

# 건설 중장비용 주행 시뮬레이터의 구현

손권\*, 구상화\*\*, 유완석\*, 이민철\*, 이장명\*\*\*

## A Driving Simulator of Construction Vehicles

Kwon Son\*, Sang Hwa Goo\*\*, Wan Suk Yoo\*, Min Cheol Lee\*, and Jang Myung Lee\*\*\*

### ABSTRACT

Vehicle driving simulators have been used in the development and modification of models. A simulator can reduce cost and time through a variety of driving simulations in the laboratory. Recently, driving simulators have begun to proliferate in the automotive industry and the associated research community. This paper presents the hardware and software developed for a driving simulator of construction vehicles. This effort involves the real-time dynamic analysis of wheel-type excavator, the design and manufacturing of the Stewart platform, an integrated control system of the platform, and three-dimensional graphic modeling of the driving environments.

**Key Words :** Driving simulator(주행 시뮬레이터), Stewart platform(스튜어트 플랫폼), Dynamic analysis(동역학 해석), Driving environment(주행 환경)

### 1. 서론

각종 운송장비의 개발·설계단계에서 시뮬레이터는 매우 중요한 역할을 담당한다. 차량 운전 시뮬레이터는 차량에 대해 실시간 시뮬레이션을 수행해 그 운전상황을 운동, 시각 및 음향 효과를 통해 제시함으로써 운전자가 실제로 차량을 운전하는 느낌을 얻을 수 있는 가상현실체험기이다. 이러한 시뮬레이터를 이용하면 실제 차량이 운전하는 상황을 실험실 내에서 재현할 수 있으므로 차량에 대한 각종 평가를 실험실 단위에서 용이하게 수행할 수 있는 이점이 있다. 운전 시뮬레이터는 차량의 설계 변수나 운전 조건의 변화가 운전성능에 어떻게 영향을 미치는지를 평가하는 데 긴요하게 사용되며, 승차감 개선을 위한 운전석의 설계에서도 크게 활

용된다. 차량의 모델 변경이나 개발 단계에서 설계가 바뀔 때마다 새 모형을 만들어 시험하는 것은 경제적으로 불가능하므로, 신 차량의 개발에는 시뮬레이터의 구비가 필수적이다. 특히 차량의 충돌, 추락, 미끄러짐과 같은 한계 상황에서 차량의 거동이 운전자에게 끼치는 영향을 알아보기 위해서도 가상실험을 수행할 수 있는 시뮬레이터가 반드시 요구된다.

현재 선진국의 주요 자동차 회사에서는 차량 시뮬레이터를 자체적으로 설계·제작하여 개발에 활용하고 있다<sup>(1-5)</sup>. 이들은 차량의 각종 운전상태를 시뮬레이션하기 위하여 엔진, 변속기, 구동장치, 현수장치, 조향기구 등 차량 서브시스템 모델, 동역학 모델, 구름저항, 공기저항 등 주행저항 모델, 가상의 주행환경은 물론 특정 시험로의 지형 모델

\* 부산대학교 기계공학부

\*\* 부산대학교 대학원 기계공학과

\*\*\* 부산대학교 전자공학과

까지 개발하였으며, 이의 실시간 처리를 위하여 병렬처리 컴퓨터 혹은 여러 대의 서버급 컴퓨터를 사용하고 있다. 운동의 재현을 위하여 8,000 ~ 18,000 kg의 큰 부하를 처리할 수 있는 대용량의 운동재현기를 개발하였으며, 그래픽 처리를 위하여는 그래픽 전용 컴퓨터를 사용하고 있고 고해상도의 그래픽 투영을 위하여 공통적으로 3대 이상의 영상 발생기(image generator)를 사용하고 있다. 이들 시뮬레이터는 다양한 상황에서의 운전자 거동, 새로운 차량 또는 서브 시스템의 개발 및 성능 개선, 도로 및 교통환경에 대한 연구 등 광범위한 목적으로 활용되어질 수 있지만 설계·제작에는 상당한 비용이 소요되기 때문에 세계적으로 일부 대형 자동차 회사에서만 사용되고 있는 실정으로 활용분야는 제한적이지만 경제적으로 구현할 수 있는 시뮬레이터의 개발이 필요하다. 한편, 우리나라의 자동차 회사에서도 몇대의 시뮬레이터를 보유하여 사용하고 있지만, 우리 자체의 기술로 제작한 것이 아니고 대부분 외국의 제품을 도입하여 사용하고 있는 실정이다. 국내의 일부 연구소와 대학에서도 몇 년 전부터 차량의 시뮬레이터에 관한 연구가 시도되고 있지만, 대부분의 연구가 한정된 분야에 대해 다루고 있으며 아직까지 실용화하기 위해서는 지속적인 연구가 필요하다<sup>(6-9)</sup>.

본 연구에서는 건설 중장비용 시뮬레이터 개발의 첫 단계로 휠 굴삭기의 주행 상황을 모사할 수 있는 주행 시뮬레이터를 구현하였다. 동역학 해석에서는 속도변환법을 이용한 고유의 프로그램을 사용하여 실시간 해석을 시도하였고, 굴삭기의 운동을 재현하기 위하여 500 kg의 부하를 처리할 수 있는 유압식 6자유도의 스텐더트 플랫폼을 설계·제작하였다. 주행 속도감을 운전자에게 제공하기 위하여 범용 소프트웨어 OpenGL을 이용한 그래픽 프로그램을 구성하여 주행 상황을 용이하게 구현할 수 있게 하였다. 그래픽 소프트웨어 상에서의 시뮬레이션에서부터 하드웨어의 작동 평가와 통합 운영시 성능 분석까지 가능한 통합 제어 시스템을 개발하여 전체 시스템을 효율적으로 운용할 수 있도록 하였다.

## 2. 주행 시뮬레이터의 구성

주행 시뮬레이터는 Fig. 1과 같이 스텐더트 플랫폼, 유압공급기, 제어기, 영상 주사장치 및 컴퓨

터들로 구성된다. 본 연구팀에서 제작한 스텐더트 플랫폼은 6자유도의 병렬형 기구로 최대 행정 500 mm인 6개의 유압실린더에 의해 작동 된다. 스텐더트 플랫폼 위에는 굴삭기 캡의 일부를 장착하여 운전석으로 사용하였다. 제어기로는 외부 입력을 빠르게 처리해줄 수 있는 고속의 입출력 장치인 80C196KC를 이용하였다. 이 프로세서는 인텔사의 16비트 제어기로 성능이 우수하고 전력소모가 적은 장점이 있으며, 입력단과 출력단 및 메모리를 갖추고 있다. 입력단을 통해 시뮬레이터의 조작부로부터 디지털 신호로 변환된 값을 입력받고, 입력된 디지털 신호에 따라 출력단으로 실린더를 제어하는 신호를 출력한다. 메모리는 시뮬레이터의 전반적인 운동상황을 관리하는 프로그램을 저장하는 프로그램 메모리와 각 센서로부터 받은 신호를 저장하는 데이터 메모리로 구성된다. 동역학 해석 컴퓨터는 타이머 모듈을 포함하여 굴삭기를 동역학적으로 자유도를 유지하면서 실시간으로 해석이 가능하도록 단순화된 모델을 이용하여 주행상태를 해석한다. 그래픽 컴퓨터는 굴삭기의 외형과 주행환경을 포함하는 그래픽 처리 외에도 시뮬레이터를 운영하는 통합 프로그램이 실행된다.

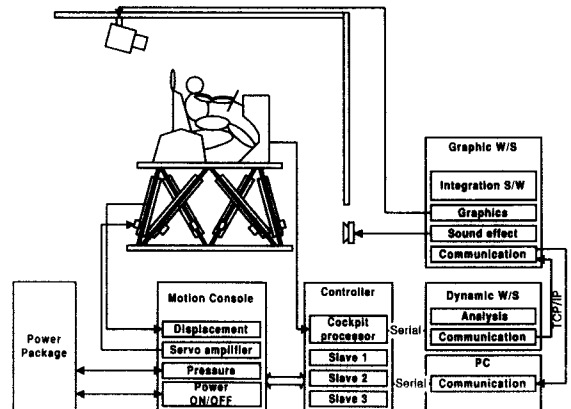


Fig. 1 Driving simulator of construction vehicles

### 2.1 차량 동역학

본 연구에서는 휠 굴삭기 시뮬레이터 구현을 위한 실시간 동역학 해석 시뮬레이션을 위해 실제 굴삭기의 자유도를 벗어나지 않는 범위에서 모델을 최대한 단순화 시켰다. 기존의 상용화된 DADS, ADAMS 등의 프로그램은 범용적인 프로그램이므로

범용성은 있으나, 특수프로그램에 비하여 컴퓨터 사용시간이 길어짐으로 휠 굴삭기의 실시간 시뮬레이션에 적용하기에는 어렵고 데이터 입출력에 있어서도 어려움이 따른다.

따라서, 본 연구에서는 속도변환법을 이용하여 운동방정식을 유도하였는데, 속도변환법은 절대좌표계에서의 속도를 상대좌표계에서의 속도로 변환함으로써 상대좌표계의 효율성과 절대좌표계의 일 반성을 동시에 얻고자 하는 방법이다<sup>(10)</sup>. 새로운 알고리즘에 적용할 굴삭기의 모델은 차체, 앞쪽 차 축, 및 타이어 4개로 구성되고, 회전조인트 3개와 유니버설 조인트 2개로 각 물체가 연결되므로 총 13 자유도를 가지게 된다.

절대좌표계를 이용하여 시스템의 운동방정식을 표시하면 다음과 같은 형태로 표시된다<sup>(10)</sup>.

$$M\dot{y} + \Phi_y^T \lambda = Q \quad (1)$$

여기서,  $y$  는 절대좌표(Cartesian coordinates),  $M$  는 절대좌표계에서의 질량행렬,  $\Phi_y$  는 자코비안행렬(Jacobian matrix)을 나타내며,  $Q$  는 일반력(generalized force)을 의미한다. 이 시스템의 운동방정식은 상대좌표  $q$  의 함수인 속도변환행렬  $B$  를 이용하여 다음과 같은 형태로 유도된다.

$$\overline{M}\dot{q} + \Phi_q^T \lambda = \overline{g} \quad (2)$$

$$\overline{M} = B^T M \quad (3)$$

$$\overline{g} = B^T (f - M\dot{B}\dot{q} - h) \quad (4)$$

여기서  $\overline{M}$  는 일반질량행렬,  $\overline{g}$  는 수정일반력(modified generalized force)을 의미한다.

닫힌 계(closed loop)에서의 구속조건식은 절단조인트 구속식(cut-joint constraints)과 시스템에 임의로 부가되는 구동구속식(driving constraints)으로 분류되어 진다. 이 중 구동구속식은 시스템을 구동하기 위해 가해지는 조건식이다. 속도변환법을 사용하는 경우 구동구속식은 상대좌표의 함수로 간단하게 표현되지 않는 경우도 있으나, 본 연구에 사용된 굴삭기 모델은 실시간 해석을 위해 단순화된 모델이므로 주행을 위한 구동구속의 부가는 어렵지 않았다.

주어진 조인트 좌표에 대한 구동 구속조건식은 시간의 함수로서 다음과 같이 표현된다<sup>(11)</sup>.

$$\Phi^d \equiv \Phi^d(q, t) = 0 \quad (5)$$

여기서 윗첨자  $d$ 는 구동구속을 의미한다. 식 (5)의 시간에 대한 1차 미분과 2차 미분을 구하면 다음과 같다.

$$\Phi_q^d \dot{q} = -\Phi_t^d \quad (6)$$

$$\Phi_q^d \ddot{q} = -(\Phi_q^d)_q \dot{q} + 2\Phi_{qt}^d \dot{q} + \Phi_{tt}^d \equiv \gamma \quad (7)$$

여기서  $\gamma$  는 가속도식의 오른쪽항(right-hand side of acceleration equation, R.H.S),  $\Phi_q^d$  는 구동구속의 자코비안 행렬  $\Phi_q^d \equiv \frac{\partial \Phi^d}{\partial q}$  을 의미한다. 따라서, 구동구속이 있는 경우에 최종적으로 구해진 운동방정식은 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} B^T M B & \Phi_q^{dT} \\ \Phi_q^d & 0 \end{bmatrix} \begin{bmatrix} \dot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q \\ \gamma \end{bmatrix} \quad (8)$$

개발된 알고리즘을 이용하여 J-선회 및 차선 변경 시뮬레이션 결과 DADS에 의한 결과와 잘 일치하였으며, 해석시간 면에서는 DADS에 비하여 약 1/10로 줄어들어 실시간 동역학 해석이 가능한 것으로 판단된다<sup>(12)</sup>.

## 2.2 플랫폼 제어

휠 굴삭기의 동역학 해석을 통하여 구한 굴삭기의 운동을 시뮬레이션하기 위하여는 공간상의 운동을 플랫폼 구동기의 길이로 표현할 수 있는 링크영역으로 변환하여야 하며 이를 위하여 스튜어트 플랫폼의 역기구학 해석을 수행하였으며, 구동기의 길이를 제어하기 위한 제어를 설계하였다.

Fig. 2는 스튜어트 플랫폼의 기하학적 형태를 간단하게 도식화한 것이다. 하판(base plate)에 대한 상판(payload plate)의 운동을 기술하기 위해 그림과 같이 상판 및 하판 각각의 중심에 좌표

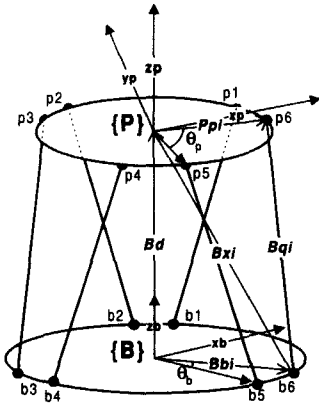


Fig. 2 Coordinate system of Stewart platform

계 {P}와 {B}를 설정하고, 6개의 구동기가 상판과 하판에 연결되는 조인트의 위치를 각각의 좌표계에서  $p_i$  및  $b_i$  ( $i=1, \dots, 6$ ) 라 둔다.

각 구동기 길이 벡터  $B_{qi}$ 와 좌표계 {B}의 원점에 대한 좌표계 {P}의 원점을 나타내는 벡터  $B_d$ 를 나타내면 다음과 같다.

$$B_d = [x \ y \ z]^T \quad (9)$$

$$B_{qi} = [q_{ix} \ q_{iy} \ q_{iz}]^T \quad (10)$$

$$B_{qi} = B_{xi} + B_{bi} \quad (11)$$

여기서  $B_{xi}$ 는 하판의  $b_i$  지점에서 상판에 부착된 좌표계 {P}의 원점까지의 거리벡터이며,  $B_{bi}$  벡터는  $P_{pi}$  벡터를 좌표계 {B} 즉 기준좌표계에서 표현한 벡터이다. 위 식에서 각각의 벡터를 표현하면 다음과 같다.

$$B_{xi} = B_d - B_{bi}$$

$$= \begin{bmatrix} x - b_{ix} \\ y - b_{iy} \\ z - b_{iz} \end{bmatrix} = \begin{bmatrix} x - b_{ix} \\ y - b_{iy} \\ z \end{bmatrix} = \begin{bmatrix} \bar{x}_i \\ \bar{y}_i \\ \bar{z}_i \end{bmatrix} \quad (12)$$

상판의  $P_{pi}$ 의 거리벡터를 하판에서 보았을 때의 벡터로 표현한  $B_{pi}$ 는 다음과 같이 구할 수 있다.

$$B_{pi} = {}^P_B R P_{pi} \quad (13)$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} p_{ix} \\ p_{iy} \\ p_{iz} \end{bmatrix} = \begin{bmatrix} r_{11}p_{ix} + r_{12}p_{iy} \\ r_{21}p_{ix} + r_{22}p_{iy} \\ r_{31}p_{ix} + r_{32}p_{iy} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \\ w_i \end{bmatrix}$$

여기서 행렬  ${}^P_B R$ 는 좌표계 {P}의 좌표계 {B}에 대한 방향을 나타내는 변환 행렬이며 브라이언트 각 (Brayant angle)<sup>(11)</sup>을 사용하면 다음과 같이 된다.

$${}^P_B R = \begin{bmatrix} C_\beta C_\gamma & -S_\gamma C_\beta & -S_\beta \\ C_\gamma S_\alpha S_\beta + C_\alpha S_\gamma & -S_\alpha S_\beta + C_\alpha C_\gamma & -S_\alpha C_\beta \\ -C_\gamma C_\alpha S_\beta + S_\alpha S_\gamma & C_\alpha S_\beta S_\gamma + S_\alpha C_\gamma & C_\alpha C_\beta \end{bmatrix} \quad (14)$$

따라서 구동기인 실린더의 길이  $l_i (= |B_{qi}|)$ 는 식 (10), (11), (12)를 이용하면 다음과 같이 구할 수 있다.

$$l_i = \sqrt{q_{ix}^2 + q_{iy}^2 + q_{iz}^2}$$

$$= \sqrt{(\bar{x}_i + u_i)^2 + (\bar{y}_i + v_i)^2 + (\bar{z}_i + w_i)^2} \quad (15)$$

역기구학 해석결과에 의한 각 실린더의 길이 명령 신호에 따른 실린더의 운동을 제어할 수 있는 제어기를 설계·제작하였다. 제어기는 1개의 메인 제어기와 각각 2개씩의 실린더를 제어하는 3개의 슬레이브 제어기로 구성되어 있다. 메인 제어기는 슬레이브 제어기의 중앙처리장치인 80C196KC의 주변기능들에 대한 기능설정과 하드웨어 초기화를 위한 프로그램과 통신프로그램으로 구성되어 있다.

슬레이브 제어기의 프로세서로는 성능이 우수하고 전류소모가 적은 인텔사의 16비트 마이크로 콘트롤러 MCS-96계열의 CMOS형인 80C196KC를 사용하였다. 슬레이브 제어기는 Fig. 3과 같이 입력단, 출력단 및 메모리로 구성되어 있다. 입력단은 시뮬레이터의 실린더로부터 동작 신호를 검출한 후 디지털 신호로 변환된 값을 입력 받을 수 있고, 출력단은 입력신호에 따라 시뮬레이터의 실린더를 제어하기 위한 신호를 출력할 수 있다. 그리고 메모리는 시뮬레이터의 전반적인 운동상황을 관리하는 프로그램을 저장하고 있는 프로그램 메모리와 각 입출력 센서들로부터 받은 신호를 저장하기 위한 데이터 메모리를 갖추고 있다.

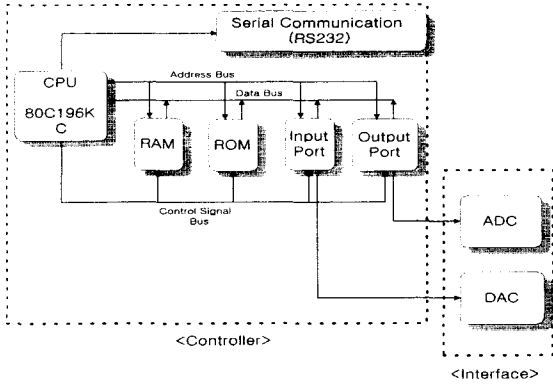


Fig. 3 Slave controller

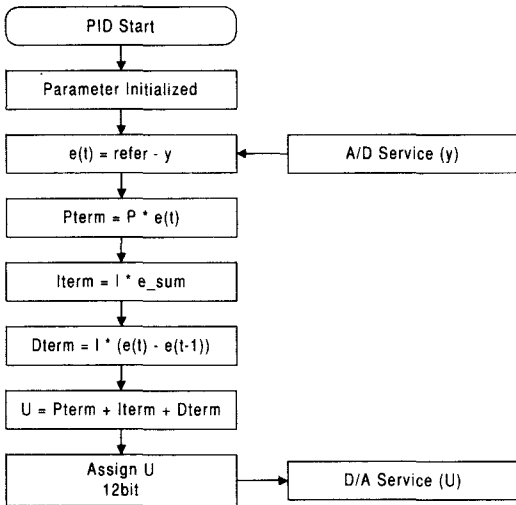


Fig. 4 PID control logic

Table 1 Motion platform requirements

6 DOF motion	Maximum displacement	Maximum velocity	Maximum acceleration
Roll	20 deg	20 deg/sec	60 deg/sec <sup>2</sup>
Pitch	25 deg	20 deg/sec	60 deg/sec <sup>2</sup>
Yaw	30 deg	20 deg/sec	60 deg/sec <sup>2</sup>
Surge	432 mm	610 mm/sec	0.6 mm/sec <sup>2</sup>
Sway	432 mm	610 mm/sec	0.6 mm/sec <sup>2</sup>
Heave	432 mm	610 mm/sec	0.6 mm/sec <sup>2</sup>

실린더의 길이를 제어하기 위하여는 PID 알고리즘을 사용하였는데 80C196KC내에서 어셈블러로 프로그램을 작성하였다. Fig. 4는 PID 제어 프로그램의 블록 다이어그램이다.

### 2.3 플랫폼 설계

굴삭기의 주행 및 작업시의 가능한 운동을 충실히 재현하기 위해서는 공간상에서 6자유도 운동을 재현해 주어야 한다. 병렬형 매니퓰레이터인 스튜어트 플랫폼을 이용하면 이러한 운동이 가능하다. 스튜어트 플랫폼을 설계하기 위하여 Table 1과 같은 항공기 시뮬레이터에 적용되는 MIL-STD-1588<sup>(13)</sup>을 참고하였다.

플랫폼에 걸리는 최대부하는 상판과 상판에 놓이게 될 굴삭기 캡을 고려하여 500 kg으로 정하였다. 최대속도는 가장 큰 값을 가지는 z 방향의 히브 운동에 안전률을 고려하여 ± 61 cm/sec로 하였다. 그리고, 굴삭기의 주행특성을 충분히 재현하기 위해 실린더의 스트로크를 ± 250 mm로 정하였다. 부하에 대한 피스톤 로드와 좌굴 등을 고려하여 피스톤 로드의 직경을 30 mm로 하였다. 또한 굴삭기의 운동을 상판의 운동에 의해 충분히 재현하기 위한 액츄에이터의 힘을 고려하여 실린더 헤드의 직경을 40 mm로 하였다. 공급압력은 최대부하와 실린더 헤드부의 면적을 이용하여 구하였다. 서보 밸브는 액츄에이터 상승시(z 방향) 최대유량이 필요하므로 정격유량 15 gpm으로 선정하였다. Figs. 5와 6은 이상과 같은 설계로 제작된 스튜어트 플랫폼과 구동원인 유압 유니트이다.



Fig. 5 Stewart platform

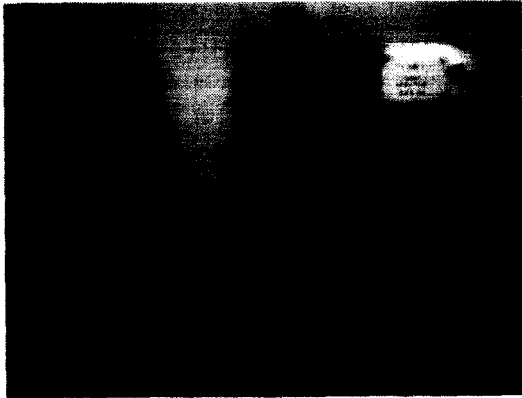


Fig. 6 Hydraulic unit

Table 2 Workspace of Stewart platform

Motion	Workspace
Roll	+25 deg ~ -25 deg @ Heave = 0 mm
Pitch	+23 deg ~ -25 deg @ Heave = 0 mm
Yaw	+20 deg ~ -20 deg @ Heave = 0 mm
Surge	+480 mm ~ -380 mm @ Heave = 0 mm
Sway	+400 mm ~ -400 mm @ Heave = 0 mm
Heave	+320 mm ~ -320 mm @ Yaw = 0 deg

병렬형 6자유도 매니플레이터의 작업영역을 직접 유도하기는 매우 어려우므로 Pennock과 Kassner<sup>(14)</sup>가 제안한 조절할 수 있는 링크를 가진 4절 기구를 이용, 각 실린더와 상판과의 관계를 유도하여 시뮬레이션을 통해 제작된 플랫폼의 작업영역을 확인하였다<sup>(15)</sup>. 시뮬레이션 결과는 Table 2와 같다. 이러한 값을 Table 1의 요구조건과 비교하면 롤, 피치, 요의 각변위는 요구조건을 만족하였다. 그러나 서지, 스웨이, 히브의 값은 요구조건에는 약간 미흡하였다.

## 2.4 그래픽 처리

시뮬레이터에 표현되는 그래픽은 실시간으로 표현되어야만 의미가 있다. PC로는 많은 데이터의 양과 화면 처리 속도를 도저히 따라갈 수 없으므로 그래픽 표현을 위해 워크스테이션을 사용하였다. 사용 기종은 실리콘 그래픽사의 워크스테이션

(SGI Indigo2 Impact R4000)을 선정하였다. SGI 워크스테이션은 그래픽 전용 워크스테이션이라고 할 만큼 그래픽 기능이 뛰어나다. 또한 GL 혹은 OpenGL이라는 그래픽 라이브러리를 제공하므로 이와 C 컴파일러를 이용하여 그래픽을 표현하였다.

### 2.4.1 대상의 표현

그래픽에서의 대상물의 표현은 가장 간단한 와이어프레임 모델과 음영 처리된 셰이딩 모델로 크게 나눌 수 있다. OpenGL은 3차원의 데이터를 이용할 수 있으므로 곡면이 없는 경우 각 모서리나 꼭지점의 좌표만 입력하면 와이어프레임 모델은 쉽게 구성할 수 있다. 그러나, 선으로만 구성되는 와이어프레임 모델은 간단하고, 속도도 빠르지만 현실감이 없으므로 본 시뮬레이터에서는 셰이딩 모델을 이용하여 화면을 구성하였다.

실제 자연계는 아날로그 신호로 구성되지만, 컴퓨터의 모니터는 디지털 신호이므로 선이나 곡면을 그리면 알리에이징 현상이 나타난다. 따라서 본 연구에서는 주변 색의 농도를 조절함으로써 알리에이징을 어느 정도 제거하여 실제와 같은 부드러운 효과를 나타내도록 하였다.

원근 효과를 나타내기 위하여는 GLU 라이브러리의 gluPerspective 함수를 사용하였다. 그러나 먼 거리에 위치한 배경은 원근 효과가 거의 나타나지 않고 시점도 거의 고정되어 있어 이러한 대상의 표현을 위해서 3차원의 자료를 처리한다는 것은 비효율적이므로 먼 곳의 배경을 표현하기 위해 2차원의 텍스처 모델을 사용하였다.

음영 처리는 그래픽 계산 시간의 대부분을 소모하는 과정으로 알고리즘도 아주 다양하고, 알고리즘에 따라 속도, 해상도 등도 차이가 나타난다. 본 시뮬레이터에서는 조명의 효과를 고려한 부드러운 음영 처리 방법을 사용하고, 곡면의 부드러운 표현을 위해 면의 방향을 연속적인 값으로 사용하였다.

### 2.4.2 그래픽 처리 속도 개선 기법

그래픽 처리과정은 본 프로그램에서 가장 많은 데이터를 이용하고 계산량도 가장 많은 부분이다. 많은 데이터와 계산량으로 인해 그만큼 처리 속도가 저하된다. 그래픽 처리 속도를 향상시키기 위해 다음의 몇 가지 방법을 사용하여 실시간 시뮬레이션이 가능하도록 프로그램하였다.

시뮬레이션 과정 중에는 동일한 대상이 시각의

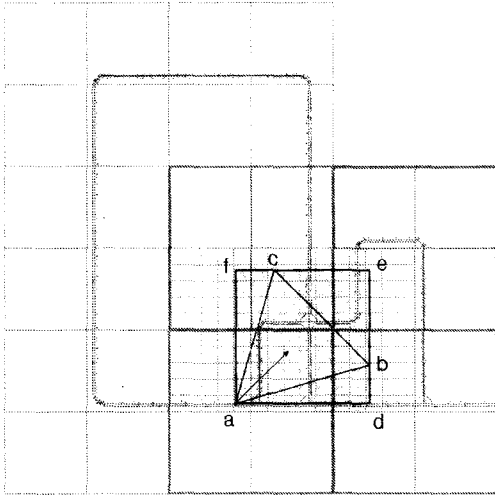


Fig. 7 Driving environment block

위치만 바뀌어 계속 표현된다. 이처럼 계속해서 반복적으로 표현되는 3차원 모델은 가장 빠른 호출장소인 메모리에 저장하여 빠르게 재생시킨다. 프로그램이 기동되면 모델 데이터를 읽어서 메모리에 저장하여 디스플레이 리스트(display list)를 생성한다. 디스플레이 리스트를 생성할 때에는 고유한 번호가 부여되고, 부여된 번호를 호출하여 저장된 자료를 이용할 수 있다. 모델을 다시 화면에 표시할 때는 메모리에 저장되어 있는 데이터를 직접 이용하여 그래픽 처리에 이용하므로 처음보다 속도가 향상된다. 메모리를 이용한 방법은 모델의 형상과 사용 횟수에 따라 속도 향상의 차이가 있는데, 사용 횟수가 많거나 모델의 자료가 복잡할수록 속도에서 많은 향상을 기대할 수 있다.

주행 시뮬레이터는 다른 그래픽과는 달리 주변에 존재하는 대상의 수가 많다. 많은 대상 중 시야에 나타나는 것은 일부분이지만, 시야에 나타나지 않는 대상도 그래픽 처리시 계산되어 많은 시간이 소요된다. 본 연구에서는 실제 존재하지만 시야에 나타나지 않는 대상은 계산에서 제외하는 방법을 사용하여 그래픽 속도를 향상시켰다.

Fig. 7은 굴삭기의 주행 상황을 위쪽에서 바라본 것이다. 정사각형은 세 개의 층으로 구성되는 주행 환경 블록으로 기준위치 검색의 비교 대상이 된다. 굴삭기는 굵은 선으로 표시된 도로 위의 한 지점인 a에 위치하고 있으며, 화살표 방향으로 시점이 향하고 있다. 삼각형 a-b-c는 시야에 나타나

는 영역으로 이 영역 내에 포함되는 대상만이 시뮬레이터의 주행 환경으로 나타난다. 시야 영역의 기준위치는 시야 영역 삼각형의 세 꼭지점을 포함하는 직사각형 a-d-e-f의 중심이며, 주변 환경의 기준 위치는 주행 환경 블록의 중심이 된다. 시야 영역의 기준 위치  $V.basic$ 과 시야 영역의 크기  $V.size$ 는 다음 식과 같이 표현된다.

$$V.basic = \frac{1}{2}(Max(a, b, c) + Min(a, b, c)) \quad (16)$$

$$V.size = |Max(a, b, c) - Min(a, b, c)| \quad (17)$$

주행 환경의 기준 위치와 크기는 주행 환경 블록에 의해 고정된 값으로  $P.basic$ 와  $P.size$ 로 나타내고, 시야 영역과 비교하면 다음의 식을 얻을 수 있다.

$$|P.basic - V.basic| \leq \frac{1}{2}(P.size + V.size) \quad (18)$$

시야 영역과 주행 환경 영역의 기준 위치를 식 (18)과 같이 비교하여 수식을 만족하지 않으면 영역을 벗어난 것으로 판단하여 계산에서 제외시킨다. 기준 위치 비교에 소요되는 시간을 단축하기 위하여 제일 먼저 가장 큰 첫 번째 단계의 블록 위치를 비교하여, 영역 내에 포함되면 두 번째 단계의 블록 위치를 비교한다. 이러한 방법의 반복으로 비교하는 블록의 범위를 줄여 나간다. 결과적으로 Fig. 7의 직사각형 a-d-e-f 내에 포함된 최소 단위의 블록만을 계산하여 그래픽 처리를 하게 된다.

## 2.5 시스템의 통신

스튜어트 플랫폼을 구동하기 위한 전체적인 통신은 Fig. 8과 같이 구성하였다. 동역학 해석 컴퓨터에서 동력학을 계산하여 그래픽 컴퓨터로 데이터를 전송한다. 이때의 데이터 전송은 LAN을 통한다. 그래픽 컴퓨터에서는 실제 구동되는 스텐더트 플랫폼의 화상을 디스플레이하며 동시에 허브를 통하여 PC로 실린더의 길이, PID 제어 알고리즘을 수행하기 위한 개인 값을 전송한다. PC와 전용제어기와 통신은 EPP(enhanced parallel port) 모드의 양방향 병렬 통신을 구현하였다. 전용제어기는 제어루프를 통하여 계산된 값을 매니플레이터로 전송하여 제어하는 구성을 하였다. 전용제어기 내에서는 멀

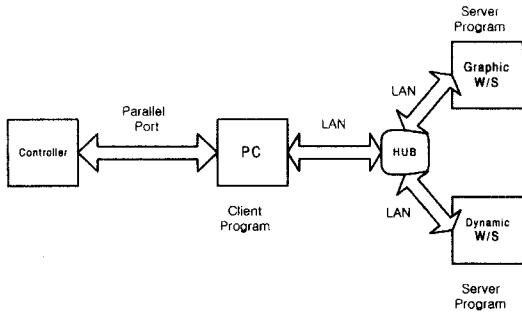


Fig. 8 Organization of communication system

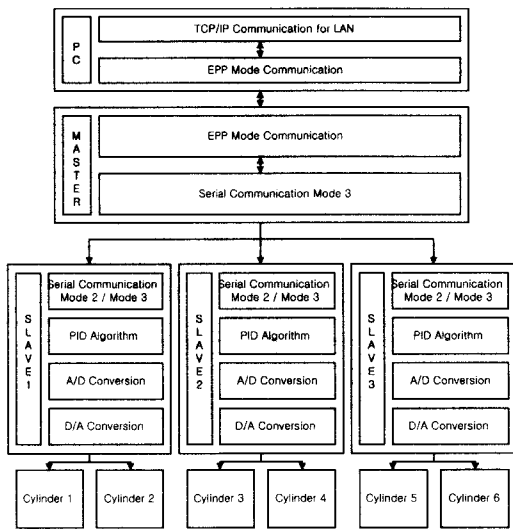


Fig. 9 Structure of communication network

터 프로세서 통신을 이용하여 마스터와 슬레이브간의 통신을 폴링방식으로 사용하여 데이터 충돌을 방지하며 각각이 독립된 제어를 수행할 수 있도록 하였다.

제어기 내에서의 통신은 비동기 멀티 시리얼 통신을 하고 있으며 마스터 제어기와 슬레이브 1, 2, 3 간의 통신을 하고 있다. 동작은 마스터제어기의 프로세서는 수신할 때마다 인터럽트를 발생하도록 모드 3으로 동작을 하고 있으며 슬레이브 제어기의 프로세서는 어드레스를 수신할 때에는 모드 2로서 그 어드레스가 자신의 어드레스이면 모드 3으로 변환하여 데이터를 받는 폴링 방식으로 통신을 하고 있다. 이러한 통신의 장점으로서 통신프로그램

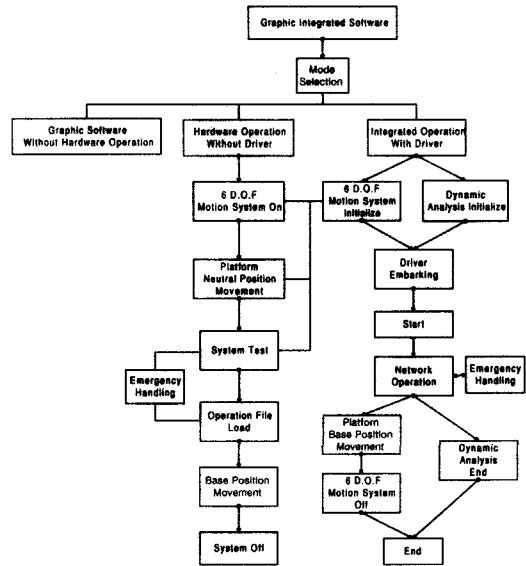


Fig. 10 Tree of integration program

을 작성하기 위한 방법의 용이성과 알고리즘 및 그 밖의 제어에 관련된 프로그램의 변동시 용이하게 작성할 수 있다는 장점을 가지고 있다.

### 3. 통합 프로그램

주행환경 및 굴삭기의 그래픽 처리 뿐만 아니라 시뮬레이터를 운영하기 위한 통합 프로그램은 사용자에게 최적의 환경을 제공하기 위해 Fig. 10과 같이 3가지 형태로 작동된다. 즉, 하드웨어 작동이 없이 단지 그래픽 소프트웨어 상에서 굴삭기의 주행 시뮬레이션을 실행하는 모듈, 운전자 없이 하드웨어의 작동을 테스트하기 위한 모듈, 그리고 운전자가 탑승되어 하드웨어 작동시 시스템을 운영하는 통합 환경 모듈로 구성되어 있다. 사용자는 안정성을 위해 먼저 하드웨어의 성능을 테스트한 뒤 운전자가 탑승하는 통합 작동 형태를 사용하여야 한다.

통합 프로그램은 모티프(Motif)를 이용하여 윈도우 기반으로 작성하였는데 윈도우는 사용자가 키보드를 사용하지 않더라도 마우스를 이용하여 아이콘이나 메뉴를 조작함으로써 간단하게 프로그램을 수행할 수 있다. 모티프는 X-윈도우 시스템에서 구동되는 다양한 기능이 함수 형식으로 구성된 객체지향적 프로그램 라이브러리며 하드웨어 환경에서도 호환성이 뛰어나다. 모티프는 ANSI C 언어 및



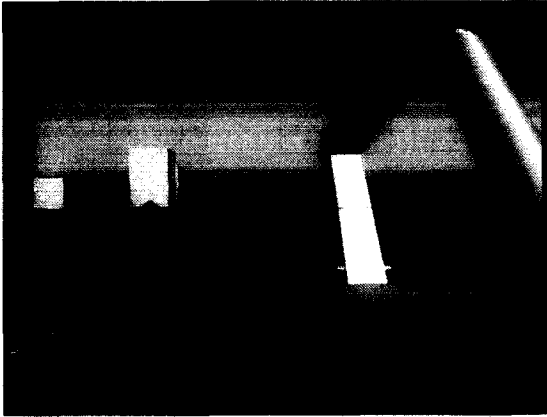


Fig. 11 Screen of driving environment

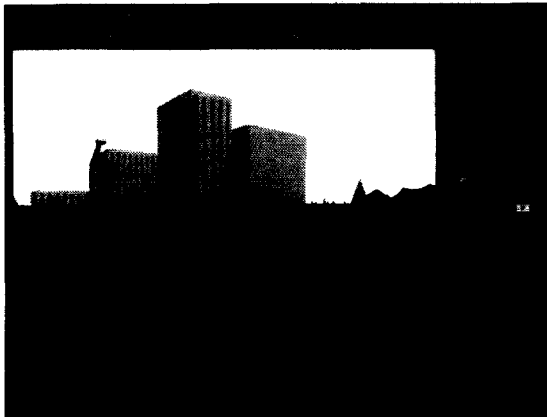


Fig. 12 Screen for driver interface

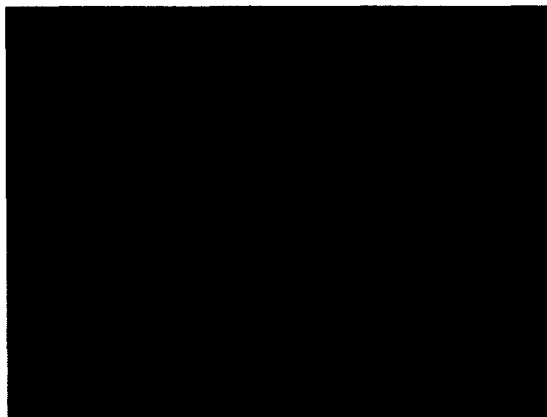


Fig. 13 Setting of PID gains and cylinder strokes

X, Xt 라이브러리 등과 호환이 가능하며 그래픽은 모티프 윈도우에 드로잉 영역을 생성하여 OpenGL을 사용하여 나타낼 수 있다.

### 3.1 하드웨어의 작동이 없는 그래픽 시뮬레이션

이 형태는 하드웨어와 연결하지 않고 소프트웨어 상에서 시뮬레이션을 수행한다. 기본적으로 두 개의 윈도우가 생성되는데 Fig. 11과 같은 사용자 인터페이스용과 Fig. 12와 같은 주행 환경을 나타내는 윈도우로 구성된다. 동역학 해석된 결과를 그래픽으로 시뮬레이션 해 볼 수 있다.

### 3.2 운전자가 없는 하드웨어 동작 구현

이 형태는 그래픽 컴퓨터에서 스텐더 플랫폼을 구동하고 성능을 테스트하는데 사용된다. 스텐더 플랫폼을 구동하면서 해석된 스텐더 플랫폼의 자세를 그래픽으로 보여주는 하나의 윈도우만 생성된다. 안정성을 위해 운전자 없이 그래픽 컴퓨터에서 Fig. 13과 같이 PID 게인 값과 실린더의 길이 값을 설정하고 제어하여 하드웨어의 성능을 시험한다. PID 게인 값과 실린더 길이 값을 텍스트 형태로 입력 할 수 있고 파일의 형태로 불러 올 수 있다. 그리고 사용의 효율성을 위해 모든 텍스트를 동시에 지울 수 있는 버튼과, 게인 값과 실린더 길이 값을 쉽게 편집할 수 있도록 텍스트 에디터를 실행시키는 버튼을 만들었다.

### 3.3 통합 환경 운영 형태

이 모듈은 그래픽 컴퓨터, 동역학 해석 컴퓨터, 그리고 하드웨어의 네트워크 동작을 관리한다. 이 형태는 운전자의 시점에서 보는 윈도우와 통합 환경 운영을 위한 윈도우로 구성된다. 통합 환경 운영을 위한 윈도우는 메모리 생성 또는 해제를 통해 여러 가지 기능의 위젯(widget)이 교체된다. 사용자의 선택에 따라 텍스트 위젯이 생성될 수 있고 그래픽 위젯이 생성될 수도 있다.

동역학 해석 결과로부터 굴삭기 중심의 6개의 좌표값 및 스텐더 플랫폼의 자세를 나타내는 6개의 좌표값이 산출된다. 그리고 역기구학 계산을 통해 6개의 실린더 길이 값이 계산된다. 그래픽 컴퓨터에서 동역학 해석 컴퓨터에서 계산된 굴삭기의 중심위치와 스텐더 플랫폼의 6개의 좌표값을 가지고 굴삭기의 움직임과 스텐더 플랫폼의 자세를

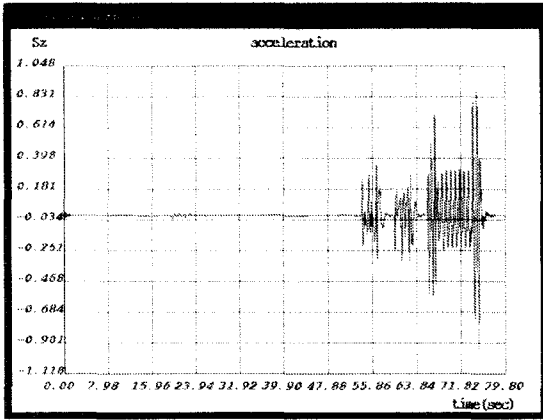


Fig. 14 Displacement of z-direction

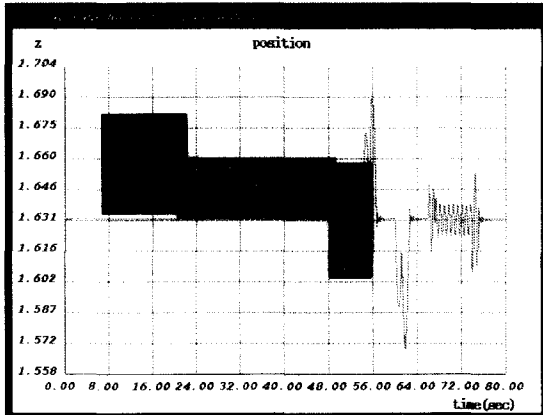


Fig. 15 Menu for graph setting

그래픽 처리한다.

굴삭기의 운동을 플랫폼의 제한된 작동 범위 내에서 재현하기 위해서 화면과 스튜어트 플랫폼의 연동이 필요하다. 이를 위해 워시아웃(washout) 알고리즘을 이용하여 주행 가속도는 플랫폼에서 담당토록 하였다. 그리고 주행 속도는 그래픽적으로 탑승자에게 제공하도록 처리하였는데, 그래픽 컴퓨터에서 액츄에이터의 실린더 값을 제어기로 보내고 같은 시간에 해석된 굴삭기의 중심 위치에 따라 한 프레임에 해당하는 주행 환경을 그리도록 하였다.

### 3.4 시뮬레이션 결과 처리

시뮬레이션 종료 후에는 시뮬레이션 결과를 그래프로 표시할 수 있다. 시뮬레이션 수행중의 동역학 해석 결과는 파일로 저장하고, 저장된 파일을 이용하여 결과 검토에 필요한 여러 값들을 그래프로 표시한다. Fig. 14는 해석된 스튜어트 플랫폼의 z축 변위 결과를 도식적으로 나타낸 그래프이다. Fig. 15는 결과 그래프에서 팝업 메뉴를 사용하는 방법을 보여 준다. 결과 그래프는 오른쪽 마우스 버튼에 의해 활성화되는 팝업 메뉴를 이용해 필요한 사항을 설정하거나, 보기를 원하는 축 방향을 선택할 수 있다. 원하는 축 방향을 선택한 후에는 위치, 속도, 가속도 중에 원하는 사항을 선택하면 자동으로 최대·최소 값을 찾아서 적당한 크기로 도시하여 준다. 굴삭기와 스튜어트 플랫폼의 중심의 6개 좌표의 위치, 속도, 가속도를 볼 수 있으며 기준 실린더 입력과 제어된 실린더 길이 값의 오차를 그래프를 통해 볼 수 있다.

## 4. 결론

건설 중장비용 시뮬레이터를 개발하면서 얻은 연구 결과를 통해 다음과 같은 결론을 내릴 수 있었다.

(1) 굴삭기 모델을 단순화시키고, 속도변환법을 사용함으로써 범용 동역학 프로그램에 비해 해석 시간이 1/10로 단축되는 동역학 프로그램을 개발하였다.

(2) MIL-STD-1588에서 제시한 운동사양을 기본으로 시뮬레이터의 운동사양을 설정하였고, 이를 바탕으로 500 kg의 부하를 처리할 수 있는 유압 유니트를 설계하였다.

(3) 실시간으로 시뮬레이터가 구동될 수 있도록 통신은 그래픽 컴퓨터와 동역학 해석 컴퓨터 및 PC 사이를 LAN으로 구성하였으며, PC와 제어기간의 통신은 EPP 모드를 이용한 양방향 병렬 방식으로 구현하였다.

(4) 그래픽은 셰이딩 모델, 안티 알리에이징, 원근 효과 및 음영 처리를 통하여 대상을 현실감있게 나타내었으며, 그래픽 처리의 속도 향상을 위하여 컴퓨터 메모리의 효율적인 사용, 2차원과 3차원 모델의 혼합 사용 및 기준위치 검색을 통한 대상 제거 알고리즘을 적용하였다

(5) 통합 프로그램은 하드웨어 작동이 없이 단지 그래픽 소프트웨어 상에서만 실행하는 모듈, 하

드웨어의 작동판 평가하는 모듈, 그리고 운전자가 탑승하여 전체 시스템을 구동하는 모듈 등 3가지의 운영 형태로 작동되도록 구성하였다.

현재 운전자가 시뮬레이터에 탑승할 때 주행감을 충실하게 제공하기 위하여 감성공학적 접근을 통한 속도감 평가 시험을 시도하고 있다. 이 실험을 통해 재현되는 운동이 운전자에게 만족스러운 현실감을 주는지를 평가하게 되며, 측정 결과를 토대로 시뮬레이터의 구성 요소와 인간의 감성 어휘 사이의 상관관계를 분석함으로써 시뮬레이터의 현실성을 증대시키고자 한다. 향후 보다 효율적인 시뮬레이터를 개발하기 위하여 주행뿐만 아니라 각종 작업상황도 실시간으로 해석 가능한 차량 해석 프로그램 및 실시간 처리 그래픽 모델의 개발, 운동 재현기의 강인한 제어 알고리즘의 개발 등의 연구가 필요하다.

## 후 기

본 연구는 한국과학재단 특정기초연구과제(97-0200-1001-5)의 지원금으로 수행되었으며 이에 관계자 여러분께 감사드립니다.

## 참고문헌

1. Käding, W. and Hoffmeyer, F., "The Advanced Daimler-Benz Driving Simulator," SAE Paper 950175, 1995.
2. Bartollini, B., "The General Motors Driving Simulator," SAE Paper 940176, 1994.
3. Greenberg, J. A. and Park, T. J., "The Ford Driving Simulator," SAE Paper 940179, 1994.
4. Repa, B. and Wierwille, W., "Driver Performance in Controlling a Driving Simulator with Varying Vehicle Response Characteristics," SAE Paper 760779, 1976.
5. Freeman, J. S., "The Iowa Driving Simulator: An Implementation and Application Overview," SAE Paper 950174, 1995.
6. 김영수, 백창흠, 최성학, 심민식, 최진성, 강득수, 정대현, "비행훈련 시뮬레이터 운동판," 대우중공업 기술개발 소식, 1995.
7. 이운성, 조준희, 김정하, "안전도 향상을 위한 Driving Simulator의 개발 및 응용 동향," 한국자동차공학회 1997년도 동력전달계·차량운동성능·Tire운동특성·Safety부문 학술장연 초록집, pp. 18-27, 1997.
8. 송병준, 양경덕, 배대성, 송창섭, 조성현, 김성규, "차량구동용 시뮬레이터의 설계 및 제작," 한국자동차공학회 1994년도 춘계학술대회 논문집, pp. 285-292, 1994.
9. 강지운, 김동환, 이교일, "6자유도 운동재현용 베드의 순기구학적 추정기 설계," 한국정밀공학회 96년도 춘계학술대회 논문집, pp. 483-487, 1996.
10. Kim, S. S. and Vanderploeg, M. J., A State Space Formulation for Multibody Dynamic Systems Subject to Control, The University of Iowa Technical Report, No. 84-20, 1984.
11. Nikravesh, P. E. Computer-Aided Analysis of Mechanical Systems, Prentice-Hall, 1988.
12. 유완석, 최대형, 김광석, 정만식, "실시간 시뮬레이션을 위한 휠 굴삭기의 다물체 동역학 모델링," '98 한국국자동제어학회 논문집, pp. 1763-1766, 1998.
13. Six Degree-of-Freedom Motion System Requirements for Aircrewmember Training Simulators, MIL-STD-1558, Department of Defense, 1974.
14. Pennock, G. R., and Kassner, D. J., "The Workspace of a General Geometry Planar Three-Degree-of-Freedom Platform-Type Manipulator," ASME J. of Mechanical Design, Vol. 115, pp. 269-276, 1993.
15. 손권, 권성수, 권병희, "굴삭기 시뮬레이터의 운동시스템," 대한기계학회 1998년도 춘계학술대회 논문집, pp. 554-559, 1998.