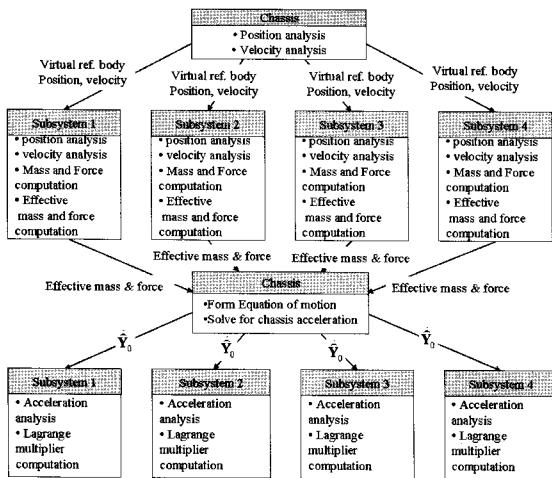


Fig. 2 Program structure of subsystem synthesis method

2.1 일반좌표 분할법(Generalized Coordinate Partitioning Method)

식 (2)의 현가부분 시스템의 운동방정식은 미분대수 방정식으로 여러 가지 방법으로 그 해를 구할 수 있다. 그 중 한 가지 방법인 일반 좌표 분할법은 현가 부분 시스템에 사용된 좌표 \bar{q} 를 독립좌표 (independent coordinate) \bar{v} 와 종속좌표(dependent coordinate) \bar{u} 로 분할하고 운동방정식을 독립 좌표의 함수로 변환하여 해를 구하는 방법이다.⁷⁾

일반 좌표 분할법으로 부분 시스템의 위치, 자세 해석을 위해서는 식 (3)의 현가 링크의 페루프 구속조건식을 풀어야한다.

$$\Phi(\bar{q}) = 0 \tag{3}$$

식 (3)은 비선형 방정식이므로 다음의 뉴턴 랩슨 공식으로 해를 구한다. 독립좌표는 정확한 해라고 가정하기 때문에 식 (4)를 반복적으로 이용하여 종속좌표의 해를 구한다.

$$\Phi_{\bar{u}} \Delta \bar{u} = -\Phi, \quad \bar{u}^{i+1} = \bar{u}^i + \Delta \bar{u}^i \tag{4}$$

속도해석은 식 (3)을 한번 미분하여 독립변수의 속도 값이 정확한 해라고 가정하여 종속변수의 속도를 다음과 같이 구한다.

$$\dot{\bar{u}} = -\Phi_{\bar{u}}^{-1} \Phi_{\bar{v}} \dot{\bar{v}} \tag{5}$$

또한 현가 부분시스템의 운동방정식은 먼저 식 (2)의 운동방정식을 독립좌표와 종속좌표에 의해서 아래의 식 (6)과 같이 나누고, 수학적인 조작에 의해서, 독립변수에 관한 운동방정식으로 식 (7)과 같이 변환 한다.

$$\begin{bmatrix} \bar{M}_{uu} & \bar{M}_{uv} & \Phi_u^T \\ \bar{M}_{vu} & \bar{M}_{vv} & \Phi_v^T \\ \Phi_u & \Phi_v & 0 \end{bmatrix} \begin{bmatrix} \ddot{\bar{u}} \\ \ddot{\bar{v}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \bar{P}_u \\ \bar{P}_v \\ \gamma \end{bmatrix} - \begin{bmatrix} \bar{M}_{yqu}^T \\ \bar{M}_{yqv}^T \\ 0 \end{bmatrix} \dot{\bar{Y}}_0(t) \tag{6}$$

$$M^* \ddot{\bar{v}} = Q_q^* - Q_y \dot{\bar{Y}}_0 \tag{7}$$

여기서

$$M^* = \bar{M}_{vv} - \bar{M}_{vu} (\Phi_u^{-1} \Phi_v) - (\Phi_u^{-1} \Phi_v)^T \bar{M}_{uu}^T + (\Phi_u^{-1} \Phi_v)^T \bar{M}_{uu} (\Phi_u^{-1} \Phi_v) \tag{8}$$

$$\mathbf{Q}_q^* = \bar{\mathbf{P}}_v - (\Phi_{\bar{u}}^{-1} \Phi_{\bar{v}})^T \bar{\mathbf{P}}_u - [\bar{\mathbf{M}}_{vu} - (\Phi_{\bar{u}}^{-1} \Phi_{\bar{v}})^T \bar{\mathbf{M}}_{uu}] (\Phi_{\bar{u}}^{-1} \bar{\gamma}) \quad (9)$$

$$\mathbf{Q}_y^* = (\bar{\mathbf{M}}_{yqu} - \bar{\mathbf{M}}_{yqv} \Phi_{\bar{u}}^{-1} \Phi_{\bar{v}})^T \quad (10)$$

이다. 종속좌표에 대한 가속도와 라그랑지 승수는 식 (11)의 가속도 구속 방정식과 식 (12)에 의해 구한다.

$$\ddot{\bar{u}} = \Phi_{\bar{u}}^{-1} \bar{\gamma} - \Phi_{\bar{u}}^{-1} \Phi_{\bar{v}} \ddot{\bar{v}} \quad (11)$$

$$\lambda = (\Phi_{\bar{u}}^{-1})^T (\bar{\mathbf{P}}_u - \bar{\mathbf{M}}_{uu} \ddot{\bar{u}} - \bar{\mathbf{M}}_{uv} \ddot{\bar{v}}) \quad (12)$$

또한 식 (1)에서 필요로 하는 각 부분 시스템 별로 차체에 미치는 효과 관성 행렬과 효과 힘 벡터는 식 (13), (14)로 구한다.⁴⁾

$$\tilde{\mathbf{M}}^c = \bar{\mathbf{M}}_{yy} - \mathbf{Q}_y^{*T} (\mathbf{M}^*)^{-1} \mathbf{Q}_y^* \quad (13)$$

$$\tilde{\mathbf{P}}^c = \bar{\mathbf{P}}_y - \bar{\mathbf{M}}_{yqu} (\Phi_{\bar{u}}^{-1} \bar{\gamma}) - \mathbf{Q}_y^{*T} (\mathbf{M}^*)^{-1} \mathbf{Q}_y^* \quad (14)$$

2.2 근사함수 방법(Approximate Function Method)

근사함수 방법은 일반좌표 분할법과 매우 유사하나, 식 (3)을 수치해석적인 방법 대신, 식 (3)의 해를 다음 식과 같이 종속좌표를 독립좌표의 해석적인 근사함수로 나타내어 위치, 자세를 구하는 방법이다.

$$\bar{\mathbf{u}} = \mathbf{f}(\bar{\mathbf{v}}) \quad (15)$$

속도 해석은 식 (15)를 미분해서 구한다.

$$\dot{\bar{u}} = \frac{\partial \mathbf{f}}{\partial \bar{\mathbf{v}}} \dot{\bar{\mathbf{v}}} \equiv -\mathbf{N} \dot{\bar{\mathbf{v}}} \quad (16)$$

마찬가지로 가속도는 식 (16)을 한번 더 미분하여 구한다.

$$\ddot{\bar{u}} = \frac{\partial \mathbf{f}}{\partial \bar{\mathbf{v}}} \ddot{\bar{\mathbf{v}}} + \frac{\partial}{\partial \bar{\mathbf{v}}} \left(\frac{\partial \mathbf{f}}{\partial \bar{\mathbf{v}}} \dot{\bar{\mathbf{v}}} \right) \equiv -\mathbf{N} \ddot{\bar{\mathbf{v}}} + \mathbf{R} \quad (17)$$

부분시스템의 운동방정식은 식 (18)과 같이 일반좌표 분할법과 동일한 형태이나, 그 계수를 구하는

방법은 해석적인 근사함수를 사용하므로 식 (16)의 N 행렬과 식 (17)의 R 벡터를 사용하여 다음 식 (19)~(21)과 같이 구한다.

$$\bar{\mathbf{M}}^* \ddot{\bar{\mathbf{v}}} = \bar{\mathbf{Q}}_q^* - \bar{\mathbf{Q}}_y^* \dot{\hat{\mathbf{Y}}}_0 \quad (18)$$

$$\bar{\mathbf{M}}^* = \bar{\mathbf{M}}_{vv} - \bar{\mathbf{M}}_{vu} \mathbf{N} - \mathbf{N}^T \bar{\mathbf{M}}_{vu}^T + \mathbf{N}^T \bar{\mathbf{M}}_{uu} \mathbf{N} \quad (19)$$

$$\bar{\mathbf{Q}}_q^* = \bar{\mathbf{P}}_v - \mathbf{N}^T \bar{\mathbf{P}}_u - (\bar{\mathbf{M}}_{vu} - \mathbf{N}^T \bar{\mathbf{M}}_{uu}) \mathbf{R} \quad (20)$$

$$\bar{\mathbf{Q}}_y^* = (\bar{\mathbf{M}}_{yqu} - \bar{\mathbf{M}}_{yqv} \mathbf{N})^T \quad (21)$$

또한 식 (1)에 필요한 효과 관성행렬과 효과 힘 벡터는 다음과 같이 구한다.

$$\tilde{\mathbf{M}}^c = \bar{\mathbf{M}}_{yy} - \bar{\mathbf{Q}}_y^{*T} (\bar{\mathbf{M}}^*)^{-1} \bar{\mathbf{Q}}_y^* \quad (22)$$

$$\tilde{\mathbf{P}}^c = \bar{\mathbf{P}}_y - \bar{\mathbf{M}}_{yqu} \mathbf{R} - \bar{\mathbf{Q}}_y^{*T} (\bar{\mathbf{M}}^*)^{-1} \bar{\mathbf{Q}}_y^* \quad (23)$$

2.3 구속조건 안정화 방법(Constraint Stabilization Method)

구속조건 안정화 방법은 식 (3)의 구속조건식과 속도 해를 위한 식 (5)의 속도 구속조건식을 풀지 않고, 가속도 구속조건식을 다음과 같이 수정하여, 간접적으로 그 해를 구하는 방법이다.⁷⁾

$$\ddot{\Phi} + 2\alpha \dot{\Phi} + \beta^2 \Phi = 0 \quad (24)$$

여기서 $\alpha > 0$ 이고 $\beta \neq 0$ 이면 식 (24)의 해는 안정적으로 0으로 수렴하여 간접적으로 $\dot{\Phi} \approx \Phi \approx 0$ 을 만족하게 된다.

식 (24)로부터 보정된 가속도 구속조건식의 우변항을 다음과 같이 얻을 수 있다.

$$\hat{\gamma} = \bar{\gamma} - 2\alpha \dot{\Phi} - \beta^2 \Phi \quad (25)$$

구속조건 안정화 방법에 의한 부분 시스템의 운동방정식은 식 (26)과 같이 일반좌표 분할법과 동일한 형태이나, 계수 $\bar{\mathbf{Q}}_q^*$ 는 식 (25)를 반영하여 식 (27)과 같으며 나머지 계수는 식 (8)과 식 (10)에서 정의한 것과 동일하다.

$$\mathbf{M}^* \ddot{\bar{\mathbf{v}}} = \bar{\mathbf{Q}}_q^* - \mathbf{Q}_y^* \dot{\hat{\mathbf{Y}}}_0 \quad (26)$$

$$\begin{aligned} \bar{Q}_q^* &= \bar{P}_v - (\Phi_u^{-1} \Phi_v)^T \bar{P}_u \\ &\quad - [\bar{M}_{vu} - (\Phi_u^{-1} \Phi_v)^T \bar{M}_{uu}] (\Phi_u^{-1} \hat{\gamma}) \end{aligned} \quad (27)$$

마찬가지로 식 (1)에 필요한 효과 관성 행렬은 식 (13)과 동일하고, 효과 힘 벡터는 다음과 같이 구한다.

$$\bar{P}^c = \bar{P}_y - \bar{M}_{yq_u} (\Phi_u^{-1} \hat{\gamma}) - Q_y^{*T} (M^*)^{-1} \bar{Q}_q^* \quad (28)$$

3. 모델 비교를 위한 맥퍼슨 현가장치의 1/4 차량 시뮬레이션

3.1 1/4차량 모델

제한한 3가지 모델의 1ms 적분 간격에 대한 실시간 비교를 위하여 맥퍼슨 현가장치의 1/4차량을 모델링 하였다. 2장에서 기술한 3가지 방법에 따른 맥퍼슨 현가 부분시스템 모듈들을 생성하였다.

Fig. 3은 맥퍼슨 현가장치의 모델을 나타내고 있다. LCA(Lower Control Arm), US(Upper Strut), 너클, 타이로드, 샤시 프레임으로 구성되어 있으며, 구면조인트(spherical joint), 병진조인트(translational joint), 회전조인트(revolute joint)로 구성 되었다. 선형적인 현가 스프링과 댐퍼를 사용하였다. 또한 범퍼 스톱퍼와 리바운드 스톱퍼를 독립적인 비선형 스프링요

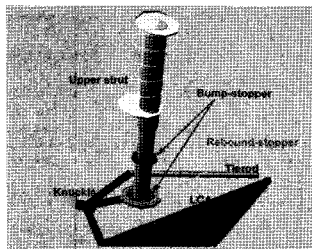


Fig. 3 McPherson Strut suspension model

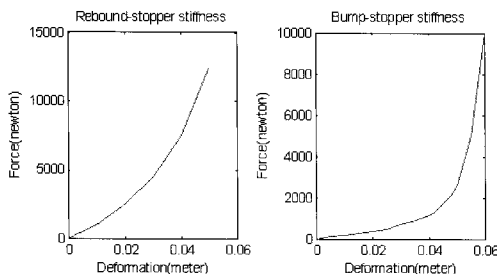


Fig. 4 Stopper stiffness characteristics

소로 취급하여 스트럿의 변위가 상하 0.15m가 될 때 스톱퍼가 작용하도록 모델링 하였다. Fig. 4는 범퍼 스톱퍼와 리바운드 스톱퍼의 강성 특성 곡선을 나타내고 있다.

3.2 Operation Count 비교

일반좌표 분할법 (GCP), 근사함수법(AFM), 및 구속조건 안정화방법(CS)들에 대한 이론적인 효율성을 비교하기 위하여 해석에 따른 operation count를 구하였다(Table 1-3). Operation count는 3가지 방법에 의한 맥퍼슨 현가 부분 시스템 모듈을 구성하고 있는 함수(서브루틴)의 실행 순서에 따라 함수별로 덧셈 및 뺄셈, 곱셈 및 나눗셈의 연산 횟수를 세었다.

근사함수 방법에서는 아담스 해석을 통해 데이터 샘플링을 하여 4차 다항식의 해석적인 근사함수를 구하였다.⁸⁾

위 결과를 살펴보면 근사함수 방법의 모델이 일반좌표분할법이나 구속조건 안정화 방법의 모델 보다 Position, Velocity, RHS, Effectmf, Acceleration 함수에서 적은 계산 량을 보였다. 특히 Position 함수에서 큰 차이를 보이고 있는데 이것은 자코비안 행렬과 자코비안 행렬의 역행렬도 구하지 않으며, 반복적인 뉴턴-랩슨(Newton-Raphson) 방법을 수행하지 않기 때문이다. 구속조건 안정화 방법도 뉴턴-랩슨 방법을 수행하지는 않지만, 식 (27)에서 보는 바와 같이, 자코비안 행렬과 그 역행렬은 이 함수에서 구

Table 1 Operation count of GCP (Generalized Coordinate Partitioning Method)

GCP	Multiplication & Division	Addition & Subtraction
Position	1500+1513*NR	833+835*NR
Velocity	227	124
State matrix	602	441
Subeqm	1023	956
RHS	140	200
Effectmf	168	150
Chassis	176	150
Accel0	45	42
Acceleration	327	304
Total count	4208+1513*NR	3200+835*NR

(NR: 뉴턴-랩슨 반복 회수)

Table 2 Operation count of AFM (Approximate Function Method)

AFM	Multiplication & Division	Addition & Subtraction
Position	673	468
Velocity	141	112
State_matrix	602	441
Subeqm	1023	956
RHS	0	0
Effectmf	136	126
Chassis	176	150
Accel0	45	42
Acceleration	311	292
Total count	3107	2587

Table 3 Operation count of CS (Constraint Stabilization Method)

CS	Multiplication & Division	Addition & Subtraction
Position	1508	836
Velocity	136	113
State_matrix	602	441
Subeqm	1023	956
RHS	172	224
Effectmf	168	150
Chassis	176	150
Accel0	45	42
Acceleration	343	312
Total count	4173	3224

하기 때문에 근사함수 방법에 비해서 더 많은 계산량을 보이고 있다.

Table 4는 NR 반복회수에 따른 3가지 모델의 비교를 나타내고 있다. 여기서 NR=0은 일반좌표 분할 방법에서 뉴턴-랩슨 방법을 수행하기 전에 먼저 구속조건식을 만족하는지를 살펴보는 데 이미 만족한 경우에는 뉴턴-랩슨을 수행하지 않는 경우를 의미한다.

Table 4에서 보는 바와 같이 근사함수 방법이 이

Table 4 Operation count comparison with different NR values

NR	GCP	AFM	CS	ratio
0	7408	5694	7397	1 : 0.77 : 1
1	9756	5694	7397	1 : 0.58 : 0.76
2	9756	5694	7397	1 : 0.47 : 0.61

론적으로 가장 효율적인 것을 알 수 있다. 그리고 적분 간격이 커지면 일반좌표분할 방법은 뉴턴-랩슨을 수행할 확률이 높아져 더욱 비효율적임을 알 수 있다.

3.3 CPU 계산시간 비교

앞서 기술한 이론적인 효율성에 대한 검증을 위해서, C code로 프로그램을 작성하여 시뮬레이션을 수행하였고 실제 CPU time을 측정하였다. 시뮬레이션은 높이 0.1m 폭이 0.2m인 반사인(half sine) 범프 통과 시뮬레이션이며, 차속은 3.6km/h 이다.

Fig. 5는 차체의 무게 중심 수직 위치, 속도, 가속도를 나타내고 있다. 3가지 모델 모두 동일한 결과를 나타내고 있다. 그러나 Fig. 6의 일반좌표 분할방법의 모델에 대한 나머지 두 모델의 상대적인 에러를 보면, 구속조건 안정화 방법이 범프 통과 후 더 많은 에러를 나타내고 있다.

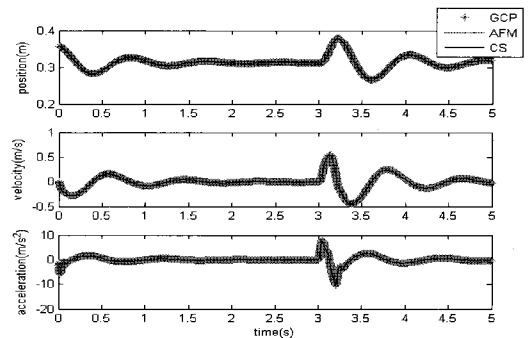


Fig. 5 Result of bump run simulation(Z axis of chassis frame)

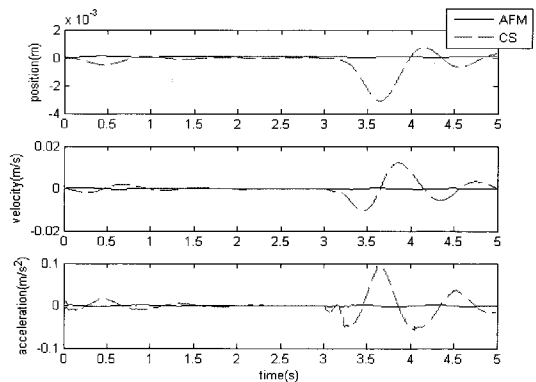


Fig. 6 Error of simulation result of AFM and CS method w.r.t. GCP (1ms)

Table 5는 적분 간격을 달리해서 시뮬레이션을 수행할 결과에 대한 실제 실행 CPU time의 비교이다. 전체 시뮬레이션 동안 총 루프수와 총 해석 시간, 그리고 한 루프를 계산하는데 걸리는 평균시간을 측정하였다. 동일한 시뮬레이션을 10번 이상 수행한 후 그 평균값을 이용하였다. 이 시뮬레이션에 사용된 PC는 Intel Pentium 4 2GHz CPU와 512MB Memory를 사용하고 있다. 그리고 적분기로는 Adams Bashforth 3rd order 공식을 사용하였다.

실제로 1ms에서 NR count를 측정해 본 결과 일반 좌표 분할법에서 모두 0인 것을 알 수 있었다. 또한 나머지 적분 간격에서는 NR 카운트가 1 이상이 되는 것을 알 수 있었다.

Table 5 CPU time comparison with different stepizes

Step size	Method	Loop count	Total time	Average	Total CPU time to real time(%)	Ratio
1ms	GCP	5004	0.2105	0.000042	4.21	1
	AFM		0.1710	0.000034	3.42	0.812
	CS		0.2035	0.000041	4.07	0.967
2ms	GCP	2505	0.1247	0.000050	2.494	1
	AFM		0.0855	0.000034	1.71	0.686
	CS		0.1020	0.000041	2.04	0.818
4ms	GCP	1255	0.0620	0.000049	1.24	1
	AFM		0.0385	0.000031	0.77	0.620
	CS		0.0545	0.000043	1.09	0.879
8ms	GCP	629	0.0385	0.000061	0.77	1
	AFM		0.0224	0.000036	0.448	0.582
	CS		0.0260	0.000041	0.52	0.675

따라서 1ms의 적분간격에서는 Table 4의 NR=0인 경우와 유사한 비(ratio)를 나타내는 것을 알 수 있다. 또한 적분 간격이 2ms 이상인 경우에서도 Table 4의 이론적인 효율성과 유사한 경향을 나타내는 것을 알 수 있다.

부분 시스템 합성방법을 이용하여 전차량을 해석하는 경우 2장 초반부에 기술한 것처럼 현재 부분 시스템 루틴을 4번 사용하므로, 전차량 모델에서 걸리는 시간은 1/4 차량 시뮬레이션의 4-5배정도로 생각할 수 있다. 따라서 가장 느린 일반좌표 분할법이 실제 시간의 4.21%을 사용하므로, 전차량 모델에서

는 약 30% 이내의 시간을 사용 하여 실시간 성을 확보할 수 있을 것으로 예상된다.⁴⁾

5. 결론

본 논문에서는 짧은 적분 간격(1ms)의 해석을 요구하는 HILS에 적합한 모델을 제시하기 위하여, 부분 시스템 합성방법에 근거를 둔 일반좌표 분할법, 근사함수방법, 구속조건식의 안정화 방법의 3가지 모델을 제시 비교하였다.

효율성 비교를 위해서 맥퍼슨 스트럿 현가장치의 1/4차량에 대해서 프로그램을 개발하였고, 공식에 사용된 연산 량을 이론적으로 비교하였다. 또한 실제 CPU 시간 측정하여 3가지 방법에 대한 실시간 성을 비교 분석하였다. 결론은 다음과 같다.

- 1) 이론적인 효율성과 실제 CPU 시간 측정을 통하여 근사함수 방법이 가장 효율적임(약 1.2배)을 알 수 있었다.
- 2) 1ms 적분간격에서 구속조건 안정화 방법은 일반 좌표 분할법과 거의 차이가 나지 않음을 알 수 있었다. 그러나 적분간격이 커질수록 더 효율적임을 알 수 있었다.
- 3) 3가지 방법 모두 1ms 적분 간격에서 실시간 성을 확보함을 알 수 있었다.

후 기

본 연구는 충남대학교 2004년도 자체 연구비 지원에 의해서 수행되었음.

References

- 1) CarSim User Manual Mechanical Simulation Corporation, USA, 2005.
- 2) Vedyna 3.7 User Manual, TESIS, Germany, 2003.
- 3) G. J. Choi, K. H. Noh, Y. M. Yoo and H. Kim, "A Strategy to Evaluate Semi-Active Suspension System using Real-Time Hardware-in-the-Loop Simulation," Transactions of KSAE, Vol.9, No.6, pp.186-194, 2001.
- 4) S.-S. Kim and Y. S. Oh, "A Real-Time Multibody Vehicle Dynamics Model using

- Subsystem Synthesis Method,” Proceedings of ASME DETC01, DETC2001/VIB21311, 2001.
- 5) S.-S. Kim, “A Subsystem Synthesis Method for Efficient Vehicle Multibody Dynamics,” *Multibody System Dynamics*, Vol.7, pp.189-207, 2002.
 - 6) F. F. Tsai and E. J. Haug, “Automated Methods for High Speed Simulation of Multibody Dynamic Systems,” Technical Report R-47, Center for Computer Aided Design, The University of Iowa, 1989.
 - 7) E. J. Haug, *Computer-Aided Kinematics and Dynamics of Mechanical Systems*, Allyn and Bacon, 1989.
 - 8) S.-S. Kim, C. Lee, W. Jeong and S. Lee, “Approximate Function Method for Real Time Multibody Vehicle Dynamics Model,” *Transactions of KSAE* (To be published).