

# 객체지향 개념을 이용한 다물체 동역학 해석 시스템 개발

한형석<sup>#</sup>, 이재경<sup>\*</sup>, 서종휘<sup>\*\*</sup>, 송현석<sup>\*\*</sup>, 박태원<sup>\*\*\*</sup>

## Development of a Multi-body Dynamics Analysis System Using the Object-Oriented Concept

Hyung Suk Han<sup>#</sup>, Jai Kyung Lee<sup>\*</sup>, Jong Whi Seo<sup>\*\*</sup>, Hyun Seok Song<sup>\*\*</sup> and Tae Won Park<sup>\*\*\*</sup>

### ABSTRACT

To analyze the applications of all types of mechanical systems, general purpose analysis programs have been developed and commercialized. However, it is customary to develop and use customized programs even though they sometimes require more work than a general purpose program. A customized program is simplified to adapt to a particular application from the beginning, is designed for small computers, and developed with hardware-in-the-loop in mind so it can be applied effectively. By adding design knowledge and bundling know-how to an analysis program, analysis time can be reduced. And because an analysis has to work in conjunction with other analysis programs, a proprietary program that the user can easily modify can be useful.

In this thesis, a multi-body dynamics analysis system is presented using one of the most useful programming techniques, object-oriented concept. The object-oriented concept defines a problem from the physical world as an abstract object, an abstract model. The object becomes encapsulated with the data and method. Simulation is performed using the object's interface. It is then possible for the user and the developer to modify and upgrade the program without having particular knowledge of the analysis program. The method presented in this thesis has the following advantages. Since the mechanical components of the multi-body system converts independent modeling into a class, the modification, exchange, distribution, and reuse of elements are increased. It becomes easier to employ a new analysis method and interface with other S/W and H/W systems. To employ a new analysis method, there is no need to modify elements of the main solver and the library. In addition, information can be communicated to each object through messaging. It makes the modeling of new elements easier using inheritance. When developing a S/W for the computer simulation of physical system, it is reasonable to use object-oriented modeling. Also, for multi-body dynamics analysis, it is possible to develop a solver that is user-oriented.

**Key Words** : Multi-body Dynamics (다물체동역학), Object-oriented Modeling (객체지향모델링)

접수일: 2003 년 1 월 2 일; 게재승인일: 2003 년 7 월 11 일

# 교신저자: 한국기계연구원

Email: hshan@kimm.re.kr, Tel: (042) 868-7814

\* 한국기계연구원

\*\* 아주대학교 대학원

\*\*\* 아주대학교

## 1. 서론

다물체 동역학 모델링 분야의 연구는 지난 약 30 년간에 걸쳐 많은 발전을 이루었으며 그 결과로 관련 해석 프로그램들이 상용화되었다. 그러한 해석 프로그램들은 기계계 전문분야의 응용 문제를 해석 프로그램의 수정 없이 모델링 하는 것에 목적을 두고 개발되었다. 그 예로 ADAMS<sup>1</sup>, DADS<sup>2</sup>, SIMPACK<sup>3</sup> 등이 있으며 상대 좌표계를 이용하며 국내에서 개발된 RecurDyn<sup>4</sup> 이 있다. 그러나 이러한 상용 프로그램이 있지만, 많은 경우에 있어서 여전히 상용화된 범용 프로그램을 사용하는 것보다 작업량은 많아 보이지만 자체 프로그램을 개발하여 사용하는 것이 통례이다. 이러한 역설적인 행동에는 여러 이유가 있다. 첫째로는 범용인 상용 프로그램들은 특정한 요구 조건에 맞도록 일반 사용자가 수정하는 것이 어렵다는 것이다. 둘째로는 자체 개발한 프로그램은 개발 초기부터 특정한 목적에 맞도록 단순화하고 사용자 컴퓨터 환경이나 하드웨어와의 연결 등을 고려하여 개발되기 때문에 효율적으로 적용될 수 있다. 셋째로는 특정한 대상에 대한 노하우를 부가시킬 수 있다. 설계 지식 및 노하우를 해석 프로그램에 부가 시킴으로써 해석 결과의 정도를 높이고 평가 업무에 필요한 시간도 줄일 수 있게 된다. 넷째로는 다물체 동역학 해석이 시스템 공학적인 측면에서 한 부분을 차지할 수 있는데, 이는 다른 해석 프로그램과 긴밀한 연관이 이루어져야 하므로, 이를 위해서는 사용자가 쉽게 수정할 수 있는 자체 프로그램이 유용하다. 이러한 대안으로 기호계산을 이용한 다물체 동역학 해석 프로그램을 생성하는 기법이 개발되었으나 사용하기 어려워 널리 이용되지는 못하고 있다. 이상의 이유로 여전히 상용 프로그램과 별도로 특정한 목적을 위한 해석 프로그램의 개발이 요구될 때가 있다. 국내에서는 객체지향 개념을 이용한 다물체 동역학 해석 프로그램 개발에 대한 공개 사례는 없다. 다만 건축분야, 조선분야, 구조해석 분야에서는 적용된 사례가 있다.<sup>5-7</sup> 자동차 분야에서는 주로 MATLAB 에 기반을 둔 제어 시스템 시뮬레이션에 적용된 사례가 있다.<sup>8-10</sup> 국외의 경우 객체지향 언어를 이용한 다물체 시스템에 대한 모델링이 수행된바 있으나 운동방정식이 일반적인 형식이 아니기 때문에 적용하기 위해서는 관련 지식에 대한 깊은 이해가 필요하다.<sup>11-13</sup>

본 논문에서는 프로그래밍 기법에서 현재 가장 유용한 방법의 하나로 알려진 객체지향 개념을 도입하여 사용자 중심적인 다물체 동역학 해석 시스템 개발 방안을 제시한다. 객체지향 개념은 실세계의 문제를 추상적인 객체(object)라는 모델로 정의한다. 객체는 데이터와 메소드(기능)로 캡슐화되며 이러한 객체들간의 인터페이스를 이용하여 시뮬레이션을 수행하게 된다. 그러면 사용자 및 개발자는 해석 시스템에 대한 상세한 지식이 없이도 원하는 요소에 대한 해석 모델의 수정 및 보완이 쉽게 가능하게 된다. 본 논문에서는 이미 보편화된 다물체 동역학 방정식을 그대로 이용하면서 객체지향 개념을 도입하기 때문에 사용자 및 개발자가 쉽게 해석 시스템을 수정하거나 재 사용할 수 있는 방법을 제시한다. 또한 제시한 방법으로 객체지향 언어 C++를 이용하여 프로그램을 개발하였다. 그 결과 본 논문에서 제시하는 해석기는 수정이 쉽고, 재 사용성이 높으며, 확장성, 유연성의 장점을 가지기 때문에 사용자 중심적인 다물체 동역학 해석 시스템으로 기대된다.

## 2. 객체지향 모델링 개요

소프트웨어 공학에서 발전한 객체지향 모델링 및 프로그래밍 기법은 소프트웨어의 개발 방법으로 이미 실용화되었다. 기존의 절차적 모델링 기법은 문제를 분석하고 프로그램을 구현하는 과정이 기능(함수) 지향적이다. 절차적 모델링 기법에서는 데이터가 부수적인 요소로 취급된다. 반면에 객체지향 기법에서는 실세계의 사물을 소프트웨어 영역으로 사상시킨 객체를 중심으로 구현된다. Fig. 1 에서와 같이 전통적인 절차적 모델링 기법에서는 데이터와 절차가 구분된 후 절차의 필요에 따라 데이터를 사용한다. 반면에 객체 지향에서는 데이터와 기능을 캡슐화시킨 객체를 중심으로 원하는 기능을 수행하게 된다.

Fig. 2 는 객체화의 개념을 나타내는 그림으로 객체는 데이터와 기능을 갖는 클래스로 추상화 된다. 이러한 모델링은 실세계의 문제 표현에 가깝고 자연스러운 것으로 알려져 있다. 객체지향 모델의 기본 개념에 이용되는 것은 객체, 클래스, 멤버(데이터), 멤버함수(기능), 메시지이다. 객체는 독립적이면서 타 객체와는 메시지를 통하여 인터페이스를 하게 된다.

많은 객체들은 상호 동일하거나 유사한 데이터 구조와 기능을 가지는데 같은 데이터 구조와 기능을 갖는 객체들의 집합을 클래스라고 한다. 결과적으로 소프트웨어 모듈의 부품화를 가능하게 하고 개발 효율 및 신뢰성을 높여준다. 또한 유지 보수, 확장성, 재 사용성이 탁월하다. 클래스는 상속성이 있어 상위 클래스를 이용하여 하위 클래스를 쉽게 파생시킬 수 있다. 또한 다형성이 있어서 한 함수가 상황에 따라 다른 기능을 수행한다. 결과적으로 현실 세계의 사물을 소프트웨어적으로 시뮬레이션하는 프로그램의 개발에 있어서 객체지향 기법을 이용하는 것이 타당하다고 할 수 있다. 객체지향 프로그래밍 언어 C++는 객체지향의 개념을 구현하도록 개발된 것이며 C의 문법을 포함하고 있어 기존의 C로 만들어진 자원을 이용할 수 있는 장점을 가진다.

### 3. 다물체계 모델링

#### 3.1 요소의 객체화

Fig. 3 은 다물체 시스템으로 물체, 구속조건, 조인트, 힘요소 및 제어 요소들로 구성된다. 조인트에는 회전 조인트, 병진 조인트, 구 조인트 등과 같은 종류가 있으며, 힘 요소에는 병진 스프링, 회전 스프링과 같은 종류가 있다. 이러한 요소들은 객체지향 개념을 도입하면 Fig. 4 와 같이 객체로 모델 된다. 객체의 특징은 독립적이라는 것이다. 객체는 객체의 속성과 기능을 포함하고 있다. 그리고 객체간의 인터페이스는 메시지를 통하여 이루어진다.

#### 3.2 클래스화

본 연구에서 제시하는 객체지향 모델링 개념을 이용한 다물체계의 동적 모델링의 핵심은 클래스의 도입이다. 클래스는 동종의 객체를 정의하기 위한 템플릿이라고 할 수 있다. 전기와 같이 클래스는 속성, 즉 데이터와 기능을 동시에 갖고 있다. Fig. 5 는 다물체계를 구성하는 객체들을 생성하기 위한 최상위 클래스를 제시하고 있다. 최상위 클래스는 모든 클래스의 공통 속성과 기능을 포함하게 된다. 속성으로는 객체명, 클래스 형식, 자유도 수, 구속조건 수, 객체번호, 인스턴스 번호, 자유도 번호, 구속조건 번호 등이다. 기능으로는 초기화, 데이터 읽기, 결과 출력으로 구성된다.

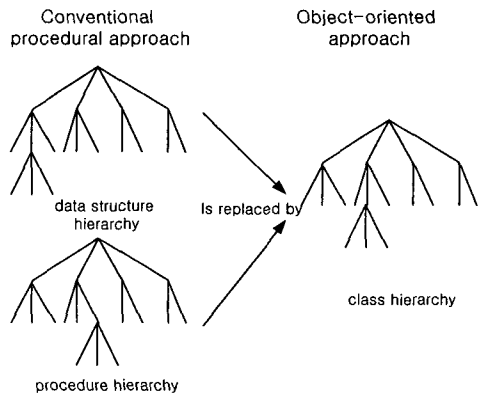


Fig. 1 An object-oriented approach has one unified hierarchy

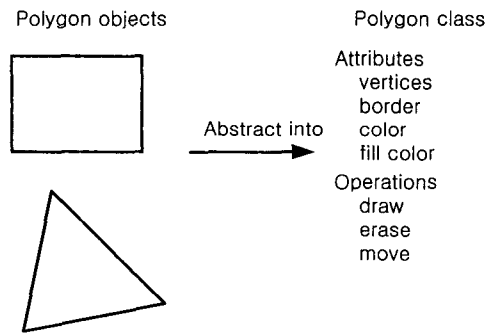


Fig. 2 Objects and classes

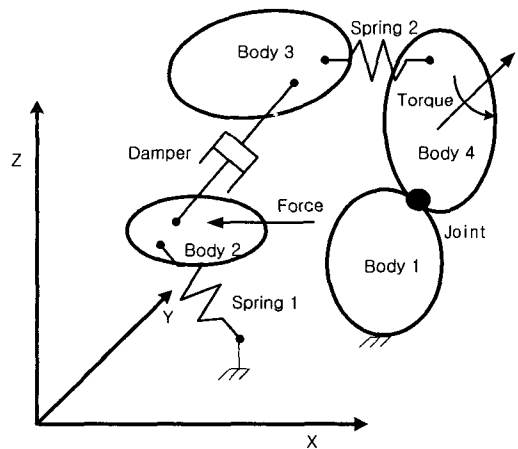


Fig. 3 Multi-body system

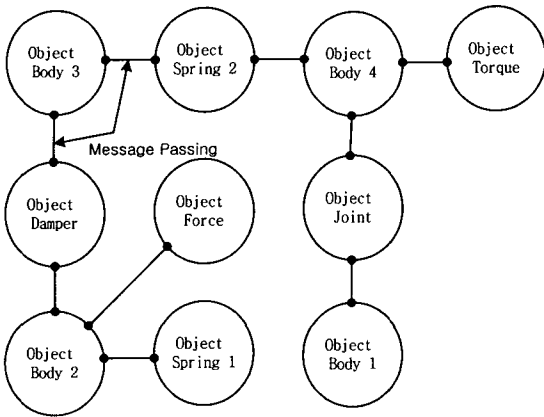


Fig. 4 Object-oriented representation of multi-body system

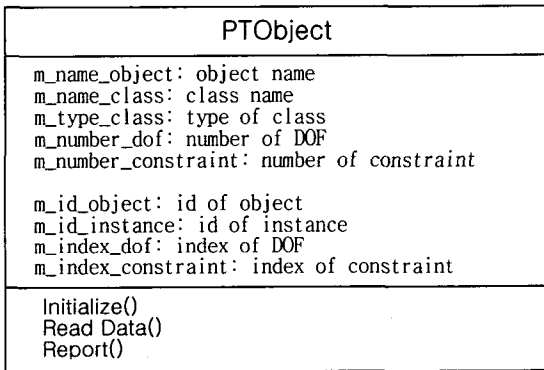


Fig. 5 Superclass diagram for multi-body system components

Fig. 6 은 Fig. 5 의 최상위 클래스를 이용하여 파생시킨 클래스 계층도이다. 최상위 클래스는 PObject 이며 PObject 를 이용하여 Constraint, Body, Force 와 같은 부 클래스를 파생시켰다. 이 부 클래스를 이용하여 물체(Body), 조인트류, 구속 조건류, 힘류를 파생시키게 된다. 클래스의 파생에는 객체지향의 특징인 상속성, 다형성이 이용되어 쉽게 새로운 클래스를 정의할 수 있다. 또한 각 클래스는 데이터가 독립적으로 관리되기 때문에 클래스 개발시 독립적으로 개발할 수 있다. Fig. 7- Fig. 9 는 PObject 클래스에서 파생된 Body, Constraint, Force 형태의 클래스를 보여주고 있다. 이러한 클래스의 주요 데이터와 기능은 3.3 에서 설명되는 식(1)-(6)과 관련되어 있다. 예를 들어

Body 클래스의 경우 위치, 속도, 가속도, 작용하는 힘, 질량행렬, 구속식의 자코비안, 속도 방정식 우변 등의 데이터를 가지고 있다. 이러한 기본적인 클래스를 이용하여 새로운 클래스를 파생시키게 된다. 이 때 기본 클래스에서 제공하지 않은 데이터나 기능을 추가하여 클래스를 정의하게 된다.

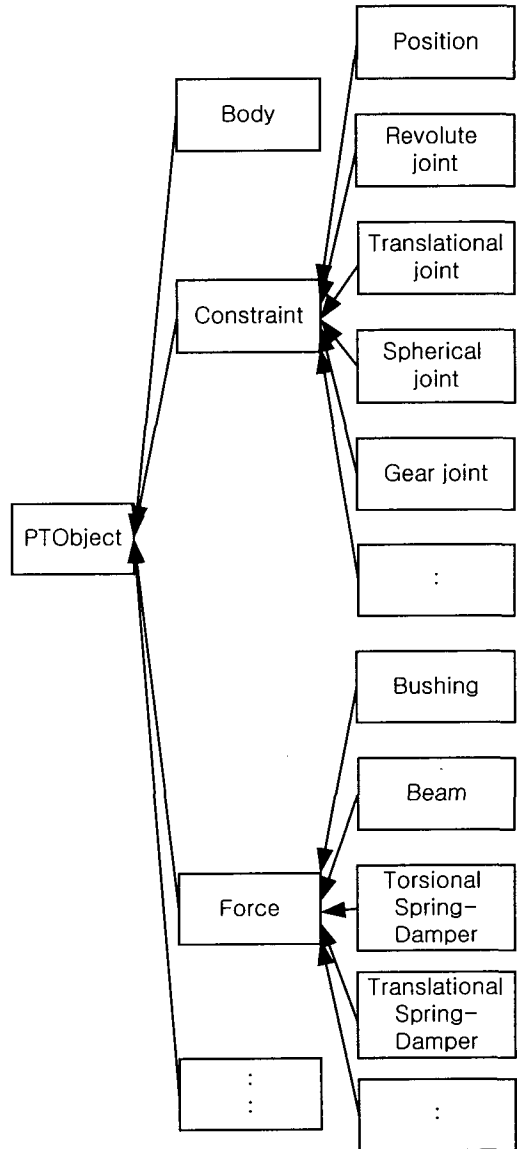


Fig. 6 Multilevel inheritance hierarchy

Body
m_position[7] m_velocity[7] m_acceleration[7] m_force[7] m_mass[7][7] m_jacobian[1][7] m_RHS_velocity[1] m_RHS_acceleration[1] m_constraint[1] m_velocity[1]
Calculate mass() Calculate jacobian() Update force() Calculate RHS_velocity() Calculate RHS_acceleration() Calculate constraint() Calculate velocity() Report Body()

Fig. 7 Class Body

Constraint
m_name_body_i m_name_body_j m_triad_i[3][3] m_triad_j[3][3] m_jacobian[6][7] m_RHS_velocity[6] m_RHS_acceleration[6] m_constraint[6] m_velocity[6] m_LagrangeMultiplier[6]
Calculate RHS_velocity() Calculate RHS_acceleration() Calculate constraint() Calculate velocity() Calculate reaction_force() Report Constraint()

Fig. 8 Class Constraint

Force
m_name_body_i m_name_body_j m_triad_i[3][3] m_triad_j[3][3] m_force[7]
Calculate force() Calculate apply_force() Report Force()

Fig. 9 Class Force

### 3.3 수치해석

Fig. 3 의 다물체계가 nb 개의 강체로 구성되어 있다면 공간에서 시스템의 운동을 표현하기 위해

서는  $6 \times nb$  개의 좌표가 필요하다. 이러한 일반좌표는 인접하는 물체들간의 조인트 때문에 모두 독립은 아니다. 각 물체들의 운동은 일반좌표와 일반좌표 속도 연관성을 정의하는 기구적 구속에 의한 영향을 받는다. 다물체 시스템의 운동을 제어하고 이해하기 위해서 공간에서의 물체, 조인트, 힘 요소의 정의가 필요하게 된다. nb 개의 물체로 구성되는 시스템이 있다면 그 시스템의 일반좌표는 식(1)과 같이 정의된다.

$$\mathbf{q} = [q_1, q_2, \dots, q_{nb \times 6}]^T \quad (1)$$

만일 시스템에  $m$  개의 구속조건이 있다면 식(2)와 같이 표시되고 구속된 기계 시스템의 Lagrange 승수 형태의 운동방정식은 식(3)과 같이 정의된다.<sup>14</sup>

$$\Phi(\mathbf{q}, t) = [\Phi_1(\mathbf{q}, t), \dots, \Phi_m(\mathbf{q}, t)]^T = \mathbf{0} \quad (2)$$

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \boldsymbol{\lambda} = \mathbf{Q} \quad (3)$$

구속식(1)을 만족하는  $\dot{\mathbf{q}}, \ddot{\mathbf{q}}$  를 수치적으로 얻기 위하여 식(1)의 1, 2 차 편미분을 다음식과 같이 구한다.

$$\Phi_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t = \mathbf{0} \quad (4)$$

$$\Phi_{\mathbf{q}} \ddot{\mathbf{q}} = -(\Phi_{\mathbf{q}\dot{\mathbf{q}}})_{\mathbf{q}} \dot{\mathbf{q}} - 2\Phi_{\mathbf{q}\dot{\mathbf{q}}\dot{\mathbf{q}}} \dot{\mathbf{q}} - \Phi_{tt} \equiv \boldsymbol{\gamma} \quad (5)$$

식 (3)과 (5)를 이용하여 행렬형태의 시스템 운동방정식을 식(6)과 같이 정의하게 된다.

$$\begin{bmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \mathbf{Q} \\ \boldsymbol{\gamma} \end{Bmatrix} \quad (6)$$

여기서,

$\mathbf{q}$ : position vector

$\dot{\mathbf{q}}$ : velocity vector

$\ddot{\mathbf{q}}$ : acceleration vector

$\mathbf{M}$ : mass matrix

$\Phi_{\mathbf{q}} \equiv [\partial \Phi_j / \partial q_i]_{m \times n}$ : constraint Jacobian matrix

$\boldsymbol{\lambda}$ : Lagrange multiplier

$\mathbf{Q}$ : generalized force

$\boldsymbol{\gamma}$ : right side of constraint acceleration

이러한 다물체 동역학 시스템 운동방정식의 해법은 Haug<sup>14</sup>의 문헌을 참조 바란다. 이러한 운동방정식에 기초를 둔 상용 프로그램들은 모두 절차적 프로그래밍 기법에 기초를 두고 있다. 다물체 시스템의 동적 해를 구하기 위해서는 식(6)의 수치해석이 요구된다. 동시에 식 (2)와 (3)에 대한 함수 계산도 요구된다. 본 논문에서는 다물체계의 운동방정식 구성 및 수치해석도 클래스를 통하여 이루어진다. 본 논문에서는 식(6)의 구성 및 수치해석을 위하여 별도의 클래스를 정의하였고 동적 해석은 Fig. 10 과 같은 순서로 진행된다. 수치해석을 수행하는 클래스는 시스템 운동방정식을 구성하기 위한 벡터, 행렬, 수치해석 함수들을 가지고 있게 된다.

여기서 본 논문에서 소개하는 객체지향의 장점이 수치해석 과정에서 잘 나타난다. 동적 해석을 위해서는 식(2)~(6)의 구성이 필요하고 Fig. 10 과 같은 흐름으로 해석을 수행하는데, 본 논문에서 소개하는 방법에서는 공통 메모리를 사용하지 않는다는 것이 가장 큰 특징이며 장점이다. 모든 데이터는 지역적 변수로 정의된다. 이를 좀더 상세히 설명하기 위하여 Fig. 11 을 예로 든다. Fig. 11 에서와 같이 물체의 질량행렬을 구성함에 있어 기존의 한 방법은 Fig. 11(a)와 같이 공통 배열을 이용한다. 각 객체(예: body i)는 질량행렬을 계산하기 위하여 body i 에 관련된 데이터를 공통 배열에서 읽어와 질량을 계산하여 공통 배열에 갱신한다. 갱신된 데이터는 시스템 질량행렬을 구성하는데 이용된다. 반면에 본 논문에서 이용된 객체지향 데이터 모델에서는 각 물체가 각 물체에 관련된 데이터를 포함하고 있으며 각 객체가 질량행렬을 갱신하여 가지고 있다. 시스템 질량행렬을 구성하기 위해서는 각 물체에 각 물체의 질량행렬을 조회하여 수행된다. 이와 같이 공통 메모리를 사용하지 않기 때문에 프로그램의 전체적인 구조에 대한 지식이 없이도 새로운 클래스를 정의할 수 있게 된다. 실제 개발과정에서 새로운 클래스 개발자가 프로그램의 전체 구조에 대한 지식이 없이도 단시간 내에 클래스를 정의하고 수치해석 객체와의 통합을 수행하는 것을 경험하였다.

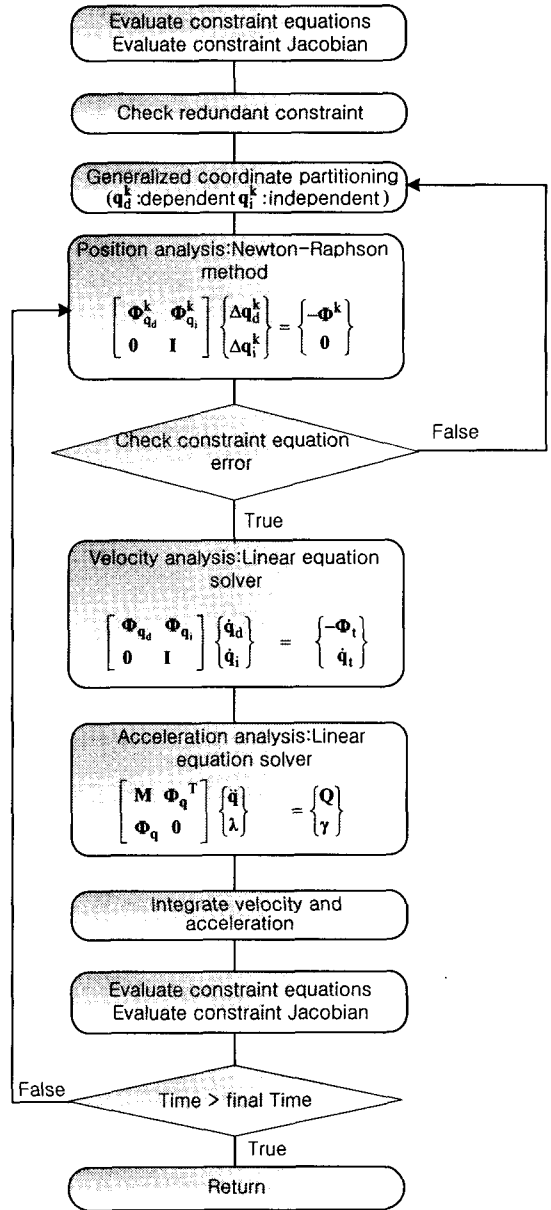


Fig. 10 Dynamic analysis flow

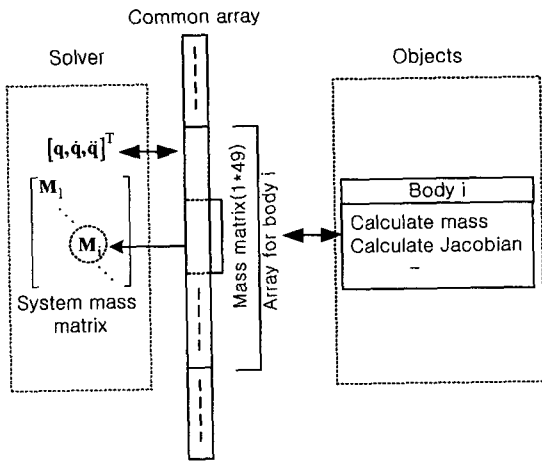


Fig. 11(a) Conventional data model for multi-body dynamic analysis

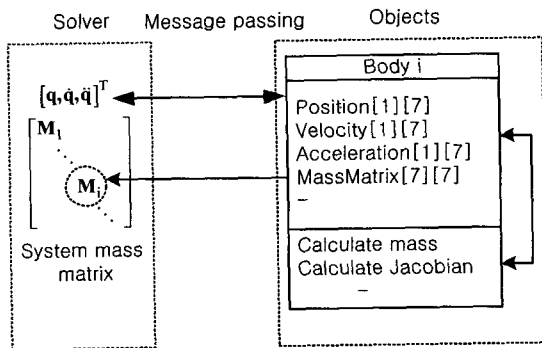


Fig. 11(b) Object-oriented data model for multi-body dynamics analysis

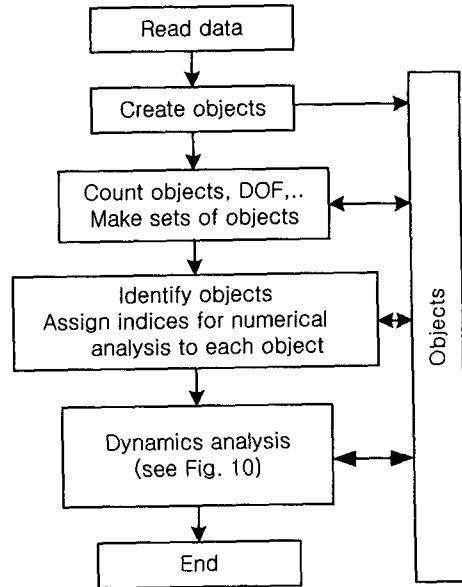


Fig. 12 O-DYN dynamics analysis flow

Table 1 O-DYN classes

Class type	Classes
Body	Rigid Body
Constraint	Ground, Position, Point, Revolute joint, Cylindrical joint, Translational joint, Bracket joint, Spherical joint, Rack-and-pinion, Gear joint, Driver
Force	Translational-spring-damper, Rotational-spring-damper, Beam, Bushing

### 3.4 해석 프로그램 개발

전 절의 객체지향 모델링 기법을 이용하여 다물체계 구성 요소들을 클래스화 하고 운동방정식의 구성 및 수치해법에 관련된 함수들을 C++언어로 개발하였다. 본 논문에서 개발한 다물체 동역학 해석기(O-DYN/Solver)의 흐름도는 Fig. 12와 같다. 현재 O-DYN은 Table 1과 같이 다물체 동역학 해석을 위한 클래스가 개발되었다. 해석기를 실행시키기 위한 전후처리는 웹기반 시스템으로 개발되었는데 이에 대하여는 다른 논문에서 상세하게 소개한다.

### 4. 적용

본 논문에서 제시한 객체지향 모델링 기법을 이용하여 개발한 다물체 동역학 해석 시스템 O-DYN을 이용하여 복동식 압축기의 동적 해석을 수행하였다. Fig. 13과 같은 복동식 압축기는 냉장고를 비롯한 냉각 장치에 많이 이용된다. 압축기의 고급화 경향에 따라 압축기의 진동 저감은 중요한 과제 중의 하나이다. 압축기의 진동 원인은 실린더의 압력 변화와 그에 따른 모터 토크의 변동에 주로 기인한다. 진동 저감을 달성하기 위해

서는 냉매 압력과 모터 성능이 주어진 조건에서, 동적 거동과 그에 따른 진동을 야기하는 가진력 예측이 요구된다. 본 논문에서는 가진력을 예측하기 위하여 본체를 구속조건으로 고정하고 구속 조건에서의 반력을 가진력으로 평가한다. Fig. 14 는 압축기의 개략적인 형상과 입출력의 기준이 되는 좌표계를 보여주고 있다. Fig. 13의 압축기에 대한 관성 특성과 다물체 동역학적 모델을 Table 2 와 Table 3 에서 각각 보여주고 있다. 압축기 모터의 토크 성능은 실험에 의하여 구해졌으며 Fig. 15 와 같다. 모터 토크는 회전자의 속도와 그에 따른 출력 토크를 Fig. 15 의 성능 곡선을 이용하여 정의 하게 된다. 실린더 내에서의 압력은 회전자의 회전 각의 함수로 Fig. 16 과 같이 실험에 의하여 얻 어진 특성 곡선을 이용한다. 압축기는 부품간의 접촉부에 회전 저항이 존재한다. 본 논문에서는 전체적인 거동에 중점을 두기 때문에 실험 결과를 기초로 하여 단지 회전자와 고정된 블록(block)의 상대 회전 속도에 따른 감쇠로 정의하였다. 실험 결과에서의 모터 회전 속도와 회전 속도의 변동을 해석 모델과 비교하여 회전 저항으로 감쇠계수 2.5Nms/mm 를 택하였다. 해석 결과의 검증을 위하여 이미 상용화된 다물체 동역학 해석 프로그램 DADS 을 이용하여 동일한 동적 해석을 수행하고 상호 비교하였다. 다음은 이와 같은 동적 모델에 대하여 본 논문에서 소개하는 O-DYN 과 DADS 의 해석 결과를 보여준다.

Fig. 17~19 는 피스톤의 x 방향 위치, 속도, 가속 도를 보여주고 있다. 우선 O-DYN 과 DADS 의 해 석 결과 차이는 미미한 것을 볼 수 있다. 이러한 해의 일치성을 통하여 본 논문에서 소개하는 객체 지향 모델 중심의 동적 해석 프로그램의 알고리즘 과 해의 신뢰성이 간접적으로 검증이 된다. 압축 기는 주기적인 운동을 하며 가속도는 Fig. 19 에서 와 같이 회전자가 0°, 180°일때 1.03mm/ms<sup>2</sup> 와 0.95mm/ms<sup>2</sup> 을 각각 나타내고 있다. 특히 피스톤 상사점에서 가속도 변화가 상대적으로 큰 것을 알 수 있다. 가속도의 크기와 변화율을 감소시키기 위해서는 회전부에 질량 밸런서를 부착하여 관성 모멘트를 증가시키는 방법이 있다. 본 논문에서 가진력으로 고려되는 고정조건에서의 반력은 Fig. 20~21 에서 보여주고 있다. O-DYN 과 DADS 의 차 이는 역시 미미한 것을 알 수 있다. 반력의 최대 크기는 x 방향이 y 방향의 3 배 값을 갖는 것을 알

수 있다. 이러한 반력을 이용하여 압축 작용에 따 른 가진력을 예측할 수 있고 그에 따라 제진 방안 의 수립이 요구된다. 또한 설계 인자에 대한 매개 변수 분석이 가능하여 시뮬레이션에 의한 저진동 압축기 설계가 가능하게 된다.

이상과 같이 복동식 압축기의 동적 해석을 통 하여 본 논문에서 소개하는 O-DYN 해의 신뢰성 을 검증하였다.

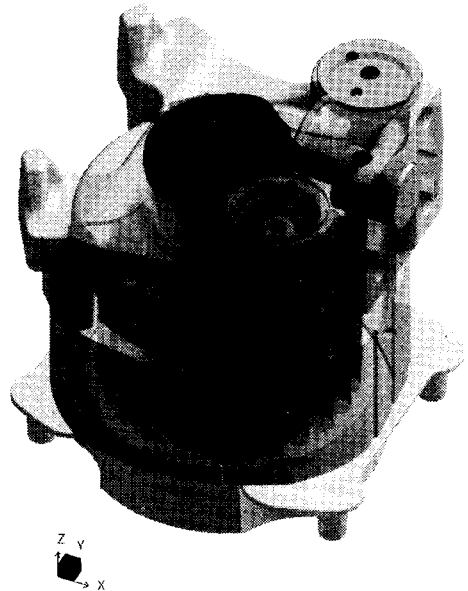


Fig. 13 Reciprocating compressor

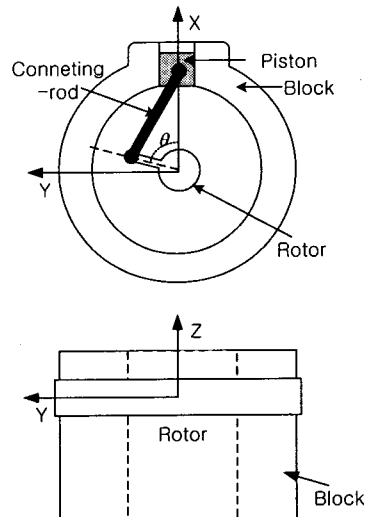


Fig. 14 Schematic diagram for the compressor



Table 2 Inertia properties of the compressor

Body	Mass(kg)	$I_{xx}$ (kgmm <sup>2</sup> )	$I_{yy}$	$I_{zz}$
Block	5.0	20.0	0.05	15.02
Rotor	0.04	500.0	500.0	30.0
Connecting-rod	0.03	1.1	10.0	3.5
Piston	0.03	3.0	3.5	4.0

Table 3 Dynamic model of the compressor

Bodies	Block, Rotor Connecting-rod, Piston
Translational joint	Block-Piston
Revolute joint	Block-Rotor
Revolute joint	Connecting-rod-Piston
Cylindrical joint	Rotor-Connecting-rod
Motor torque	Fig. 15
Compression force	Fig. 16
Rolling resistance	Block-Rotor (2.5 Nms/mm)

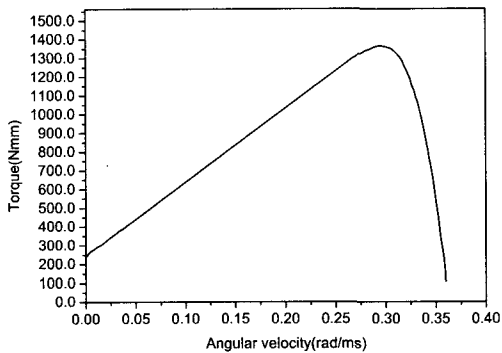


Fig. 15 Characteristics of motor

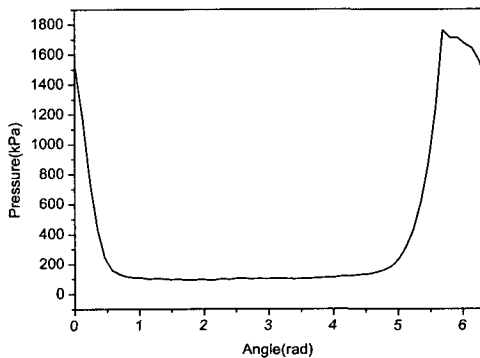


Fig. 16 Pressure versus rotor angle

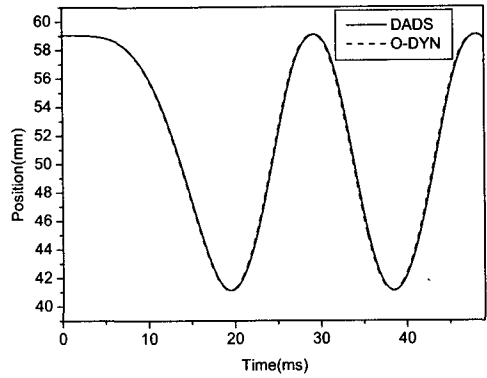


Fig. 17 x position of piston versus time

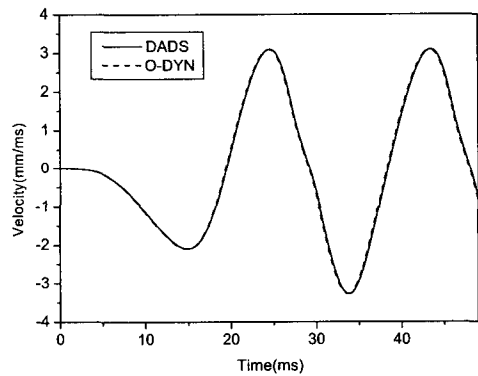


Fig. 18 x velocity of piston versus time

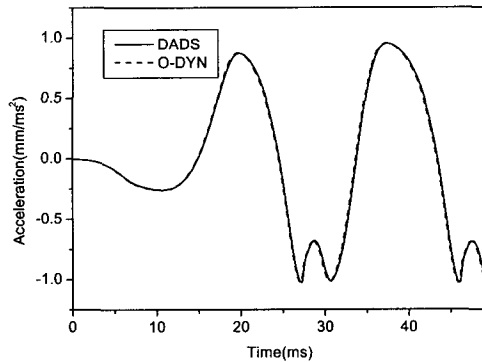


Fig. 19 x acceleration of piston versus time

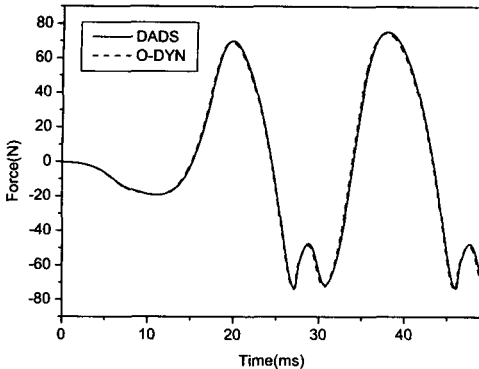


Fig. 20 x reaction force on block at origin

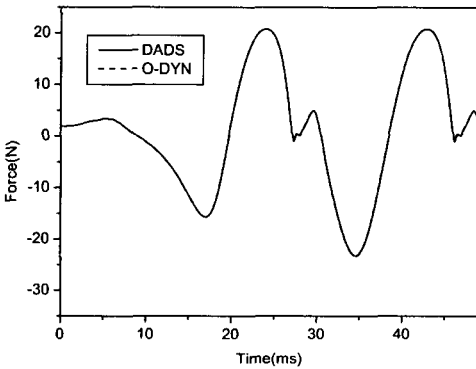


Fig. 21 y reaction force on block at origin

## 5. 결론

본 논문에서는 객체지향 모델링 개념을 적용한 다물체 동역학 해석 시스템 개발 방법을 소개하였다. 이를 통하여 본 논문에서 제시하는 기법이 다음과 같은 장점이 있는 것을 확인하였다. 첫째로는 다물체 시스템 구성 요소를 클래스로 정의하여 독립화 시켰기 때문에 요소의 수정, 교환성, 분산 개발성, 재 사용성을 증가시켰다. 실 예로 분산개발을 실시하고 개발된 클래스를 등록하는 작업이 전체 해석기의 구조에 대한 이해가 없이도 쉽게 수행할 수 있음을 경험하였다. 둘째로는 새로운 해법의 적용이나 타 S/W, H/W와의 인터페이스가 용이할 것으로 기대된다. 이는 새로운 해법을 적용하기 위하여 주 해석기와 라이브러리를 구성하는 요소들의 수정이 필요 없이 필요에 따라 각 객체에게 메시지를 통하여 정보를 주고 받을 수 있기 때문이다. 셋째로는 새로운 요소의 정의

를 상속성을 이용하여 쉽게 정의할 수 있다는 것이다. 결과적으로 물리적 시스템의 컴퓨터 시뮬레이션을 위한 소프트웨어 개발시 객체지향 모델링 개념을 이용하는 것이 합리적이라 사료된다. 또한 다물체 동역학 시스템의 성능 해석에 있어서 요구되는 사용자 중심의 해석기를 개발할 수 있다.

## 후 기

본 연구는 과학기술부에서 지원하는 주력산업의 고부가가치화 사업 “웹기반 범용 동력전달계 성능해석 시스템 개발”과제의 연구비로 수행되었음을 밝히며 본 연구의 지원에 대하여 감사드립니다.

## 참고문헌

1. <http://www.adams.com>
2. <http://www.lmsintl.com>
3. <http://www.simpack.de>
4. <http://www.functionbay.com>
5. Kim, J. H., Cho, C. H. and Kim, S. J., “Distributed Parallel Computing of Finite Element Analysis on a Network Using Object Oriented Programming Paradigm,” The Korean Society for Aeronautical & Space Sciences, Vol. 23, No. 1, pp. 97-106, 1995.
6. Shin, Y. S., Seo, J. K., Park, Y. S. and Choi, H. W., “Development of Object-Oriented Structural Analysis Program for PC,” J. of the Computational Structural Engineering Institute of Korea, Vol. 5, No. 4, pp. 125-132, 1992.
7. Kim, H. K., Lee, J. Y., Kim, J. J. and Lee, B. H., “Application of Object-Oriented Methodology for Structural Analysis and Design,” J. of the Computational Structural Engineering Institute of Korea, Vol. 8, No. 3, pp. 123-133, 1995.
8. Yoon, P. J., Lee, S. J. and Sunwoo, M. H., “A Study on an Engine Control System Using an Object-Oriented Programming Method,” Transactions of Korea Society of Automotive Engineers, Vol. 8, No. 3, pp. 98-109, 1995.
9. Yang, K. J., Hong, K. S. and Lee, K. I., “An Object-Oriented Model for Gasoline Engine and Automatic Transmission Systems,” J. of Control, Automation

- and Systems Engineering, Vol. 4, No. 4, pp. 534-542, 1998.
10. Jung, B. Y., Cho, D. I., "Object-Oriented Programming and Automotive Powersystem," J. of Control, Automation and System Engineering, Vol. 2, No. 2, pp. 127-132, 1996.
  11. Otter, M., Elmqvist, H. and Cbllier, F. E., "Modeling of Multi-body Systems with the Object-Oriented Modeling Language Dymola," Nonlinear Dynamics 9, pp. 91-112, 1996.
  12. Kecskemethy, A., Hiller, M., "An Object-oriented Approach for an Effective Formulation of Multi-body Dynamics," Comput. Methods Appl. Mech. Engrg. 115, pp. 287-314, 1994.
  13. Kecskemethy, A., Lange, C. and Grabner, G., "Object-oriented Modeling of Multi-body Dynamics Including Impacts," ECCM-2001, pp. 1-28, 2001.
  14. Haug, E. J., "Computer Aided Kinematics and Dynamics of Mechanical Systems," Allyn and Bacon, USA, 1989.