

## 다물체동역학 해석 프로그램 CADyna의 제어모듈 개발

### Development of a Control Module in Multibody Dynamics Program CADyna

김 승 오\*, 전 경 진\*\*, 손 정 현\*\*\*, 유 완 석\*\*\*\*  
Seungo Kim, Kyoungjin Jun, Jeonghyun Sohn, Wansuk Yoo

#### ABSTRACT

A procedure to model and simulate control systems is presented using CADyna and MATLAB/Simulink computer codes. For the plant modeling, a technique for obtaining the state matrices from CADyna is presented. To obtain state matrices from CADyna models, perturbation theory is used. These state matrices are then used in MATLAB to design a controller for the plant. The controller design can subsequently be incorporated into the CADyna model and its closed loop performance is evaluated. Examples are presented to verify the developed methodology.

주요기술용어 : State space equation(상태공간방정식), System jacobian(시스템자코비안), Perturbation theory(섭동이론)

#### 1. 서 론

제어를 고려한 기계시스템의 거동에 대한 이해가 해석 및 설계와 관련되어 다양한 목적으로 요구되고 있다. 복잡한 비선형시스템에 대하여 이런 정보를 얻기 위한 방법으로 컴퓨터 시뮬레이션이 있다. 그러나 제어시스템의 해석은 선형화 모델을 이용하여 시뮬레이션하는 한계점이 있어 비선형 기계시스템과의 시뮬레이션을 조합하지 못하고 각각의 특성만 강조하며 특수목적 응용 소프트웨어로 개발되어왔다.

컴퓨터를 이용한 해석 기법을 다물체동역학

분야에 적용하여 해의 신뢰성과 시뮬레이션 효율을 높이기 위한 수치적인 방법들이 활발히 연구되는 가운데, 자동차 산업체에서도 차량동역학 분야에 다물체동역학 해석 기법을 도입하여 설계 단계에서부터 차량의 동적 거동을 해석함으로써 이를 설계에 반영하려는 연구가 활발히 진행되고 있다. 이런 연구를 통해 국내외에서 개발한 ADAMS,<sup>1)</sup> DADS,<sup>2)</sup> RecurDyn<sup>3)</sup> 등 여러 종류의 상용프로그램이 활용되고 있으나, 공급 가격적 측면에서 보면 부품을 생산하는 중소기업이 도입하기에는 다소 부담스러운 것이 사실이다.

따라서, 정부의 지원으로 개발하였던 워크스테이션 버전의 차량동역학 해석 프로그램 AutoDyn<sup>7,5)</sup>을 윈도우NT 버전으로 바꾸면서 전후 처리기를 Visual C++, OpenGL, ActiveX control인 TeeChart를 이용하여 자체 개발하고 차량동역학

\* 회원, (주)일진오토모티브

\*\* 회원, (주)한국타이어

\*\*\* 회원, 부산대학교 기계기술연구소

\*\*\*\* 회원, 부산대학교 기계공학부

해석 프로그램 CADyna(Computer Aided Dynamics)로 명명하였다. 기계시스템의 모델링에 있어서 CADyna<sup>6)</sup>는 매우 다양한 모듈을 통해 광범위한 기계시스템을 모델링하는 것이 가능하다.

최근, 제어시스템 설계가 MATLAB/ Simulink,<sup>7)</sup> MATRIXx/SystemBuild<sup>8)</sup> 등의 뛰어난 제어 해석 범용프로그램을 사용하여 행해지고 있다. 하지만 이들 프로그램은 제어를 위한 동역학 모델을 사용자가 제공하기를 요구하기 때문에, 전달함수를 구하거나 사용자 코드를 제공하여 이 문제를 해결한다. 이 때 제어 해석 범용프로그램이 매우 뛰어난 제어 기법을 제공하여도 동적 모델을 간략화 하기 위하여 여러 가정을 설정하는 것이 보통이다.

마찬가지로 CADyna는 미리 주어진 힘에 대해 시간에 대한 상태만을 제공해줄 뿐 뛰어난 제어를 통합하여 시스템을 모델링하고 시뮬레이션하는 것은 불가능하였다. 따라서 동역학 해석 프로그램과 제어 설계 및 해석 프로그램의 환경을 통합하는 연구가 필요하게 되었다. ADAMS/ Control<sup>9,11)</sup>과 DADS/ Control<sup>12,13)</sup>이 그 예라 하겠다. 본 논문에서는 동역학 해석 프로그램 CADyna와 제어 설계 해석 프로그램 MATLAB의 통합환경을 구현하는 과정과 적용 예제를 통해 각 프로그램의 단점을 보완하여 더욱 실제적인 기계시스템의 모델링이 가능함을 보인다.

## 2. 제어를 위한 CADyna 모델의 선형화

### 2.1 플랜트(plant) 및 제어기 모델

제어시스템 설계<sup>14)</sup>의 관점에서 볼 때, 제어하고자 하는 시스템을 플랜트(plant) 모델로 고려한다. Fig. 1과 같이 플랜트의 출력이 센서에 의해 감지되어 기준입력과 비교 후 오차신호가 계산된다. 이 오차신호는 플랜트 모델의 구동기로의 제어입력을 생성하도록 제어기에 공급된다. 그리고 나서 구동기는 시스템의 작동에 영향을 미치게 된다. 제어기의 주된 역할은 오차를 줄이기 위한 제어입력을 구동기에 주는 것이다.

Fig.1과 같은 플랜트 모델이 비선형 CADyna

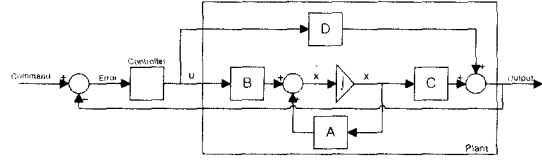


Fig. 1 Closed loop system model

모델 운동방정식으로부터 유도되어야 한다. 선형방정식의 형태로 나타내면 다음과 같은 상태공간방정식으로 표현된다.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad (1)$$

여기서  $x \in R^n$ 은 시스템의 상태벡터,  $u \in R^m$ 은 제어입력벡터,  $y \in R^p$ 는 출력벡터이고, 행렬  $A(n \times n)$ ,  $B(n \times m)$ ,  $C(p \times n)$ ,  $D(p \times m)$ 은 각 벡터 사이의 상호관계를 나타내는 행렬들이다. 상태공간 표현 속에는 센서, 구동기와 관련되어 동역학 시스템 표현식이 포함되어 있고, 그와 관련된 상태가 상태벡터  $x$ 에 포함되어 있다. 일반적으로 출력 피드백 제어법칙은 다음과 같이 게인벡터  $k$ 와 기준입력  $y_0$ 로 표현된다.

$$u = k(y_0 - y) \quad (2)$$

식 (1)과 (2)가 함께 쓰여 폐쇄계시스템을 표현한다.

### 2.2 운동방정식의 선형화

속도변환기법<sup>15,16)</sup>을 사용한 CADyna 운동방정식의 선형화 과정은 Lee<sup>17)</sup>에 의해 제안된 시스템 자코비안을 이용하여 다음과 같이 확장하였다.

$$G(Y, YP, t) = \begin{bmatrix} w - \dot{q} \\ B^T M B \dot{w} - B^T f + B^T M \dot{B} w + \Phi_q^T \lambda \\ y - q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (3)$$

여기서  $y$ 는 출력벡터이고,  $M$ 은 계의 질량행렬,  $f$ 는 일반력,  $q$ 는 일반좌표벡터, 상태변수는

$$\begin{aligned} Y &= [q^T \ w^T \ u^T \ y^T]^T \\ DY &= [\dot{q}^T \ \dot{w}^T \ \dot{u}^T \ \dot{y}^T]^T \end{aligned} \quad (4)$$

으로 정의한다.

식 (3)과 같은 제어를 위한 CADyna 모델의 운동방정식은 작동점  $Y^*=(Y_0, DY_0, t_0)$ 에서 섭동 이론(perturbation method)<sup>14,18)</sup>을 적용시켜 다음과 같이 선형화 시킬 수 있다.

$$\left(\frac{\partial G}{\partial Y}\right) \delta Y + \left(-\frac{\partial G}{\partial DY}\right) \delta DY = 0 \tag{5}$$

여기서,

$$\frac{\partial G}{\partial Y} = \begin{bmatrix} \frac{\partial G_1}{\partial q} & \frac{\partial G_1}{\partial w} & \frac{\partial G_1}{\partial u} & \frac{\partial G_1}{\partial y} \\ \frac{\partial G_2}{\partial q} & \frac{\partial G_2}{\partial w} & \frac{\partial G_2}{\partial u} & \frac{\partial G_2}{\partial y} \\ \frac{\partial G_3}{\partial q} & \frac{\partial G_3}{\partial w} & \frac{\partial G_3}{\partial u} & \frac{\partial G_3}{\partial y} \end{bmatrix}$$

$$\frac{\partial G}{\partial DY} = \begin{bmatrix} \frac{\partial G_1}{\partial \dot{q}} & \frac{\partial G_1}{\partial \dot{w}} & \frac{\partial G_1}{\partial \dot{u}} & \frac{\partial G_1}{\partial \dot{y}} \\ \frac{\partial G_2}{\partial \dot{q}} & \frac{\partial G_2}{\partial \dot{w}} & \frac{\partial G_2}{\partial \dot{u}} & \frac{\partial G_2}{\partial \dot{y}} \\ \frac{\partial G_3}{\partial \dot{q}} & \frac{\partial G_3}{\partial \dot{w}} & \frac{\partial G_3}{\partial \dot{u}} & \frac{\partial G_3}{\partial \dot{y}} \end{bmatrix}$$

이고,  $\delta Y$ 와  $\delta DY$ 는  $Y^*$ 으로 정의된 작동점 주위에서의 미소 섭동 값이다. 또한,  $\delta Y$ 의 요소 중에는 플랜트에 대한 입력과 출력이 포함된다.

### 3. CADyna와 MATLAB의 연결

#### 3.1 연결방법

이 논문에서 제시한 방법론은 Fig. 2에서와 같이 CADyna의 각 모듈로 모델링한 다물체시스템의 개방계에서 시작한다.

CADyna로 기계시스템을 모델링한 후 Fig. 3(b)과 Fig.3(c)에 나타난 CADyna 전처리기의 Control Plant/In, Plant/Out 패널을 이용하여 플랜트 입력과 출력 요소를 선정한 후 Fig. 3의 Export 버튼을 눌러 모델의 플랜트에 대한 센서(sensor)와 구동기(actuator)가 정의된 m-file을 얻게된다. 또한 CADyna/Linear 해석기가 2절에서 논의한 방법으로 작동점  $Y^*$ 에 대하여 모델링한 시스템에 대한 선형화 해석을 한 후 A, B, C, D 행렬의

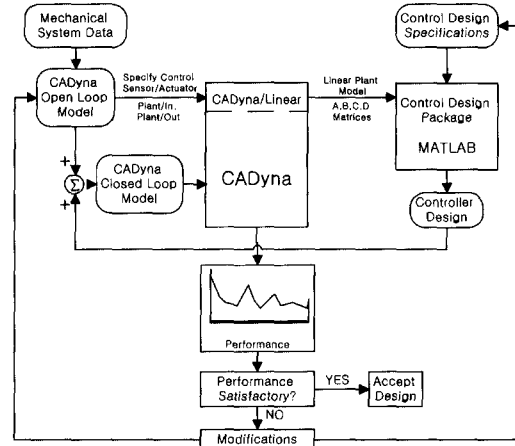


Fig. 2 Schematic diagram of the proposed methodology

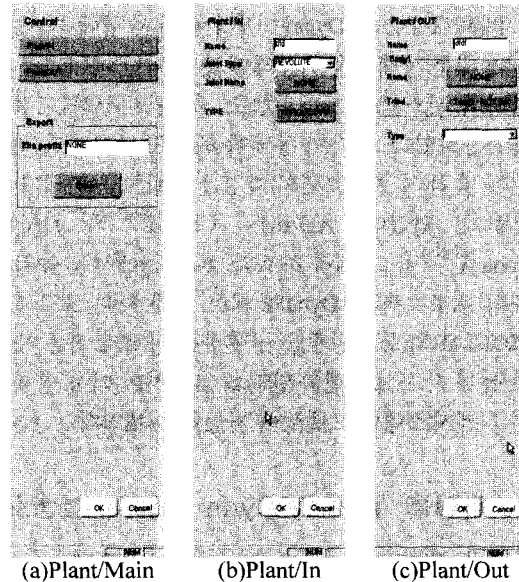


Fig. 3 CADyna/Control panels

값을 MAT File로 얻을 수 있다.

제어 해석 및 설계 범용프로그램으로 본 논문에서 적용시킨 MATLAB을 실행한 후, 앞의 과정에서 얻은 m-file을 명령 창에서 실행하고 'cadyana\_sys'를 실행하면 Fig. 4(a)와 같은 창이 자동으로 뜬다. Fig. 4(b)는 Fig. 4(a)의 cadyana\_sub의 블록 구성도이다. Fig. 4(a)에서 새로운 블록 편집 창을 띄우고 'cadyana\_sub'와 연결된 스코프를

모두 끌어당겨 붙인 후 제어기를 설계하여 시뮬레이션한다. 이 때 얻어지는 입력의 변화값이 CADyna 개방계 모델에 더해짐으로써 폐쇄계 모델이 구성된다. 폐쇄계 CADyna 모델의 시뮬레이션을 통해 수행 결과를 평가하여, 기대했던 결과이면 제어기 설계를 선정하고, 그렇지 않으면 제어기 설계를 수정해나간다. 이런 설계와 시뮬레이션의 과정을 반복적으로 수행함으로써 모델링한 기계시스템에 대한 만족스런 제어 해석 결과를 얻을 수 있다.

### 3.2 MEX File과 MAT File

설계, 해석 프로그램을 통합된 환경으로 결합하는 것은 설계 주기를 줄여주게 된다. CADyna는 동역학, 기구학 해석을 위하여 기계시스템의 모델링, 시뮬레이션하여 시각화하는데 있어 강력한 프로그램이다. 반면 MATLAB은 제어시스템을 모델링하고 설계하는데 있어 강력한 프로그램으로 LQG 설계, 강건제어, 비선형제어와 같은 제어 설계 및 해석 기법을 제공하고 있다. 이와 같이 MATLAB의 뛰어난 성능을 사용하여 CADyna에서 모델링한 기계시스템의 제어 해석이 가능하도록 통합할 필요가 있다.

MATLAB은 외부 프로그램 및 데이터와 상호 작용 할 수 있도록 응용 프로그램 인터페이스(application program interface)를 제공하며,<sup>19)</sup> 예로 MEX 파일과 MAT 파일이 있다.

MEX 파일은 MATLAB 해석기가 자동으로 읽

어 실행할 수 있는 동적 연결 서브루틴이다.

본 논문의 연구에서 S-Function(system - function)과 CADyna/Linear 해석기를 MEX 파일을 이용하여 프로그래밍하여 컴파일 하였다. S-Function은 사용자가 원하는 성능을 구현하는 Simu-link 블록으로 사용자가 C, C++, Fortran 등으로 프로그래밍한 것을 MEX 파일의 문법에 맞게 수정하여 컴파일하여 생긴 동적 연결 라이브러리(dynamic link library) 파일에 의해 그 기능을 연결할 수 있다.<sup>20)</sup>

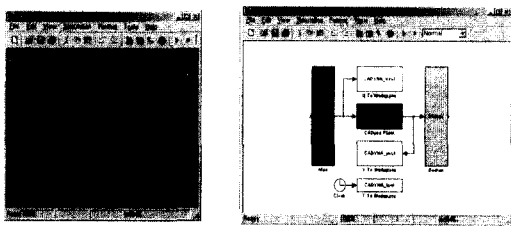
MAT 파일은 MATLAB에서 저장하고(save), 불러오거나(import) 내보낼 때(export) 사용하는 데이터 파일 형태이다. 따라서 MAT 파일을 이용하게 되면 MATLAB 데이터를 사용자가 작성한 C나 Fortran 프로그램에서 읽고 쓰는 게 편리해진다. 본 논문의 연구에서는 CADyna/ Linear 해석기의 해석 결과로 얻어지는 A, B, C, D 행렬을 MAT 파일로 저장하여 MATLAB에서 읽어들이도록 하였다.

## 4. 수치 예제

### 4.1 역진자 모델

회전 조인트의 토크를 조정함으로써 진자의 위치가 바뀌는 1 자유도 역진자 모델의 제어를 예로 다루어 본다. 역진자 모델의 형상과 거리벡터는 Fig. 5와 같으며, 진자의 질량과 길이는 각각 10kg, 2m이다.

단일진자의 질량 행렬  $M$ 과 속도변환행렬  $B$ 을 유도하고, 식 (3)을 상태변수  $Y, DY$ 에 대하여 편미분하여 해석적으로 시스템자코비안을 유도한 후 정리하면 다음과 같다.



(a) CADyna\_sys

(b) CADyna\_sub

Fig. 4 CADyna\_sys and CADyna\_sub in MATLAB/Simulink

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & \frac{1}{3} ml^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \\ \delta u \\ \delta y \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ -\frac{1}{2} m_2 g l \sin\theta_2 & 0 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \\ \delta u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

$K_1$                        $K_2$                        $K_3$                        $K_4$                        $K_5$

상태변수에 대해 식을 정리하면 다음과 같다.

$$\begin{aligned} \begin{bmatrix} \delta\ddot{\theta} \\ \delta\dot{\theta} \end{bmatrix} &= -(M_1)^{-1}K_1 \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \end{bmatrix} - (M_1)^{-1}K_2 \delta u \\ &= \begin{bmatrix} 0 & 1 \\ \frac{3}{2} \frac{g}{l} \sin \theta_2 & 0 \end{bmatrix} \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{3}{ml^2} \end{bmatrix} \delta u \end{aligned} \quad (11)$$

$$\begin{aligned} \delta y &= -(K_3)^{-1}K_3 \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \end{bmatrix} - (K_3)^{-1}K_4 \delta u \\ &= [1 \ 0] \begin{bmatrix} \delta\theta \\ \delta\dot{\theta} \end{bmatrix} \end{aligned} \quad (12)$$

검증을 위하여 라그랑지 식을 사용하여 역진자 모델의 토크방정식을 구하여 상태공간방정식의 형태로 유도하면 다음과 같다.

$$\begin{aligned} x(t) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \quad u(t) = \tau, \quad y(t) = \theta \\ \dot{x} = G(x, u) &= \begin{bmatrix} \dot{\theta} \\ \frac{3}{ml^2} u - \frac{3g}{2l} \cos \theta \end{bmatrix} \end{aligned} \quad (13)$$

시스템방정식을 섭동 원리를 이용하면

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ \frac{3g}{2l} \sin \theta & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{3}{ml^2} \end{bmatrix} u \quad (14)$$

$$y = \theta = [1 \ 0] x$$

이다. 식 (11), (12)와 동일함을 알 수 있다.

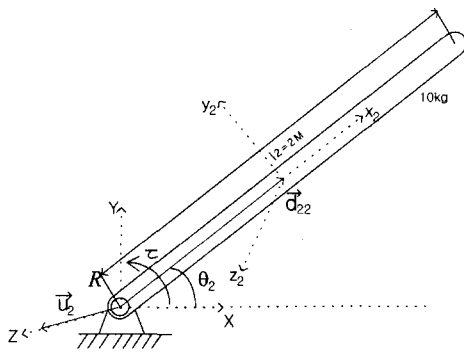


Fig. 5 Definition of vectors in inverted pendulum

CADyna를 이용하여 Fig. 6과 같이 역진자 모델을 모델링한 후 진자가 초기 위치에서 수직 위치에 대하여 안정상태를 찾도록 PD제어 법칙을 이용하여 MATLAB/Simulink에서 Fig. 7과 같이 제어기를 모델링 하였다. Fig. 8은 제어입력(토크)과 제어출력(위치)의 결과를 나타내고 있다.

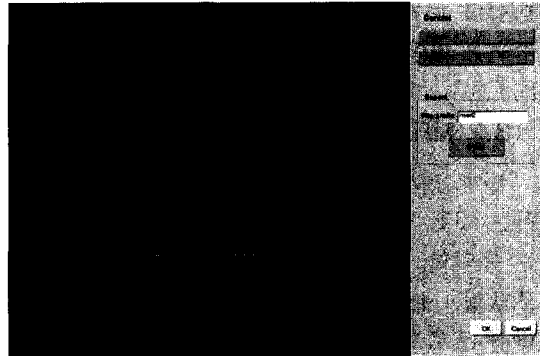


Fig. 6 Inverted simple pendulum model in CADyna

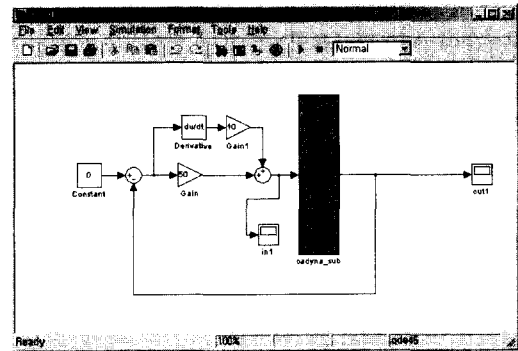


Fig. 7 PD control law for inverted pendulum example in Simulink

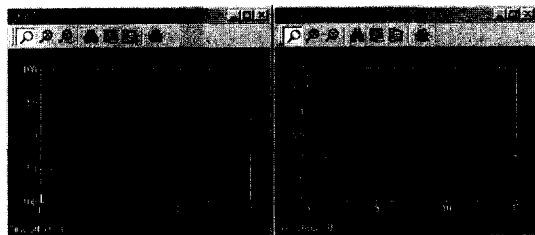


Fig. 8 Joint torque(input) and pendulum angle(output)

### 4.2 역진자-차량 모델

차량에 가하는 힘을 달리하여 차량의 위치를 조정함으로써 차량에 연결된 역진자를 수직 위치로 유지하는 2자유도 역진자-차량 모델의 제어를 예로 다루어 본다. 역진자-차량 모델의 형상과 거리벡터는 Fig. 9와 같으며, 진자의 질량은 10kg, 길이는 2m이며, 차량 질량은 20kg이다.

단일진자의 예와 같이 하면 식 (15)를 얻을 수 있다. 이 때  $\theta_3$ 이 미소각이라고 가정하여  $\cos\theta_3 = 1, \sin\theta_3 = \theta_3$ 로 한다.

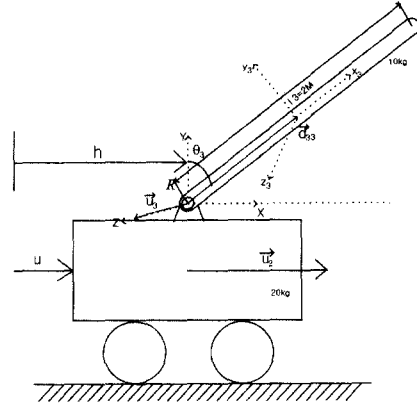


Fig. 9 Definition of vectors in cart with inverted pendulum

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & m_2 + m_3 & 0 & \frac{1}{2} l_3 m_3 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} l_3 m_3 & 0 & \frac{1}{3} m_3 l_3^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta h \\ \delta \dot{h} \\ \delta \theta \\ \delta \dot{\theta} \\ \delta u \\ \delta y \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{2} l_3 m_3 g & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta h \\ \delta \dot{h} \\ \delta \theta \\ \delta \dot{\theta} \\ \delta u \\ \delta y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (15)$$

상태변수에 대해 식을 정리하면 식 (16), (17)이 된다. 단일진자 예에서와 같이 상태공간방정식 형태를 얻을 수 있다.

단일진자의 예에서와 같이 라그랑지 식을 이용한 검증이 가능하였고, 같은 단계를 거쳐 Fig. 10의 역진자-차량 모델의 모델링을 한 후 Fig. 11과 같이 MATLAB/Simulink에서 PD 제어법칙을 이용하여 제어를 모델링하였다. Fig. 12는 제어입력인 차량에 가해진 힘과 제어 출력인 차량의 위치와 진자의 위치의 결과 그림이다.

$$\begin{bmatrix} \delta \dot{h} \\ \delta \ddot{h} \\ \delta \dot{\theta} \\ \delta \ddot{\theta} \end{bmatrix} = -(M_1)^{-1} K_1 \begin{bmatrix} \delta h \\ \delta \dot{h} \\ \delta \theta \\ \delta \dot{\theta} \end{bmatrix} - (M_1)^{-1} K_2 \delta u \quad (16)$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{3m_3 g}{4m_2 + m_3} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{6(m_2 + m_3)g}{(4m_2 + m_3)l_3} & 0 \end{bmatrix} \begin{bmatrix} \delta h \\ \delta \dot{h} \\ \delta \theta \\ \delta \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{4}{4m_2 + m_3} \\ 0 \\ -\frac{6}{l_3(4m_2 + m_3)} \end{bmatrix} \delta u \quad (16)$$

$$\delta y = -(K_5)^{-1} K_3 \begin{bmatrix} \delta h \\ \delta \dot{h} \\ \delta \theta \\ \delta \dot{\theta} \end{bmatrix} - (K_5)^{-1} K_4 \delta u \quad (17)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \delta h \\ \delta \dot{h} \\ \delta \theta \\ \delta \dot{\theta} \end{bmatrix}$$

### 5. 결론

본 연구에서는 속도변환기법을 사용하여 운동방정식을 구성하는 비선형 CADyna 모델로부터 상태공간방정식 형태의 선형화 모델 표현으로 유도하는 방법을 소개하였다. 선형 모델 표현을 통해 얻어지는 A, B, C, D 행렬이 플랜트 모델

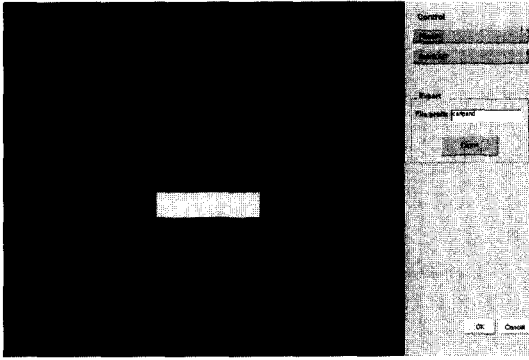


Fig.10 Cart with inverted pendulum model in CADyna

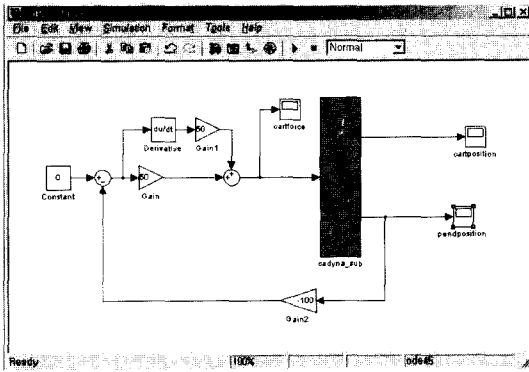


Fig. 11 PD control law for cart with inverted pendulum example in Simulink

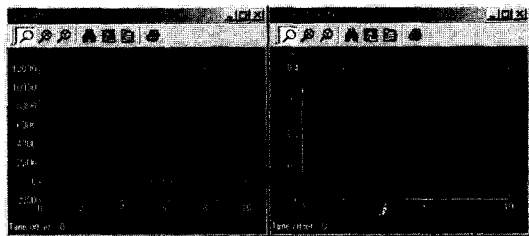


Fig. 12 Force on the cart(input) and pendulum angle(output)

을 위한 제어를 설계하기 위해 MATLAB과 같은 제어 범용프로그램에서 사용될 수 있음을 확인하였다. 또한 CADyna와 MATLAB의 연결 구현을 위해 사용한 MEX 파일과 MAT 파일의 특징과 용도를 소개하였다.

역진자 모델과 차량-진자 모델의 예제에 대하

여 시스템자코비안을 이용하여 비선형 운동방정식을 선형 표현으로 유도한 방법의 신뢰성을 확인하였고, 구현한 CADyna의 제어 모듈을 사용하여 선형화 해석을 수행하여 모델의 A, B, C, D 행렬을 추출하여 MATLAB에서 행렬을 읽어 들여 제어기 설계 및 해석이 가능함을 보았다.

현재까지는 차량동역학 해석 프로그램 CADyna의 동역학, 기구학 해석만이 가능하였지만 본 연구로 CADyna와 제어 해석 및 설계 범용프로그램의 통합환경이 구현되어, CADyna를 이용한 제어 요소를 고려한 기계시스템의 동적 해석이 가능해질 것으로 본다. 물론 속도변환기법을 이용한 운동방정식의 시스템자코비안 구성을 일반화하는 연구와 섭동 원리를 이용한 미소변위에 대한 선형적 표현이 가지는 한계점인 대변형에 대한 선형적 표현에 대한 연구가 뒤따라야 더욱 신뢰할 수 있는 CADyna의 제어모듈이 개발될 것으로 본다.

### 후 기

본 연구는 한국과학재단 목적기초연구(R01-1997-00037) 및 부산대학교 기계공학부의 지원으로 수행되었음.

### 참 고 문 헌

- 1) ADAMS User's Guide, Mechanical Dynamics Inc., Ann Arbor, MI, U.S.A..
- 2) DADS User's Manual, CADSI, Oakdale, IA, U.S.A..
- 3) RecurDyn Program Manual, Function Bay Inc., 2000.
- 4) 김광석, 유완석, 김성수, 김상섭, "차량동역학 해석 프로그램 AutoDyn7의 개발 (I)-프로그램의 구성 및 주요 알고리즘," 한국자동차공학회논문집, 제7권 제3호, pp.321-330, 1999.
- 5) 한중규, 김두현, 김성수, 유완석, 김상섭, "차량동역학 해석 프로그램 AutoDyn7의 개발 (II)-전처리 및 후처리 프로그램," 한국자동

- 차공학회논문집, 제8권 제3호, pp.190-197, 2000.
- 6) 김승오, 유완석, "차량동역학 해석 프로그램 CADyna 개발," 한국자동차공학회 추계 학술대회 논문집, Vol.2, pp.897-901, 2001.
  - 7) MATLAB 5.2 Product Family, The MATH WORKS Inc., 1998.
  - 8) MATRIXx PRODUCT FAMILY, Integrated System Inc., 1996.
  - 9) V. N. Sohoni, J. Whitesell, "Automatic Linearization of Constrained Dynamical Models," Transactions of the ASME, Journal of Mechanisms, Transmissions, and Automation in Design, Vol.108, No.3, pp.300-304, 1986.
  - 10) V. N. Sohoni, "Control Design and Simulation of Systems Modeled Using ADAMS," Proceedings of the 3rd Annual Aerospace Computation Controls Conference, Oxnard, CA, 1989.
  - 11) V. N. Sohoni, J. Winkelman, Control Systems Analysis Using ADAMS and MATRIXx, MDI, Internal Report.
  - 12) J. Yen, C.-C. Chou, Automatic Generation of Constrained Equations of Motion and Its Jacobian Multibody Dynamics, Technical Report R-114, 1991.
  - 13) T.-C. Lin, K. H. Yae, Linearization and Jacobian Evaluation of the Dynamics of Closed-chain Mechanical Systems, Technical Report R-141, 1992.
  - 14) 김종식, 이민철, 한명철, 최재원, 제어시스템 설계, 청문각, 1997.
  - 15) S. S. Kim, M. J. Vanderploeg, A State Space Formulation for Multibody Dynamic Systems Subject to Control, Univ. of Iowa, Technical Report No.84-20, Dec., 1984.
  - 16) W. Jerkovsky, "The Structure of Multibody Dynamics Equations," Journal of Guidance, Control and Dynamics, Vol.1, No.3, pp.173-182, 1978.
  - 17) 이재욱, 손정현, 김광석, 유완석, 속도변환법을 이용한 시스템자코비안 행렬의 구성, 대한기계학회논문집 A, 25(12), pp.1966-1973, 2001.
  - 18) Murdock, James A, Perturbations : Theory and Methods, New York : Wiley, 1991.
  - 19) MATLAB Application Program Interface Guide Version 5, The MATH WORKS Inc., 1996.
  - 20) SIMULINK Writing S-Functions Version 4, The MATH WORKS Inc., 1998.