

조인트 좌표계를 이용한 부분시스템 합성방법의 내재적 적분기법

조준연* · 김명호* · 김성수**†

* 충남대학교 기계·기계설계·메카트로닉스공학과, ** 충남대학교 메카트로닉스공학과

An Implicit Integration Method for Joint Coordinate Subsystem Synthesis Method

Junyoun Jo*, Myoung Ho Kim* and Sung-Soo Kim**†

* Graduate school of Mechanical·Mechanical Design·Mechatronics Engineering, Chungnam Nat'l Univ.,

** Dept. of Mechatronics Engineering, Chungnam Nat'l Univ.

(Received December 12, 2011; Revised January 2, 2012; Accepted January 19, 2012)

Key Words : Implicit Integration(내재적 적분기법), Joint Coordinates(조인트 좌표계), Subsystem Synthesis Method(부분시스템 합성방법)

초록: 본 논문에서는 효율적인 다물체 시스템의 동역학 해석을 위해 조인트 좌표계 기반의 부분시스템 합성방법을 위한 내재적 적분기법을 개발하였다. 부분시스템 합성방법의 내재적 적분기법을 검증하기 위해, 동일 구조를 갖는 6 개의 독립적인 현가 부분시스템으로 이루어진 무인 로봇 차량에 적용하였다. 내재적 적분기법의 복잡한 시스템 자코비언을 효율적으로 생성하기 위해 기호연산법을 도입하였다. 제안한 방법의 검증을 위해 험지주행 시뮬레이션을 수행하였으며, 일반적인 내재적 적분기법 모델과 그 결과를 비교하였다. 또한 효율성을 확인하기 위해 해석 시간을 비교하였다.

Abstract: To analyze a multibody system, this paper proposes an implicit numerical integration method for joint coordinates subsystem synthesis method. To verify the proposed method, a multibody model for an unmanned robot vehicle, which consists of six identical independent suspension systems, is developed. The symbolic method is applied to compute the system Jacobian matrix for the implicit integration method. The proposed method is also verified by performing rough terrain run-over simulation in comparison with the conventional implicit integration method. In addition, to evaluate the efficiency of the proposed method, the CPU time obtained by using this method is compared with that obtained by using the conventional implicit method.

1. 서 론

최근 무인 차량에 대한 연구는 국내외를 통해 활발히 진행되고 있다. 특히 자율 주행 로봇의 경우 주어진 경로에 대해서 차량의 주행 가능성을 확인하고 주행 가능한 속도를 결정하기 위해서는 실시간 다물체 동역학 해석이 이루어져야 한다.

실시간 다물체 동역학 해석을 위해 효율적인 다물체 동역학 해석 기법과 강인한 적분기법의 적용이 필수적이다. 효율적인 다물체 동역학 해석을 위해 외재적 적분기를 적용한 조인트 좌표계 기반

의 부분시스템 합성방법이 개발되었다.⁽¹⁾ 조인트 좌표계를 기반의 운동방정식은 직교 좌표계 기반의 운동방정식에 비해 그 크기가 작아지므로 계산 효율을 증대시킬 수 있다. 또한 부분시스템 합성방법을 이용함으로써 크기가 큰 전체 시스템 운동방정식을 계산하는 대신, 크기가 작은 운동방정식으로 나누어 계산하므로 그 계산 효율을 극대화할 수 있다. 또한 프로그램 관점에서 부분시스템 합성방법은 모듈 구조를 제공하기 때문에 각 부분시스템 별로 다양한 공식의 적용이 가능하며, 부분시스템의 구조가 동일한 경우 입력데이터만 변경하면 하나의 프로그램 모듈을 이용하여 여러 부분시스템을 계산할 수 있으므로 사용자에게 편의성을 제공한다. 그러나, 기존 개발된 부분시스템 방법에서는 외재적 적분기법을 사용하기 때문에

† Corresponding Author, sookim@cnu.ac.kr

stiff 한 시스템에 대해서는 적분간격이 매우 작아져야 하는 단점이 있다. 이러한 경우 외재적 적분 방법보다 내재적 적분방법을 적용하는 것이 효과적이다.

Stiff 한 요소가 포함된 동역학 모델의 효율적인 동특성 해석을 위해 Newmark- β , HHT- α 알고리즘을 이용한 내재적 적분기법에 대한 연구가 수행되었다.^(2,3) 내재적 적분방법은 동역학 공식에 적용이 복잡하고 계산량이 많지만, 비교적 큰 적분간격에서도 안정적인 적분결과를 기대할 수 있다. 그러나, 내재적 적분기법은 적분을 위해 현재 물체의 상태(state)가 필요하므로 Newton-Raphson 알고리즘을 이용한 반복계산이 필요하다. 이를 위해 전체 시스템 운동방정식의 편미분 항인 시스템 자코비언 행렬을 구해야 하는 어려움이 있다.

본 논문에서는 효율적인 다물체 동역학 해석을 위해, 조인트 좌표계 기반의 부분시스템 합성방법을 위한 내재적 적분기법을 개발하는데 그 목적이 있다. 또한, 내재적 적분기법의 복잡한 시스템 자코비언의 효율적인 생성을 위해 기호연산법을 도입하였다. 제안한 방법의 타당성을 평가하기 위해 무인 로봇 차량의 험지주행 시뮬레이션을 수행하고, 일반적인 내재적 적분기법과 해를 비교하였다.

2. 일반적인 다물체 동역학 해석 방법의 내재적 적분기법

일반적으로 HHT(Hilber-Hughes-Talyor) 수치적분 방법⁽⁵⁾은 구조해석에서 널리 사용되고 있는 방법이다. 이 방법은 2 차 선형 미분방정식에서는 A-stable 하면서 2 차수 정확성과 수치 감쇄효과의 반영이 용이하다.

식 (1)은 개루프 구조를 갖는 조인트 좌표계의 다물체 운동방정식을 나타낸다.

$$\begin{bmatrix} \bar{\mathbf{M}}_{yy} & \bar{\mathbf{M}}_{yq} \\ \bar{\mathbf{M}}_{yq}^T & \bar{\mathbf{M}}_{qq} \end{bmatrix} \begin{bmatrix} \dot{\hat{\mathbf{y}}}_0 \\ \ddot{\hat{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{P}}_y \\ \bar{\mathbf{P}}_q \end{bmatrix} \quad (1)$$

$\ddot{\hat{\mathbf{q}}}$ 은 상대 조인트 가속도를 나타내며, $\dot{\hat{\mathbf{y}}}_0$ 은 물체 0 에 대한 상태 가속도 벡터를 나타낸다. 식 (1)의 관성 행렬과 힘 벡터의 각 요소들은 다음과 같은 행렬식으로 표현된다.

$$\bar{\mathbf{M}}_{yy} = \hat{\mathbf{M}}_0 + \mathbf{K}_1 \quad (2)$$

$$\bar{\mathbf{M}}_{yq} = [\mathbf{K}_1 \mathbf{B}_1 \quad \mathbf{K}_2 \mathbf{B}_2 \quad \dots \quad \mathbf{K}_n \mathbf{B}_n] \quad (3)$$

$$\bar{\mathbf{M}}_{qq} = \begin{bmatrix} \mathbf{B}_1^T \mathbf{K}_1 \mathbf{B}_1 & \mathbf{B}_1^T \mathbf{K}_2 \mathbf{B}_2 & \dots & \mathbf{B}_1^T \mathbf{K}_n \mathbf{B}_n \\ & \mathbf{B}_2^T \mathbf{K}_2 \mathbf{B}_2 & \dots & \mathbf{B}_2^T \mathbf{K}_n \mathbf{B}_n \\ & & \ddots & \vdots \\ & & & \text{symm} & \mathbf{B}_n^T \mathbf{K}_n \mathbf{B}_n \end{bmatrix} \quad (4)$$

$$\bar{\mathbf{P}}_y = \hat{\mathbf{Q}}_0 + (\mathbf{L}_1 - \mathbf{K}_1 \mathbf{D}_1) \quad (5)$$

$$\bar{\mathbf{P}}_q = \begin{bmatrix} \mathbf{B}_1^T (\mathbf{L}_1 - \mathbf{K}_1 \mathbf{D}_1) \\ \mathbf{B}_2^T (\mathbf{L}_2 - \mathbf{K}_2 \sum_{i=1}^2 \mathbf{D}_i) \\ \vdots \\ \mathbf{B}_n^T (\mathbf{L}_n - \mathbf{K}_n \sum_{i=1}^n \mathbf{D}_i) \end{bmatrix} \quad (6)$$

$$\text{여기서, } \mathbf{B}_j = \begin{bmatrix} \frac{\partial \mathbf{d}_j}{\partial \mathbf{q}_j} + \tilde{\mathbf{r}}_j \mathbf{H}_j \\ \mathbf{H}_j \end{bmatrix} \text{ 로 조인트에 따른 속}$$

도 변환 행렬(velocity transformation matrix)를 나타내며, 시스템 관성 행렬과 시스템 힘 벡터 요소들인 \mathbf{K}_i 와 \mathbf{L}_i 에 관하여 다음과 같은 역순환(backward recurrence) 공식이 유도된다.

$$\mathbf{K}_n = \hat{\mathbf{M}}_n \quad (7)$$

$$\mathbf{K}_i = \mathbf{K}_{i+1} + \hat{\mathbf{M}}_i, \quad i = n-1, n-2, \dots, 1$$

$$\mathbf{L}_n = \hat{\mathbf{Q}}_n \quad (8)$$

$$\mathbf{L}_i = \hat{\mathbf{Q}}_i + (\mathbf{L}_{i+1} - \mathbf{K}_{i+1} \mathbf{D}_{i+1}), \quad i = n-1, n-2, \dots, 1$$

식 (1) 에 HHT- α 적분기법을 적용하면 식 (9)의 보완된 운동방정식을 얻을 수 있다.⁽³⁾

$$\Psi = \begin{bmatrix} \Psi^0 \\ \Psi^i \end{bmatrix} \equiv \begin{bmatrix} \bar{\mathbf{M}}_{yy} \dot{\hat{\mathbf{y}}}_0 + \bar{\mathbf{M}}_{yq} \ddot{\hat{\mathbf{q}}} - \bar{\mathbf{P}}_y \\ \bar{\mathbf{M}}_{yq}^T \dot{\hat{\mathbf{y}}}_0 + \bar{\mathbf{M}}_{qq} \ddot{\hat{\mathbf{q}}} - \bar{\mathbf{P}}_q \end{bmatrix} = \mathbf{0} \quad (9)$$

여기서,

$$\bar{\mathbf{P}}_q = \mathbf{B}_i^T \left\{ (1+\alpha)(\hat{\mathbf{Q}}_i)_{n+1} - \alpha(\hat{\mathbf{Q}}_i)_n - \mathbf{K}_i \sum_{i=1}^{nb} \mathbf{D}_i \right\}$$

$$\bar{\mathbf{P}}_y = \left\{ (1+\alpha)(\hat{\mathbf{Q}}_0 + \hat{\mathbf{Q}}_i)_{n+1} + \mathbf{K}_i \mathbf{D}_i \right\} - \alpha(\hat{\mathbf{Q}}_0 + \hat{\mathbf{Q}}_i)_n$$

로 HHT- α 방법의 수치 감쇄 효과가 포함된다.

내재적 적분방법에서는 미지수인 $\dot{\hat{\mathbf{y}}}_0$ 과 $\ddot{\hat{\mathbf{q}}}$ 을 다음과 같이 Newton-Raphson 알고리즘을 적용한 반복 계산을 통해 구할 수 있다.

$$\begin{bmatrix} \Psi_{\dot{y}_0}^0 & \Psi_{\dot{q}}^0 \\ \Psi_{\dot{y}_0}^i & \Psi_{\dot{q}}^i \end{bmatrix} \begin{bmatrix} \Delta \dot{y}_0 \\ \Delta \dot{q} \end{bmatrix} = \begin{bmatrix} -\Psi^0 \\ -\Psi^i \end{bmatrix} \quad (10)$$

$$\begin{aligned} \dot{y}_0^{(k+1)} &= \dot{y}_0^{(k)} + \Delta \dot{y} \\ \dot{q}^{(k+1)} &= \dot{q}^{(k)} + \Delta \dot{q} \end{aligned}$$

식 (10)의 $\Psi_{\dot{y}_0}^0, \Psi_{\dot{q}}^0, \Psi_{\dot{y}_0}^i, \Psi_{\dot{q}}^i$ 는 식 (9)의 전체 시스템 운동방정식을 미지수에 대해 편미분한 시스템 자코비언 행렬을 나타내며 연쇄법칙(Chain rule)을 이용하여 다음과 같이 일반적인 식으로 유도할 수 있다.^(5,6)

$$\Psi_{\dot{y}_0}^0 = \mathbf{M}_{yy} + \left\{ \beta h^2 (\Psi^0)_y + \gamma h (\Psi^0)_{\dot{y}} + (\Psi^0)_{\ddot{y}} \right\} (\mathbf{T} + \gamma h \dot{\mathbf{T}}) \quad (11)$$

$$\Psi_{\dot{q}}^0 = \mathbf{M}_{yq} + \left\{ \beta h^2 (\Psi^0)_q + \gamma h (\Psi^0)_{\dot{q}} \right\} \quad (12)$$

$$\Psi_{\dot{y}_0}^i = \mathbf{M}_{yq}^T + \left\{ \beta h^2 (\Psi^i)_y + \gamma h (\Psi^i)_{\dot{y}} + (\Psi^i)_{\ddot{y}} \right\} (\mathbf{T} + \gamma h \dot{\mathbf{T}}) \quad (13)$$

$$\Psi_{\dot{q}}^i = \mathbf{M}_{qq} + \left\{ \beta h^2 (\Psi^i)_q + \gamma h (\Psi^i)_{\dot{q}} + (\Psi^i)_{\ddot{q}} \right\} \quad (14)$$

여기서, β 와 γ 는 HHT- α 적분상수를 의미하고, h 는 적분간격의 의미한다.

3. 부분시스템 합성방법의 내재적 적분 기법

식 (10)의 시스템 운동방정식에 부분시스템 합성방법을 적용하면 식 (15)와 같은 기준물체에 대한 보완된 운동방정식과 식 (16)과 같은 부분시스템에 대한 보완된 운동방정식으로 나누어 쓸 수 있다.

$$\Psi_{\dot{y}_0}^0 \Delta \dot{y}_0 + \Psi_{\dot{q}}^i \Delta \dot{q} = -\Psi^0 \quad (15)$$

$$\dot{y}_0^{(k+1)} = \dot{y}_0^{(k)} + \Delta \dot{y}_0$$

$$\Psi_{\dot{q}}^i \Delta \dot{q} = -\Psi^i - \Psi_{\dot{y}_0}^i \Delta \dot{y}_0 \quad (16)$$

$$\dot{q}^{(k+1)} = \dot{q}^{(k)} + \Delta \dot{q}$$

식 (16)의 부분시스템에 대한 운동방정식을 $\Delta \dot{q}$ 에 대해 정리하여 식 (15)에 대입하면 기준물체에 대한 운동방정식은 식 (17)과 같이 다시 정리할

수 있다.

$$\begin{aligned} \left\{ \Psi_{\dot{y}_0}^0 + \tilde{\Psi}_{\dot{q}}^c \right\} \Delta \dot{y}_0 &= \left\{ -\Psi^0 + \tilde{\Psi}^c \right\} \\ \dot{y}_0^{(k+1)} &= \dot{y}_0^{(k)} + \Delta \dot{y}_0 \end{aligned} \quad (17)$$

여기서, $\tilde{\Psi}_{\dot{q}}^c$ 와 $\tilde{\Psi}^c$ 는 부분시스템으로부터 오는 효과 항들로서 각각 식 (18)-(19)로 정의할 수 있다.

$$\tilde{\Psi}_{\dot{q}}^c = -\Psi_{\dot{q}}^0 (\Psi_{\dot{q}}^i)^{-1} \Psi_{\dot{y}_0}^i \quad (18)$$

$$\tilde{\Psi}^c = \Psi_{\dot{q}}^0 (\Psi_{\dot{q}}^i)^{-1} \Psi^i \quad (19)$$

먼저, 식 (17)을 계산하여 기준물체에 대한 $\Delta \dot{y}_0$ 을 구하고 이를 식 (16)에 대입하여 부분시스템에 대한 \dot{q} 을 순차적으로 구할 수 있다.

조인트 좌표계 좌표계로 표현된 운동방정식은 그 공식의 구성이 매우 복잡하기 때문에 식 (11)-(14) 내부의 편미분 항들은 직접 유도하기가 매우 어렵다. 따라서 본 연구에서는 복잡한 시스템 자코비언 행렬을 효과적으로 유도하기 위해 기호연산법을 도입하였다.

4. 기호연산법을 이용한 시스템 자코비언 생성

다물체 시스템의 운동방정식을 생성하는 데는 크게 수치적 기법과 기호연산법이 이용된다. 수치적 기법은 운동방정식을 생성하는데 필요한 변수 값을 계속 계산해가는 방법으로 DADS⁽⁷⁾ 및 ADAMS⁽⁸⁾ 등의 상용 프로그램에서 사용하고 있다. 기호연산법은 기호형태로 운동방정식을 세워 한번에 계산하는 방법으로 NEWEUL 및 AUTOSIM 등의 상용 프로그램에서 채택하고 있다. 기호연산법은 초기 운동방정식의 생성 과정에서 시간이 더 소요되는 단점이 있으나, 생성된 후에는 더 빨리 계산할 수 있으며, 또한 대상 시스템에 관계되는 부분만을 포함한 간편하고 효율적인 코드를 생성할 수 있는 장점을 가지고 있다. 기호연산법을 적용할 경우 각 과정에서 생성된 기호 형태의 항들이 계속 더해지고 곱해지면서 매우 긴 코드가 생성되는데, 이로 인한 계산시간이 불필요하게 증가되고 또한 메모리가 작은 컴퓨터의 경우 용량을 초과할 수 있다. 따라서 기호연산법으로 생성된 운동방정식을 단순화 하는 기능은 기호연산법에서 필수적이다.

```
Pq
[-m1 dph1 rh1px dph0 s0lpy cos(q1) + m1 dph1 rh1py dph0 s0lpx sin(q1)
 -m1 dph1 rh1px dph0 s0lpy sin(q1) - m1 dph1 rh1px dph0 s0lpx cos(q1)
 -k q1 + k ini + cos(phi1) rh1px m1 g - sin(phi1) rh1py m1 g
 -cos(phi1) rh1px m1 dph1 dy0 + sin(phi1) rh1py m1 dph1 dy0
 + m1 dq1 sin(phi1) rh1px dy0 + m1 dq1 cos(phi1) rh1py dy0
 + m1 dq1 cos(phi1) rh1px dx0 - m1 dq1 sin(phi1) rh1py dx0
 + m1 dph1 cos(phi1) rh1py dx0 + m1 dph1 sin(phi1) rh1px dx0 - c dq1
 -m1 dph12 rh1px2 + m1 dq1 rh1py dph0 s0lpy sin(q1) - m1 dph12 rh1py2
 + m1 dq1 rh1px dph0 s0lpx sin(q1) - m1 dq1 rh1px dph0 s0lpy cos(q1)
 + m1 dq1 rh1py dph0 s0lpx cos(q1)]
```

```
jacobian(Pq, q1)
[m1 dph1 rh1py dph0 s0lpy sin(q1) + m1 dph1 rh1py dph0 s0lpx cos(q1)
 -m1 dph1 rh1px dph0 s0lpy cos(q1) + m1 dph1 rh1px dph0 s0lpx sin(q1) - k
 + m1 dq1 rh1py dph0 s0lpy cos(q1) + m1 dq1 rh1px dph0 s0lpx cos(q1)
 + m1 dq1 rh1px dph0 s0lpy sin(q1) - m1 dq1 rh1py dph0 s0lpx sin(q1)]
```

Fig. 1 Symbolic code generation example

```
with(CodeGeneration) :
C(jacobian(Pq, q1), resultname="Pq_q1")
Pq_q1[0][0] = m1 * dph1 * rh1py * dph0 * s0lpy * sin(q1) + m1 * dph1 * rh1py *
dph0 * s0lpx * cos(q1) - m1 * dph1 * rh1px * dph0 * s0lpy * cos(q1) + m1 * dph1
* rh1px * dph0 * s0lpx * sin(q1) - k + m1 * dq1 * rh1py * dph0 * s0lpy * cos
(q1) + m1 * dq1 * rh1px * dph0 * s0lpx * cos(q1) + m1 * dq1 * rh1px * dph0 *
s0lpy * sin(q1) - m1 * dq1 * rh1py * dph0 * s0lpx * sin(q1);
```

Fig. 2 C code generation using MAPLE software

본 논문에서는 내재적 적분기법의 복잡한 시스템 자코비언의 생성을 위해 기호연산법을 도입하였으며, 기호처리 소프트웨어로는 처리속도가 빠르고 기능이 다양한 MAPLE 소프트웨어를 이용하였다.⁽⁹⁾ 먼저 식 (11)-(14)의 일반적인 수식은 수치적분 기법을 이용하여 계산하고, 내부의 편미분 항들은 기호연산법을 이용하여 계산하는 혼합된 방식을 이용하였다.

Fig. 1은 MAPLE 소프트웨어를 이용하여 편미분을 수행하는 예를 나타낸다.

Fig. 1과 같이 기호연산법으로 생성된 코드는 기호형태의 항들의 합과 곱의 형태로 구성된다. 이렇게 생성된 코드를 MAPLE 소프트웨어의 기능 중 하나인 코드 생성(CodeGeneration)을 이용하여 다른 상용프로그램의 코드를 생성할 수 있다. Fig. 2는 코드 생성을 이용하여 Fig. 1의 결과를 C 코드로 변환한 결과를 나타낸다.

기호연산법으로 생성된 코드는 중복계산이 많이 존재하게 된다. 또한 시스템 운동방정식의 형태가 복잡할수록 중복계산항의 수가 많아질 가능성이 있으므로 이러한 중복계산을 최소화 하는 과정이 필요하다.

5. 무인 로봇 차량의 험지주행 시뮬레이션

제안된 방법을 검증하기 위해 Fig. 3과 같은 6

Table 1 Force properties of the suspension

Force properties	Values
Rotational spring coefficient	31.416 Nm/degree
Rotational damping coefficient	10.47 Nm·s/degree
Tire vertical stiffness	200,000 N/m



Fig. 3 An unmanned robot vehicle

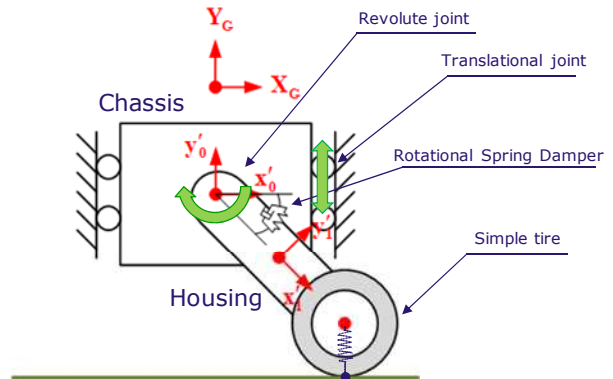


Fig. 4 2 dimensional 1/6 unmanned robot vehicle

개의 독립적인 현가장치로 구성된 무인 로봇 차량에 적용하였다. 부분시스템 합성방법을 적용하기 위해 1/6 차량 부분시스템 모듈을 개발하고, 부분 부분시스템부터 전차량 시스템으로 확장하였다. 1/6 차량 부분시스템은 실제 복잡한 구조의 현가장치 대신 Fig. 4의 단순화된 2D 모델을 이용하였다.

Fig. 4의 단순모델에 대한 물성치를 Table 1에 나타내었다.

먼저 개발된 1/6 차량 모델을 이용하여 PSD road D 등급의 험지주행 시뮬레이션을 수행하였다. 시뮬레이션은 차량이 10km/h의 속도로 주행할 때 차체의 수직방향 거동을 확인하였다. Fig. 5는 시

Table 2 Simulation results comparison (1/6 model)

Method	Conventional Implicit	Subsystem Implicit
Stepsize	4.9ms	4.9ms
CPU time	0.3437s	0.3524s
Ratio	1	1.03
Ratio of Real time	1.15%	1.18%

Table 3 Simulation results comparison (1/2 model)

Method	Conventional Implicit	Subsystem Implicit
Stepsize	4.9ms	4.9ms
CPU time	1.485s	1.082s
Ratio	1	0.73
Ratio of Real time	4.95%	3.61%

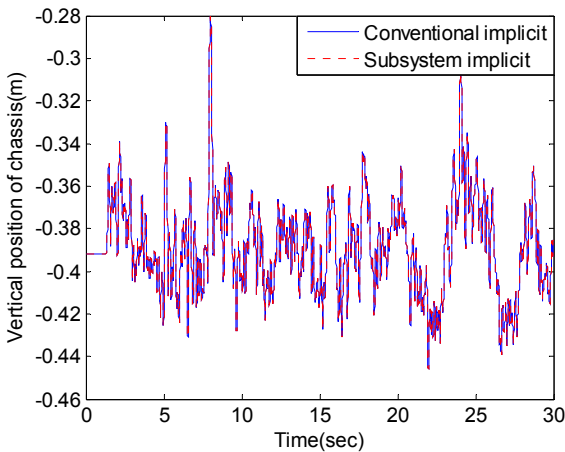


Fig. 5 Vertical position of the chassis (1/6 model)

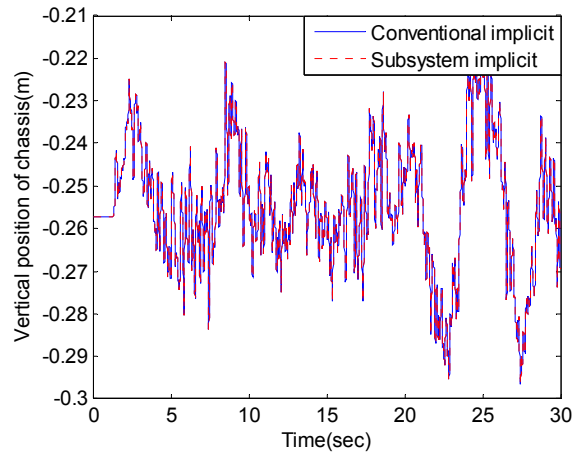


Fig. 6 Vertical position of the chassis (1/2 model)

물레이션 결과를 나타낸다.

위 결과로부터 일반적인 방법과 부분시스템 합성방법 모두 동일한 해를 갖는 것을 알 수 있다. 두 모델에 대한 효율성을 확인하기 위해 CPU time 을 측정하였다. 측정 결과는 Table 2 에 정리하였다.

위 결과로부터 1/6 차량의 경우 일반적인 해석 방법과 부분시스템 합성방법의 CPU time 이 거의 유사하게 나타나는 것을 알 수 있다.

개발된 1/6 차량 모델을 이용하여 전차량 모델로 확장하여 동일한 험지주행(PSD road D 등급) 시물레이션을 수행하였다. Fig. 6-7 은 시물레이션 결과를 나타내고 있다.

Fig. 6-7 의 결과로부터 제안한 부분시스템 합성 방법과 일반적인 방법 모두 동일한 해를 갖는 것을 알 수 있다. 두 방법에 대한 효율성을 비교하기 위해 CPU time 를 측정하였으며, 그 결과를 Table 3 에 정리하였다.

CPU time 측정 결과 일반적인 내재적 적분방법에 비해 부분시스템 합성방법의 내재적 적분방법이 약 27% 정도의 빠른 계산 효율을 갖는 것을

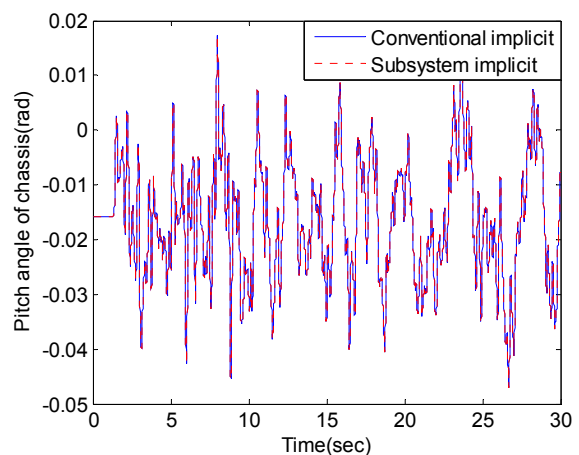


Fig. 7 Pitch angle of chassis (1/2 model)

알 수 있다.

물체의 개수에 따른 부분시스템 합성방법의 효율성을 확인하기 위해 1/6 차량 모델과 전차량 모델에 대한 CPU time 를 비교하였다. Fig. 8 에는 물체의 개수에 따른 일반적인 해석 방법과 부분시스템 합성방법의 해석 시간을 비교하였다.

1/6 차량 모델의 시물레이션 결과는 일반적인 방

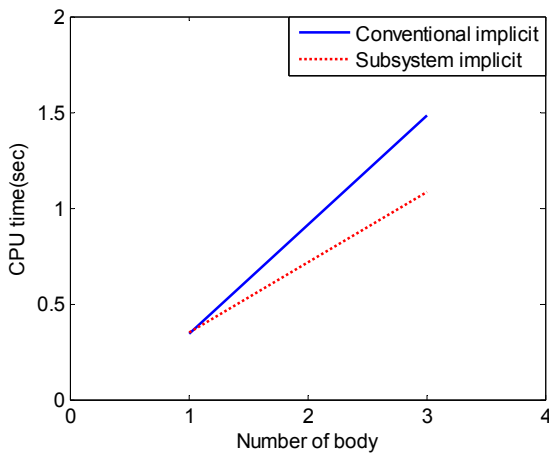


Fig. 8 Computational efficiency comparison between conventional method and subsystem synthesis method

법과 부분시스템 합성방법의 차이가 나타나지 않지만 1/2 차량으로 확장할 경우 일반적인 방법에 비해 부분시스템 합성방법의 CPU time의 증가량이 적게 나타난다. 위 결과로부터, 물체의 수가 증가할수록 일반적인 해석 방법에 비해 부분시스템 합성방법의 계산 효율이 증가하는 것을 알 수 있다.

6. 결론

본 논문에서는 다물체 무인 로봇 차량의 효율적인 동역학 해석을 위해 조인트 좌표계 기반 부분시스템 합성방법의 내재적 적분방법이 개발되었으며 복잡한 시스템 자코비언을 효과적으로 계산하기 위하여 기호 연산법과 수치적 기법을 혼합하여 적용하였다. 제안한 부분시스템 합성방법의 효과를 검증하기 위해, 일반적인 내재적 적분방법과 비교 연구를 수행하였다. 험지 주행 시뮬레이션 결과, 1/6 차량의 경우 그 효율성에 있어서 일반적인 방법과 부분시스템 방법이 거의 비슷하게 나타났지만, 부분시스템 개수가 증가한 1/2 차량의 경우에는 제안한 부분시스템 합성방법의 내재적 적분방법이 일반적인 내재적 적분방법보다 약 30% 정도 계산 효율성이 더 높아지는 것을 확인하였다.

후 기

본 연구는 한국과학기술원 국방무인화기술 특화

연구센터를 통한 방위사업청과 국방과학연구소 연구비 지원으로 수행되었습니다.

참고문헌

- (1) Kim, S. S., 2002, "A Subsystem Synthesis Method for Efficient Vehicle Multibody Dynamics," *Multibody System Dynamics*, Vol. 7, pp. 189~207.
- (2) Haug, E. J., Negrut, D. and Iancu, M., 1997, "A State-Space-Based Implicit Integration Algorithm for Differential-Algebraic Equations of Multibody Dynamics," *Mechanics of Structures and Machines*, Vol. 25, No. 3, pp. 311~334.
- (3) Negrut, D., Rampalli, R. and Ottarsson, G., 2005, "On the Use of the HHT Method in the Context of Index 3 Differential Algebraic Equations of Multibody Dynamics," *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, pp. 1~12.
- (4) Tsai, F.F. and Haug, E. J., 1989, "Automated Methods for High Speed Simulation of Multibody Dynamics Systems," *Technical Report R-47*, Center for Computer Aided Design, The University of Iowa.
- (5) Haug, E. J. and Yen, J., 1992, "Implicit Numerical Integration of Constrained Equations of Motion Via Generalized Coordinate Partitioning," *ASME Journal of Mechanical Design*, Vol. 114, pp. 296~304.
- (6) Haug, E. J., Negrut, D. and Iancu, M., 1997, "A State-Space-Based Implicit Integration Algorithm for Differential-Algebraic Equations of Multibody Dynamics," *Mechanics of Structures and Machines*, 25(3), pp. 311~334.
- (7) Rudranarayan, M. M., Kishor, D. B. and Kurt, S. A., 2007, "A Divide-and-Conquer Direct Differentiation Approach for Multibody System Sensitivity Analysis," *Structural and Multidisciplinary Optimization*, Vol. 35, No. 5, pp. 413~429.
- (8) Cuadrado, J., Dopico, D., Barreiro, A. and Delgado, E., 2009, "Real-Time State Observers Based on Multibody Models and the Extended Kalman Filter," *Journal of Mechanical Science and Technology*, Vol. 23, No. 4, pp. 894~900.
- (9) CADSI, 1993, DADS User's Manual.
- (10) MSC.software Inc., 2005, ADAMS 2005 manual.
- (11) Maplesoft, 2010, Maple 14 the Essential Tool for Mathematics and Modeling User Manual.