

<학술논문>

DOI <https://doi.org/10.3795/KSME-A.2017.41.6.507>

ISSN 1226-4873(Print)

2288-5226(Online)

병렬 처리를 이용한 부분 시스템 기반 유연다물체 동역학의 효율적인 해석 연구

한종부* · 송하준* · 김성수*†

* 충남대학교 메카트로닉스 공학과

Study on Parallel Processing for Efficient Flexible Multibody Analysis based on Subsystem Synthesis Method

Jong-Boo Han * Hajun Song * and Sung-Soo Kim *†

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

(Received February 16, 2017 ; Revised March 5, 2017 ; Accepted March 7, 2017)

Key Words: Flexible Multibody Dynamics(유연다물체동역학), Parallel Processing(병렬처리), Rotating Blade System(회전 블레이드 시스템)

초록: 많은 절점 자유도로 표현이 되는 유연다물체 시스템의 효율적인 해석을 위해서는 병렬처리 기법이 적용될 수 있다. 이 분야에서의 병렬처리기법은 주로 선형대수방정식의 효율적인 해법에 초점이 맞추어 연구가 진행되었다. 본 논문에서는 기존의 방법과는 달리 병렬처리에 적합한 유연다물체 동역학 공식을 부분 시스템 합성방법을 이용하여 개발하고, OpenMP를 사용한 효율적인 병렬처리 방식을 제안하였다. 서로 다른 두 가지 병렬처리 방식을 3개의 동일한 유연체 회전 날개 시스템 시뮬레이션 통하여 비교하였다. 또한 실제의 CPU시간을 비교하여 제안한 병렬처리 방법의 효율성을 고찰하였다.

Abstract: Flexible multibody simulations are widely used in the industry to design mechanical systems. In flexible multibody dynamics, deformation coordinates are described either relatively in the body reference frame that is floating in the space or in the inertial reference frame. Moreover, these deformation coordinates are generated based on the discretization of the body according to the finite element approach. Therefore, the formulation of the flexible multibody system always deals with a huge number of degrees of freedom and the numerical solution methods require a substantial amount of computational time. Parallel computational methods are a solution for efficient computation. However, most of the parallel computational methods are focused on the efficient solution of large-sized linear equations. For multibody analysis, we need to develop an efficient formulation that could be suitable for parallel computation. In this paper, we developed a subsystem synthesis method for a flexible multibody system and proposed efficient parallel computational schemes based on the OpenMP API in order to achieve efficient computation. Simulations of a rotating blade system, which consists of three identical blades, were carried out with two different parallel computational schemes. Actual CPU times were measured to investigate the efficiency of the proposed parallel schemes.

1. 서론

유연다물체 동역학은 80년대 중반부터 기계시스템의 경량화에 따른 부품의 진동 특성을 고려하기

위해서 개발되어 왔다.⁽¹⁾ 이러한 유연 다물체 시스템에서 변형을 고려하는 방법으로는 물체 기준 좌표계의 강제 운동을 기술하는 좌표계와 물체 기준 좌표계로부터 상대적인 변형을 기술하는 변형좌표를 사용하여 운동방정식을 도출하는 부유좌표계 공식(Floating Reference frame Formulation)이 대표적이다. 이 공식은 유연체의 변형이 선형범위내에

† Corresponding Author, sookim@cnu.ac.kr

© 2017 The Korean Society of Mechanical Engineers

있다고 가정하고, 유한요소로 나눈 절점 공식과 모델의 자유도를 효과적으로 줄여서 해석을 하는 변형모드해석을 통한 모달공식으로 대별된다.^(1~4)

변형을 기술하는 또 다른 방법으로는 관성좌표계에서 절점의 위치와 변형을 표시하는 절대 절점좌표계 공식이 있고, 이러한 공식은 대변위-미세응력 문제에도 적용된다.^(5,6)

하지만 이러한 유연 다물체 공식들은 매우 많은 변형자유도를 다루고 있어서 수치해석 방법을 적용하는데 있어서, 매우 큰 운동방정식과 고주파 변형 변수에 의한 매우 작은 적분 간격을 사용해야 하는 문제점이 있다. 결과적으로 유연체 해석에는 많은 시간이 소요될 수 밖에 없다.

효율적인 계산을 위한 방법으로는 컴퓨터의 병렬처리 연산 기능을 활용한 연구들이 진행되었다. 다물체 동역학 시스템의 병렬 해석 연구로는 대수방정식의 병렬 처리를 통하여 효율적으로 운동방정식의 해를 구하는 알고리즘이 개발 되었다.^(7,8) 특히 다물체 시스템의 효율적인 해석에서는 병렬처리에 적합한 효율적인 공식의 유도과 그에 적합한 병렬처리 방식의 개발이 필요하다. 이러한 원리에 따라서 다물체 시스템을 두개의 부분으로 연속적으로 분할해서 계산하는 divide and conquer 알고리즘에 대한 병렬처리 연구가 수행되었고,⁽⁹⁾ 유한요소 기반의 유연 다물체 시스템의 도메인 분할법에 의한 병렬처리 알고리즘⁽¹⁰⁾도 개발되었지만 이들 연구들은 모두 이론적인 효율 고찰만 되었을 뿐 실제 프로그램으로 구현하여 효율성을 검증한 사례는 없었다.

따라서 본 논문에서는 유연 다물체 시스템의 효율적인 해석을 위한 부분시스템 합성 공식을 기술하고, 부분시스템 합성방법의 병렬처리 적용과 부분시스템 내에서의 병렬계산을 위한 다층 구조의 병렬처리 방식을 제안하였다. 또한 OpenMP를 이용하여 실제 구현을 통한 병렬처리 방법의 효율성 고찰하고, 구현과정에서 나타난 주의점에 대한 가이드 라인을 제시하였다.

2. 직교좌표와 변형좌표계의 유연체 동역학 공식

Fig. 1은 유연체의 운동을 기술하기 위한 좌표시스템을 나타내고 있다. $X-Y-Z$ 는 관성 기준 좌표계이며, $x'-y'-z'$ 는 물체 기준 좌표계를 나타낸다.

유연체의 전체 운동은 관성 기준 좌표계에 대한 물체 기준 좌표계의 운동으로 나타낼 수 있다. 부유좌표계(Floating Reference Frame) 공식에서는 변

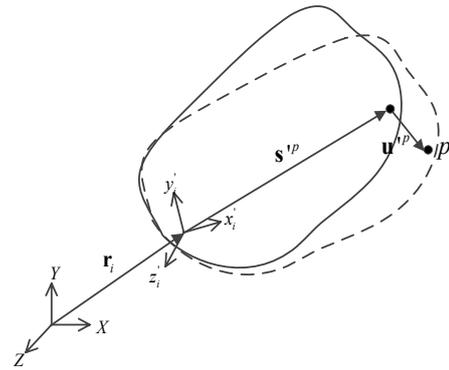


Fig. 1 Coordinate system in a flexible body

형은 물체 기준 좌표계에 대해서 상대적으로 정의된다. 따라서 일반적인 p점의 위치 벡터는 식 (1)과 같이 정의된다.

$$\begin{aligned} \mathbf{r}^p &= \mathbf{r}_i + \mathbf{A}_i(\mathbf{s}_i^p + \mathbf{u}_i^p) \\ &= \mathbf{r}_i + \mathbf{A}_i(\mathbf{s}'_i + \mathbf{N}^{ij} \mathbf{d}_j) \end{aligned} \quad (1)$$

여기서 \mathbf{r}_i 는 유연체의 전역 위치 벡터, \mathbf{A}_i 는 좌표계 변환 행렬, \mathbf{s}_i^p 는 물체 기준 좌표계에서 변형 전의 p점까지의 국부 위치 벡터를 나타낸다. \mathbf{u}_i^p 는 변형 전의 p점에서 변형 후의 p점까지의 국부 위치 벡터이며, 이는 유한요소 이론에 의해 절점 좌표계 \mathbf{d}_j 와 형상 함수 \mathbf{N}^{ij} 로 나타낼 수 있다.⁽¹⁾ 여기서, 아래 첨자 i는 i번째 물체를 의미하고, j는 j번째 절점을 의미한다.

유한요소 유연체 운동방정식은 다음과 같은 가정을 적용한다.⁽¹⁾

- (1) 변형률(strain)과 미소 변위는 선형관계이다.
- (2) 응력과 변형률은 전형적인 Hook's law에 의해 표현된다.
- (3) 유연체의 경계 조건은 운동 중에는 변하지 않는다.

가상 변위에 의한 가상 일의 유연체 운동방정식은 식 (2)와 같이 표현가능하다.⁽¹⁾

$$\begin{aligned} & -\int_{\Omega} \mu \delta \mathbf{r}^{pT} \dot{\mathbf{r}} d\Omega + \int_{\Omega} \mu \delta \mathbf{r}^{pT} \mathbf{f}^p d\Omega + \int_{\sigma} \mu \delta \mathbf{r}^{pT} \mathbf{T}^p d\sigma \\ & \equiv \int_{\Omega} \mu \delta \boldsymbol{\varepsilon}^{pT} \boldsymbol{\tau}^p d\Omega \end{aligned} \quad (2)$$

여기서, μ 는 물체의 밀도, \mathbf{f} 는 체적력, \mathbf{T} 는 표면력, $\boldsymbol{\varepsilon}$ 는 변형률, $\boldsymbol{\tau}$ 는 응력 텐서, Ω 는 체적, σ 는 표면을 의미한다.

식 (2)에 식 (1)의 유연체 기구학을 적용하여 정

리하면 식 (3)과 같이 표현할 수 있다.

$$\begin{bmatrix} \delta \mathbf{r}_i^T & \delta \theta_i^T & \delta \mathbf{d}_{ij}^T \end{bmatrix} \begin{pmatrix} \mathbf{M}_{ij} \begin{bmatrix} \ddot{\mathbf{r}}_i \\ \dot{\theta}_i \\ \ddot{\mathbf{d}}_{ij} \end{bmatrix} + \mathbf{S}_{ij} - \mathbf{Q}_{ij} + \mathbf{U}_{ij}(\mathbf{d}_{ij}) \end{pmatrix} = \mathbf{0} \quad (3)$$

여기서 \mathbf{M}_{ij} 는 유연체의 관성 행렬, \mathbf{S}_{ij} 는 자이로 효과 벡터, \mathbf{Q}_{ij} 는 일반적인 힘 벡터, \mathbf{U}_{ij} 는 일반적인 탄성력 벡터를 나타낸다. 그리고 θ_i 는 유연체 부유좌표의 전역 회전 값을 의미한다.

부유좌표계 공식을 사용하면, 물체 기준 좌표의 운동과 변형 운동은 유연체의 관성 행렬을 통해서 서로 연성이 되어 나타난다.⁽¹⁾

3. 병렬 처리를 위한 유연다물체 시스템의 부분 시스템 합성방법

3.1 유연체를 포함한 부분 시스템 운동방정식

부분 시스템 합성방법은 다수의 부분 시스템으로 이루어진 다물체 시스템을 부분 시스템 별로 운동방정식을 도출하고, 부분 시스템의 동역학적 효과를 기준 물체에 포함해서 독립적으로 운동방정식을 도출하는 방법으로 부분 시스템 별로 운동방정식의 해를 구할 수 있어 보다 효과적이며, 병렬 계산에 적합하다. 부분 시스템 합성방법을 설명하기 위해서 Fig. 2는 3개의 부분 시스템으로 이루어진 유연다물체 시스템의 도식적인 그림을 나타낸다. 각각의 부분 시스템은 다수의 유연 다물체 시스템으로 구성되어 있다.

위와 같은 시스템의 운동방정식은 일반적인 다물체 시스템 운동방정식 공식에 의해서 다음과 같이 직교좌표와 변형좌표에 의해 유도된다.⁽¹¹⁾

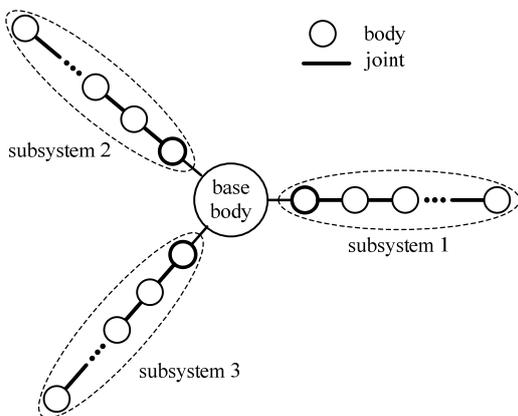


Fig. 2 A flexible multibody system with several subsystems

$$\begin{bmatrix} \mathbf{M}_0 & \mathbf{0} & \bar{\Phi}_{\mathbf{Y}_0}^T \\ \mathbf{0} & \bar{\mathbf{M}}_{sub} & \bar{\Phi}_{\mathbf{q}_{sub}}^T \\ \bar{\Phi}_{\mathbf{Y}_0} & \bar{\Phi}_{\mathbf{q}_{sub}}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{Y}}_0 \\ \ddot{\mathbf{q}}_{sub} \\ \bar{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_0 \\ \bar{\mathbf{Q}}_{sub} \\ \bar{\gamma} \end{bmatrix} \quad (4)$$

여기서, \mathbf{M}_0 는 기준 물체의 질량 관성 행렬, $\bar{\mathbf{M}}_{sub}$ 는 부분 시스템의 질량 관성 행렬, $\bar{\Phi}_{\mathbf{Y}_0}$, $\bar{\Phi}_{\mathbf{q}_{sub}}^T$ 는 각각 기준 물체와 부분 시스템의 구속조건 식에 대한 자코비언 행렬, \mathbf{Q}_0 , $\bar{\mathbf{Q}}_{sub}$ 힘 벡터, $\bar{\lambda}$ 는 라그랑지 승수이다. 또한 부분 시스템의 질량 행렬은 식 (5)와 같이 각각의 부분 시스템의 대각 행렬 구조로 구성된다.

$$\bar{\mathbf{M}}_{sub} = \begin{bmatrix} \mathbf{M}_{sub1} & & \\ & \mathbf{M}_{sub2} & \\ & & \mathbf{M}_{sub3} \end{bmatrix} \quad (5)$$

또한 자코비언 항은 기준 물체와 각각 부분 시스템 항으로 식 (6)과 같이 표현할 수 있다. 여기서 위 첨자는 i 번째 부분 시스템을 의미한다.

$$\bar{\Phi}_{\mathbf{Y}_0}^T = \begin{bmatrix} \Phi_{\mathbf{Y}_0}^{1T} & \Phi_{\mathbf{Y}_0}^{2T} & \Phi_{\mathbf{Y}_0}^{3T} \end{bmatrix}$$

$$\bar{\Phi}_{\mathbf{q}_{sub}}^T = \begin{bmatrix} \Phi_{\mathbf{q}_{sub1}}^{1T} & & \\ & \Phi_{\mathbf{q}_{sub2}}^{2T} & \\ & & \Phi_{\mathbf{q}_{sub3}}^{3T} \end{bmatrix} \quad (6)$$

이러한 시스템에 부분 시스템 합성방법을 적용하면 식 (7), (8)과 같은 3개의 부분 시스템 운동방정식과 각 부분 시스템의 라그랑지 승수 방정식으로 정리할 수 있다.⁽¹²⁾

$$\mathbf{M}_{sub i} \ddot{\mathbf{q}}_{sub i} = \mathbf{Q}_{sub i} - \Phi_{\mathbf{q}_{sub i}}^{iT} \lambda_{sub i}, \quad (i=1,2,3) \quad (7)$$

$$\lambda_{sub i} = \left(\Phi_{\mathbf{q}_{sub i}}^i \mathbf{M}_{sub i}^{-1} \Phi_{\mathbf{q}_{sub i}}^{iT} \right)^{-1} \left(\Phi_{\mathbf{Y}_0}^i \ddot{\mathbf{Y}}_0 + \Phi_{\mathbf{q}_{sub i}}^i \mathbf{M}_{sub i}^{-1} \mathbf{Q}_{sub i} - \gamma_{sub i} \right) \quad (8)$$

(i=1,2,3)

기준 물체의 운동방정식은 각 부분 시스템으로부터 효과 항을 합성하여 식 (9)와 같이 정의된다.

$$\left(\mathbf{M}_0 + \sum_{i=1}^3 \mathbf{M}_{sub i}^C \right) \ddot{\mathbf{Y}}_0 = \mathbf{Q}_0 + \sum_{i=1}^3 \mathbf{Q}_{sub i}^C \quad (9)$$

여기서, $\mathbf{M}_{sub i}^C$ 는 i 번째 부분 시스템의 효과 질량 행렬을 의미하고, $\mathbf{Q}_{sub i}^C$ 는 i 번째 부분 시스템의 효

과 힘 벡터를 의미하며, 식 (10)과 같이 표현할 수 있다.⁽¹²⁾

$$M_{sub\ i}^C = \Phi_{Y_0}^{i,T} \left(\Phi_{q_{sub\ i}}^i M_{sub\ i}^{-1} \Phi_{q_{sub\ i}}^{i,T} \right)^{-1} \Phi_{Y_0}^i, \quad (i=1,2,3)$$

$$Q_{q_{sub\ i}}^C = \Phi_{Y_0}^{2,T} \left(\Phi_{q_{sub\ i}}^i M_{sub\ i}^{-1} \Phi_{q_{sub\ i}}^{i,T} \right)^{-1} \left(\gamma_{sub\ i} - \Phi_{q_{sub\ i}}^i M_{sub\ i}^{-1} Q_{q_{sub\ i}} \right), \quad (i=1,2,3)$$

부분 시스템 합성방법은 식 (4)와 같이 전체 시스템의 운동방정식 대신, 식 (7~9)와 같이 여러 개의 부분 시스템 운동방정식과 기준 물체의 운동방정식으로 분할하여 계산할 수 있다. 따라서 식 (4)와 같이 표현되는 일반적인 방법과 비교하여, 계산 효율을 높일 수 있다. 또한 부분 시스템이 추가되는 경우에도 추가되는 부분 시스템으로부터 효과 질량 행렬과 힘 벡터를 기준 물체의 방정식에 추가하므로 부분 시스템 계산 모듈 프로그램 구조를 변경하지 않고, 용이하게 시스템을 확장할 수 있다.⁽¹³⁾

또한 식 (7), (8)은 독립적으로 계산이 가능하므로 병렬처리가 가능하다. 식 (10)에서의 역행렬 크기는 부분 시스템 내의 유연체 물체의 자유도에 따라 달라지지만, 역행렬의 구조는 대각 행렬 (block diagonal matrix)로 식 (11)과 같이 표현된다. 따라서 부분 시스템 별로 그리고 부분 시스템 내에서도 대각 행렬의 병렬 처리가 가능하다. 여기서 i 는 부분 시스템의 개수를 의미하고, j 는 각 부분 시스템의 유연체 개수를 의미한다.

$$M_{sub\ i}^{-1} = \begin{bmatrix} M_{sub\ i1}^{-1} & & & \\ & M_{sub\ i2}^{-1} & & \\ & & \ddots & \\ & & & M_{sub\ ij}^{-1} \end{bmatrix} \quad (11)$$

4. 병렬처리 구현

3장에서 기술한 유연 다물체 부분 시스템 합성방법에 의해서 기준 물체의 운동방정식을 계산한 후 부분 시스템 운동방정식은 독립적으로 해석을 할 수 있으므로 병렬 처리가 가능하다. 병렬 처리는 멀티코어 CPU의 PC에서 OpenMP API(Application Programming Interface)를 사용하여 구현되었다.

OpenMP directive를 사용하면 프로그램의 병렬처리 부분을 지정할 수 있는데, 병렬 처리로 계산할 부분의 독립성이 먼저 검토되어야 한다.

부분 시스템 합성방법을 이용하면, 식 (7), (8), (10)과 같이 각 부분 시스템 별 계산이 독립적으

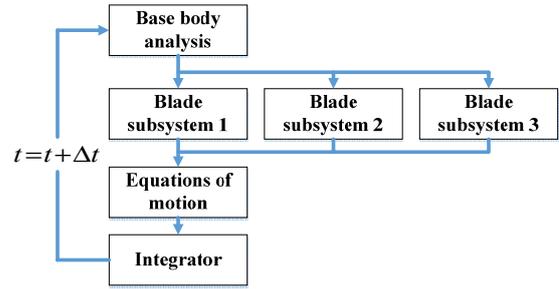


Fig. 3 Flow chart to identify independent tasks

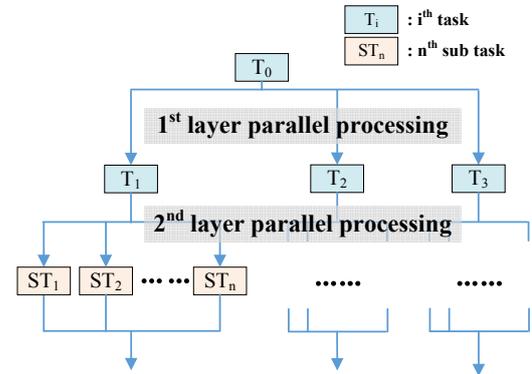


Fig. 4 Multilayered parallel tasks

로 나뉘지기 때문에 Fig. 3과 같이 표현할 수 있다.

나아가 부분 시스템 내부에서 식 (11)과 같은 대각 행렬 별로 독립적인 계산이 가능하므로 Fig. 4와 같은 역행렬 계산에서도 다중병렬처리 적용이 가능하다. 이때 본 논문에서는 Fig. 4에서와 같이 부분 시스템 별 병렬 계산 및 내부 병렬 계산을 각각 첫 번째 층(1st layer), 두 번째 층(2nd layer) 병렬 처리로 표현하였다.

병렬 처리의 구현은 많은 프로그램 경험을 필요로 한다. 다물체 시스템 해석은 일반적으로 Fig. 3과 같이 수치 적분을 위해서 반복적인 루프의 계산이 필요하다. 따라서 병렬스레드의 생성 시키는 위치가 중요하게 된다. 일반적으로 반복적인 루프가 없는 경우는 병렬 계산이 필요한 부분에서 병렬스레드를 생성 시킨다. 하지만 반복적인 루프가 있는 경우는 매 루프 안에서 병렬스레드의 생성과 소멸이 이루어지므로, 이것에 의해서 오히려 효율성이 떨어지게 된다.⁽¹⁴⁾ 따라서 병렬 프로세서를 생성 소멸의 시점도 OpenMP를 활용한 병렬 처리에 있어서 중요한 요소가 된다.

본 논문에서는 다중병렬처리를 위하여 병렬 프로세서 생성 시점에 따른 두가지 방법을 비교하였다. 첫번째 방법은 Fig. 5에 나타난 바와 같이 수치 적분 루프 밖에서 첫 번째 층(layer)과 두 번째

Table 1 Specification of blade model

Items	Value	Items	Value
L	2.0m	μ	2766.7kg/m ³
B	3.68 x 10 ⁻² m	E	68.9Mpa
H	2.99 x 10 ⁻³ m	ν	0.3
A	7.29 x 10 ⁻⁵ m ²	I	8.21x10 ⁻⁹ m ⁴

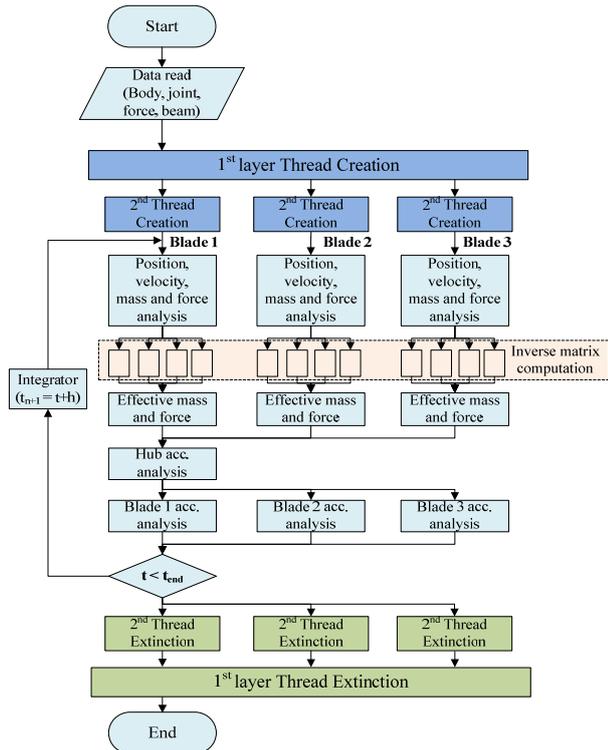


Fig. 5 Layered parallel processing scheme 1

층의 병렬프로세서를 생성 및 소멸 시키는 방법이다. 두번째 방법은 Fig. 6에 나타난 방법으로 루프 안에서 독립적인 계산에 따라서 첫 번째 층 두 번째 층의 병렬스레드를 반복적으로 생성 및 소멸 시키는 방법이다.

5. 유연체 회전 블레이드(blade)시뮬레이션

5.1 회전 블레이드 모델

부분 시스템 기반의 유연다물체 시스템의 병렬 처리를 위해서, 유연체 회전 블레이드 시뮬레이션을 수행하였다. 유연체 회전 블레이드 모델은 회전에 의한 강성화 효과를 고려하기 위해서 Fig. 7과 같이 강체 허브와 4개의 보 요소로 구성된 4개의 유연체를 고정 조인트로 연결하였다.⁽¹³⁾ 이러한 회전블레이드를 총 3개로 120도 간격으로 위치해 있다. 보 요소의 물성에 대한 자세한 사양은 Table

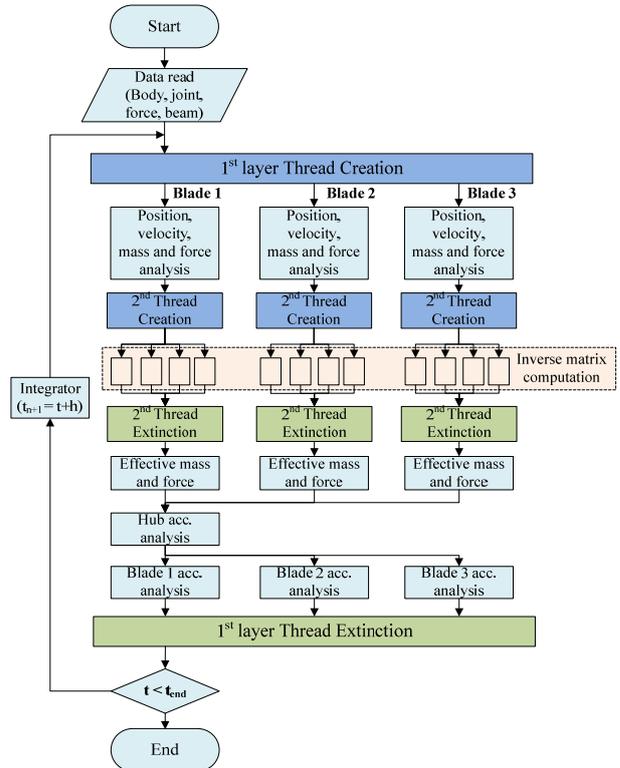


Fig. 6 Layered parallel processing scheme 2

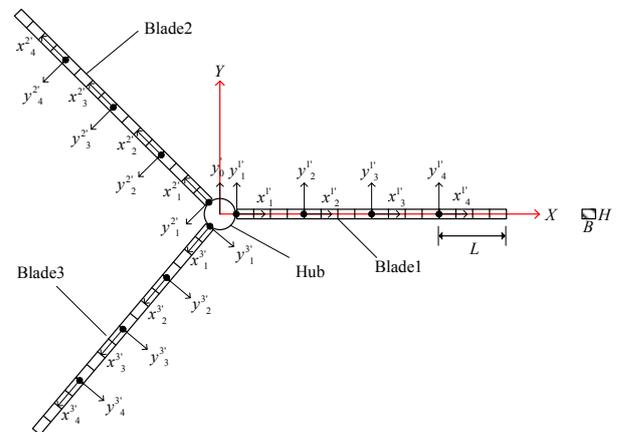


Fig. 7 Triple blade flexible model

1에 표기하였다.

여기서 L은 보 한 개의 길이, B는 보 단면의 밑면, H는 보 단면의 높이 A는 보 단면의 넓이, μ 는 밀도, E는 탄성계수, ν 는 포아송비, I는 단면 이차 모멘트를 각각 의미한다.

부분 시스템 합성방법으로 개발된 해석 모델은 Fig 6과 같은 해석 순서로 이루어지며, 해석 과정의 계산량을 살펴보면 Table 2와 같다. 여기서, 부분 시스템 별 효과 질량 행렬 및 힘 벡터의 계산량이 전체의 98.8%에 해당한다. 특히, 약 74.9%에

Table 2 Operation count

Function name	Count		Ratio (%)
Position analysis	$N \cdot \{3 \cdot 756 + 402\}$	8,010	0.10
Velocity analysis	$N \cdot 1,785$	5,355	0.07
Definition of mass, inertia and force	$N \cdot M \cdot E \cdot 1,022$	49,056	0.63
Effective mass and force	$N \cdot \{(3M)(3(E+1))\}^3 / 3$	5,832,000	74.86
	$2 \cdot \{N \cdot C_1 \cdot (3M)(3(E+1)) \cdot 143\}$	1,853,280	23.78
	$N \cdot C_1^3 / 3 + N \cdot C_1 \cdot 269$	11,412	0.15
Hub acceleration	$(3C_2)^3 / 3 + 3C_2 \cdot 6$	297	0.01
Blade acceleration	$N \cdot C_2 \cdot 72 + (3M)(3(E+1)) \cdot 167$	30,708	0.39
Total	-	7,790,118	100

여기서, N=3 (Num. of subsystem)
 M=4 (Num. of body at one subsystem)
 E=4 (Num. of beam element)
 C₁=12 (Num. of constraint of subsystem)
 C₂=3 (Num. of constraint of hub)

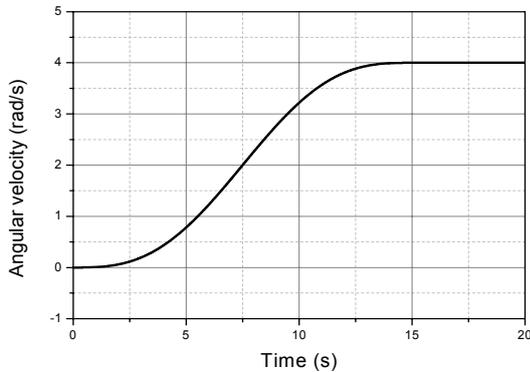


Fig. 8 Driver motion of the rotating blade

해당되는 효과 질량 및 힘 함수의 첫번째 항목이 식 (10)의 부분 시스템 질량 행렬의 역행렬(M_{sub}^{-1})을 계산하는 부분이다. 따라서 앞 절에서 언급된 바와 같은 병렬처리 기법들이 보다 효율적인 계산을 보장할 수 있다.

5.2 병렬처리 해석의 효율성 분석

유연체 회전 블레이드 모델에서 허브의 회전 조인트에 식 (12)와 같은 모션을 인가하였다. 해석 시간은 약 20초로 Fig. 8과 같은 각속도가 입력된다.

$$\dot{\phi} = \frac{\omega_s}{T_s} \left(t - \left(\frac{T_s}{2} \pi \right) \left(\sin \left(\frac{2\pi t}{T_s} \right) \right) \right), \quad 0 < t < T_s \quad (12)$$

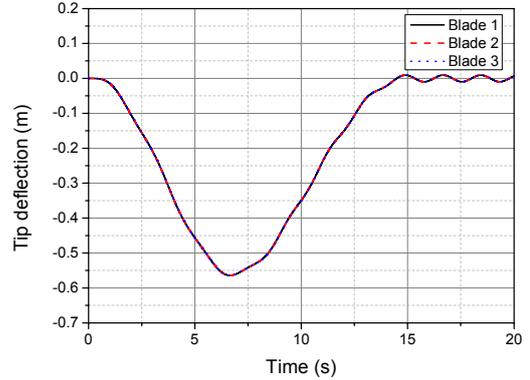


Fig. 9 Deformational displacement of end point

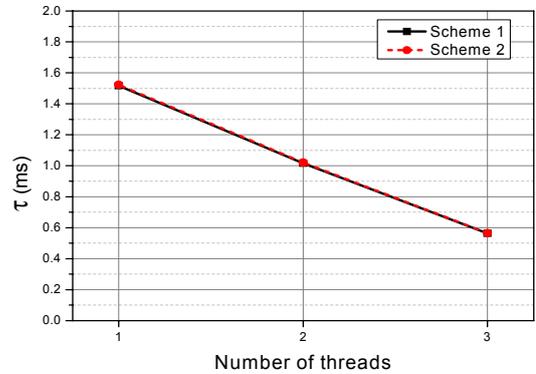


Fig. 10 CPU time using parallel processing of subsystem

Fig. 9는 회전 블레이드 끝 단의 상대적인 변형 변위를 나타내고 있다. 각 블레이드 부분 시스템의 끝 절점은 초기 가속 구간에서 변형이 점점 크게 증가하다가 감속 구간에서 줄어들며, 등속 구간에서는 작은 크기의 변형 값으로 진동하는 경향을 볼 수 있다.⁽¹⁵⁾

위 해석 모델에서 부분 시스템 병렬 처리를 우선 수행하였다. 이때 사용된 해석 PC의 사양은 Intel Xeon CPU E5-1660, 16GB RAM Windows 10, Visual Studio 2015이다.

총 3개의 부분 시스템으로 구성되어 있어서, 3개의 스레드를 할당시켰다. 해석방법으로는 Fig. 5, 6과 같이 두 가지 방법으로 해석하였다. 이때는 두 번째 층 스레드 생성 및 소멸은 포함되지 않는다. Fig. 10과 같이 스레드의 개수를 증가시킬수록 스레드 개수에 반비례 하여 해석 시간이 감소하는 경향을 보인다. 이러한 경향은 스레드 3인 경우에 Table 2의 N이 1이 되는 것과 같다. 또한 방법 1이 방법 2에 비하여 효율적인 것을 확인하였으나, 그 차이는 미비하였다. 이때 x축은 스레드 개수를 의미하고, y축은 스레드당 CPU 시간을 의미한다.

부분 시스템뿐만 아니라 역행렬 연산의 병렬 처리를 수행하는 다중병렬처리 알고리즘을 적용하였

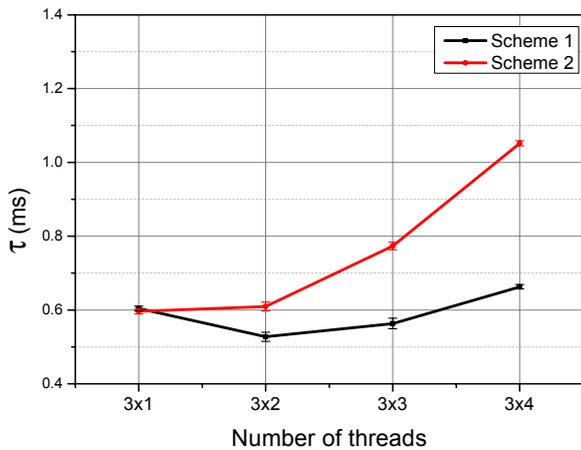


Fig. 11 layered parallel processing CPU time

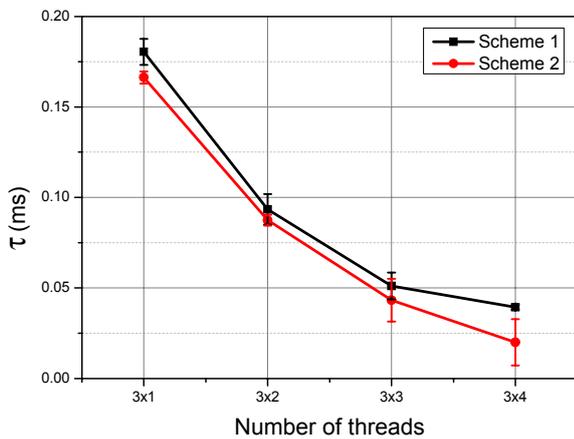


Fig. 12 layered parallel processing CPU time

다. Fig. 11은 각각의 다중병렬처리 알고리즘 사용 시 해석 시간을 나타내었다. 여기서 x축의 3x1은 부분 시스템 스레드 3개, 각 부분 시스템 스레드 내에서 역행렬 연산 스레드 1개를 의미하고, y축은 스텝당 CPU 시간을 의미한다. 해석결과는 총 20회를 해석 한 뒤 스텝당 CPU시간의 평균과 표준편차를 나타내었다.

위 결과를 살펴보면, 역행렬 연산 스레드가 2개 일 때 가장 효율이 좋은 것을 확인하였고, 3, 4개로 증가할수록 오히려 해석 시간이 증가하는 경향을 보였다. 또한 방법 1이 방법 2에 비해 더 효율적인 것을 확인할 수 있다. 방법 2의 경우에는 다중병렬처리의 스레드를 증가시킬수록 해석 시간이 증가하는 경향을 보이며, 4개를 사용시 1개를 사용할 때 보다 해석 시간 약 1.8배 증가하였다.

예상과는 다르게 다중병렬처리의 스레드가 증가할 수록 해석 시간이 증가되는 경향을 분석하기 위하여, 역행렬 연산 부분만 해석 시간을 추가적

Table 3 CPU time comparison

	Convention method	Subsystem method	Parallel processing	Layered parallel processing
τ (ms)	17.92	4.93	1.78	0.49
Ratio	36.57	10.06	3.64	1

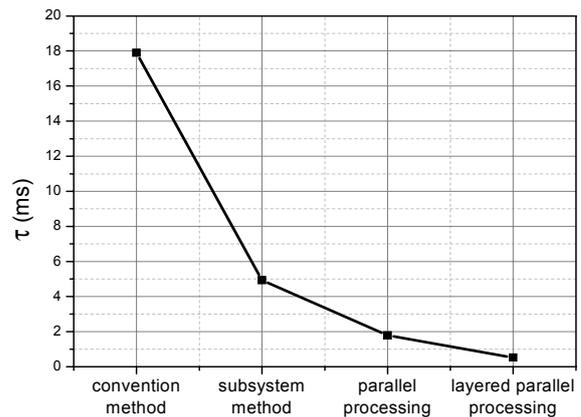


Fig. 13 Blade model CPU time measurement

으로 분석하였다. Fig. 12와 같이 역행렬 계산 시간은 역행렬 계산에 할당된 스레드가 증가할수록 감소하는 효과가 확인된다. 이러한 일관되지 못한 현상을 규명하기 위하여, C++의 프로파일러를 활용하여 계산에 사용된 함수에 대한 자세한 분석을 수행하였다. 실제 해석에 사용되는 스레드를 제외한 나머지 스레드가 부분 시스템 및 역행렬 계산시 공회전 모드(spin mode)로 전환되면서, 효율이 나빠짐을 알 수 있었다. 또한 방법 2는 공회전 모드 뿐만 아니라 적분 루프 안에서 두 번째 층의 스레드의 생성과 소멸에도 시간이 누적되어 Fig. 11과 같이 더욱 효율이 나빠지는 것을 알 수 있었다.

결과적으로, 다중병렬처리 알고리즘을 적용하기 위해서는 방법 1과 같은 반복 루프 밖에서 스레드를 생성 및 소멸 하는 것이 효과적이다. 다만 다중병렬처리의 스레드 생성시 공회전 모드를 고려한 스레드 개수 설정이 중요하다.

Fig. 13은 부유좌표계 공식의 직교 좌표 및 변형 절점좌표계 공식을 사용하는 일반적인 다물체 동역학 해석 방법과 부분 시스템 합성법을 적용한 해석 방법, 부분 시스템 합성법에 병렬처리 해석 방법을 적용한 방법 그리고 3x2 스레드를 사용한 다중병렬처리 해석의 방법 1을 적용하여 해석 시간을 비교하였다.

Table 3과 같이 부분 시스템 합성방법의 적용이 일반적인 해석 방법에 비해서 약 3.6배의 효율이

있었고, 또한 부분 시스템을 병렬처리 할 경우는 부분 시스템 합성 방법 보다 약 2.8배에 가까운 효율이 있었다. 다중병렬처리 해석이 부분 시스템 병렬처리 방식 보다 약 3.6배 빠른 것을 확인하였다. 여기서, Table 3의 ratio는 다중병렬처리 해석 시간을 1로 정규화 했을 경우에 각 방법의 해석 시간 비율을 의미한다.

6. 결 론

본 논문에서는 부분 시스템 기반의 유연다물체 동역학 시스템 공식과 그에 적합한 병렬처리 기법을 개발하였다. 부분 시스템 합성 방법에 따라 독립적인 계산을 규명하고, 그에 따라 다층구조의 병렬 처리에 따른 두 가지 방법을 제시하였다. 방법 1은 해석 루프 밖에서 미리 스레드를 생성하고 해석이 끝난 후 소멸하는 방식이고, 방법 2는 루프 안에서 반복적으로 생성 및 소멸 과정을 수행하는 방식이다.

일반적인 해석 방법에 비하여, 부분 시스템 합성방법을 적용하면 약 3.6배 효율이 좋아지는 것을 확인하였다. 또한 부분 시스템 합성방법을 적용한 공식에 병렬처리 기법을 적용하면 약 10배, 다중병렬처리 기법을 적용하면 약 36배 빨라지는 것을 확인하였다.

본 논문에서 제시된 스레드 생성 및 소멸에 따른 방법 1, 2는 부분 시스템만 적용한 경우에는 차이가 크게 않았고, 다중병렬처리를 적용한 경우에는 방법 1이 방법 2에 비해 약 1.8배 이상 효율적인 것을 확인하였다. 그리고 다중병렬처리 해석 시 두 번째 층의 스레드 개수에 따라서 해석 속도가 오히려 느려지는 경향을 확인하였다. 이는 OpenMP로 병렬 처리를 해석을 수행하지 않는 스레드의 공회전 모드(spin mode)가 효율성을 저해하는 것으로 판단된다. 따라서 다중병렬처리 해석을 계획할 때 두 번째 층의 스레드 수를 효과적으로 결정하는 것 중요하다. 이때 스레드의 수를 결정하는 방법으로 컴파일러에서 지원하는 프로파일러를 사용하여 스레드 할당 개수에 대한 각 함수에 사용되는 CPU time 할당량을 분석하고, 최소 시간을 가지는 스레드 개수를 찾아야 한다.

또한 추후, 스레드의 공회전 시간을 줄이기 위한 컴퓨터 공학 전문가와 협업 연구가 필요하다.

후 기

이 논문은 “충남대학교 재직교원 학술 연구비”

지원으로 수행한 연구과제입니다.

참고문헌 (References)

- (1) Wu, S.C., Haug, E. J. and Kim, S. S., 1989, "A Variational Approach to Dynamics of Flexible Multibody System," *Mech.Struct.&Mach.*, Vol. 17, No. 1, pp. 3~32.
- (2) Kim, S. S. and Haug, E. J., 1990, "Selection of Deformation Modes for Flexible Multibody Dynamics," *Mech.Struct.&Mach.*, Vol. 18, No. 4, pp. 565~586.
- (3) Spanos, J. T. and Tsuha, W. S., 1991, "Selection of Component Modes for Flexible Multibody Simulation," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, pp. 278~286.
- (4) Kim, S. S. and You, J. Y., 1998, "Gun System Vibration Analysis using Flexible Multibody Dynamics," *KSNVE*, Vol. 8, No. 1, pp. 203~211.
- (5) Shabana, A. A., 1997, "Definition of the Slopes and the Finite Element Absolute Nodal Coordinate Formulation," *Multibody System Dynamics*, Vol. 1, No. 3, pp. 339~348.
- (6) Shabana, A. A., 1998, "Computer Implementation of the Absolute Nodal Coordinate Formulation for Flexible Multibody Dynamics," *Nonlinear Dynamics*, Vol. 16, pp. 293~306.
- (7) Hwang, R. S., Bae, D. S., Kuhl, J. G. and Haug, E. J., 1990, "Parallel Processing for Real-time Dynamic System Simulation," *Journal of Mechanical Design*, Vol. 112, No. 4, 520-528.
- (8) Gonzalez, F., Luaces, A., Lugin, U. and Gonzalez, M., 2009, "Non-intrusive Parallelization of Multibody System Dynamic Simulations," *Computational Mechanics, Springer Verlag*, Vol. 44, 493~504.
- (9) Critchley, J. H. and Anderson, K. S., 2004, "A Parallel Logarithmic Order Algorithm for General Multibody System Dynamics," *Multibody System Dynamics*, Vol. 12, No. 1, pp. 75~93.
- (10) Bauchau, O. A., 2010, "Parallel Computation Approaches for Flexible Multibody Dynamics Simulations," *Journal of the Franklin Institute*, Vol. 347, No. 1, pp. 53~68.
- (11) Shabana, A. A., 2005, *Dynamics of Multibody Systems*, Cambridge, New York, USA.
- (12) Yun, H. S., Kim, S. S., Lee, C. H. and Kim, H. W., 2015, "A Study on the Efficient Flexible Multibody Dynamics Modeling of Deep Seabed Integrated Mining System with Subsystem Synthesis Method," *KSME-A*, Vol. 39, No. 12, pp. 1213~1220.
- (13) Kim, S. S. and Han, J. G., 1999, "A Study on Subsystem Synthesis Method for Vehicle System," *KSME-A*, Vol. 23, No. 3, pp. 520~534.
- (14) Kang, H. C. and Kim, S. S., 2015, "Parallel

- Processing with the Subsystem Synthesis Method for Efficient Vehicle Analysis," *JMST*, Vol. 29, No. 7, pp. 2663~2669.
- (15) Wu, S.C. and Haug, E. J, 1988, "Geometric Non-linear Substructuring for Dynamics of Flexible Mechanical Systems," *Int. J. Numer. Meth. Eng.*, Vol 26, pp. 2211~2226.