

6-DOF 시뮬레이터의 Real-time 시스템 제어에 관한 연구 (Real-time System Control for the 6-DOF Simulator)

김 영대 김 충영 백 인철 민 성기
(Y. D. Kim, C. Y. Kim, I. C. Paek, S. K. Min)

대전 기계창 (Agency for Defence Development)

6-DOF simulator system is designed to real-time processing for motion control, data acquisition, image generation and image processing etc.. In this paper, we introduce hardware and software design technologies for distributed processing, event-trapping, system monitoring and time scheduling procedure in 6-DOF simulator system design.

1. 서론

6-DOF 운동 시뮬레이터는 지금까지 주로 항공기 조종 시뮬레이터로서 항공기의 운동과 유사한 가속 및 중력장을 탑승자가 느낄 수 있도록 함으로써 실제 비행 상황을 재현하도록 설계 및 발전되어 오고 있다.[1][2]

6-DOF 시뮬레이터는 Motion 제어, Data 획득, 그리고 영상 발생 및 처리 등의 기능을 Real-time에 수행하도록 설계되어져 있다. 전체 시스템의 기능은 6 개의 모듈화된 Sub-system들로 구성되고 각 Sub-system은 전체 시스템 내에서 고유의 기능을 수행하면서 다른 Sub-system과는 LAN(Local Area Network)을 통하여 정보 교환이 이루어 진다.[4]

일반적으로, Real-time 운영체제로서 VAX VMS 등 몇몇 운영체제가 알려져 있는 반면 UNIX는 기본적으로 Time-sharing 대화식 운영체제로서 Real-time 응용에는 부적합하다고 인식되어 있다. 그러나 Real-time 측면에서 UNIX를 자세히 관찰해 보면 Real-time에 필요한 대부분의 기능들을 가지고 있으며 더우기 근래에는 UNIX의 Real-time 적용에 많은 관심들이 집중되고 있고 몇몇 Real-time UNIX Version이 나오고 있다.[3] 6-DOF 시뮬레이터 시스템에서는 전체 시스템을 조정 및 제어하는 Host Computer의 운영체제로서 UNIX(ULTRIX-32)를

사용하였고, 또한 전체 시스템의 기능들을 Sub-system별로 분산 처리하게 하여 각 Sub-system들 간의 인터페이스 혹은 통신을 최소화함으로써 LAN을 사용한 시스템의 통합에 있어서 통신을 위한 시간 지연을 최소화할 수 있으므로 전체 시스템의 Real-time 처리를 실현할 수 있게 된다.

본 논문에서는 Real-time 6-DOF 시스템을 설계하는데 있어서의 분산처리 기법과 Event-trapping, System Monitoring 및 Time Scheduling 과정의 시뮬레이터 적용 기술에 대한 H/W 및 S/W 설계 기법에 대해 연구 결과를 보여준다.

2. 6-DOF 시뮬레이터 시스템 구성

6-DOF 시뮬레이터는 자체의 운동을 시뮬레이션 하기 위한 6 개의 유압 실린더를 사용한 6 자유도 Motion의 운동을 Display 하고 차량의 동작 상태를 점검할 수 있는 영상처리 시스템, 시험 Data를 수집 분석하는 Data 획득 시스템, 그리고 모든 시험 상태를 조정하고 Monitoring하는 컴퓨터 제어 시스템 등으로 구성되며 전체적인 블록 선도를 그림 1.에 나타내었다. 6-DOF 시뮬레이터 시스템은 실제적으로 크게 6 개의 Sub-system으로 구성되어

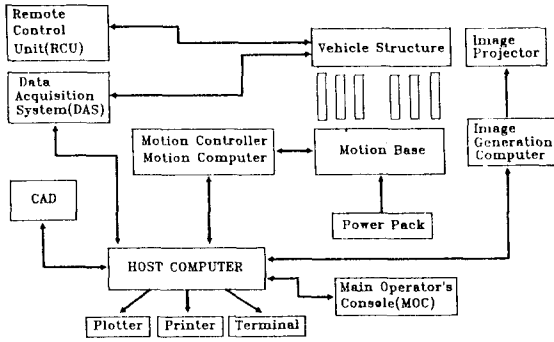


그림 1. 6-DOF 시뮬레이터의 개념도

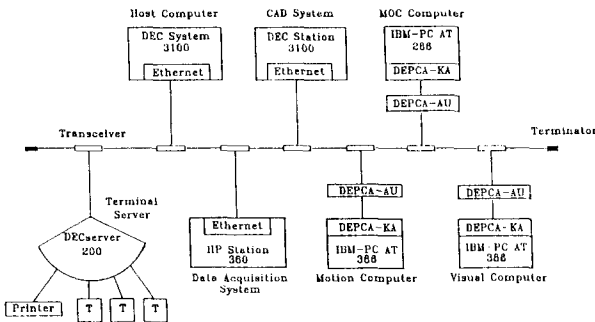


그림 2. 시스템 Configuration

있으며 각 Sub-system들은 그림 2.와 같이 Ethernet에 접속되어 상호 Data 교환이 이루어진다. 각 Sub-system들의 구성은 다음과 같다.

2.1. Motion System

6-DOF 운동을 할 수 있는 Motion 구조물 및 Power Pack 그리고 Motion 컴퓨터와 Motion 제어기로 구성되어 있으며, Motion 구조물 위에 장착된 차량에 실제 주행 상태와 동일한 환경을 만들어 준다.

2.2. Visual System

실제 차량에서 관측되는 것과 거의 동일한 영상을 실험실 내에서 제공하기 위한 영상 발생 및 처리 시스템으로서 영상 발생 컴퓨터, 영상 투사기(Projector), 대형 화면(Screen), Video 카메라, 영상처리 컴퓨터 등으로 구성된다.

2.3. Data Acquisition System

시험중 Motion System으로부터 발생하는 각종 Data를 수집하는 Data 획득 전용 주변장치 및 획득된 Data들을 처리 분석하는 컴퓨터로 구성되며 시험 단자들로부터 전기 신호를 받아 컴퓨터가 받아들일 수 있는 신호로 바꾸는 전 과정을 포함한다.

2.4. Main Operator's Console

시뮬레이터 운용시 각 Sub-system들의 상태를 Monitoring하고 시뮬레이터를 수동으로 조작할 수 있도록 하며 위험 상황 발생시 Emergency Stop 기능을 수행하도록 설계되어 있다.

2.5. Host Computer System

각 Sub-system들을 제어하며 시험 Schedule 대로 전체 시험을 조정 및 감시하고 시험전 Sub-system 들의 초기 상태를 설정한다.

2.6. CAD System

CAD 전용 시스템으로서 개발 환경에서는 각종 설계 및 시뮬레이션을 위한 도구를 제공하며 시험시에는 획득한 Data와 시뮬레이션 결과와의 비교 분석 기능을 갖는다.

3. 시스템의 Real-time 제어

전체 시뮬레이터 시스템의 Real-time 제어는 각 Sub-system의 Real-time 제어 및 처리로 나누어 볼 수 있으며 각 Sub-system에 있어서의 Real-time 제어는 다음과 같다.

3.1. Motion System

Motion System은 6 개의 유압 실린더에 의해 Pitch, Roll, Yaw 3 방향의 각속도 운동과 Heave, Sway 그리고 Surge 3 방향의 병진 운동을 일으키도록 설계되었으며 운동의 명령은 Host Computer에 의해 주어지는 주행 시험 기록이나 Pre-programmed Schedule 혹은 Signal Generator에 의해 인가되는 Analog 명령에 의해 움직여 주며

이러한 Pitch, Roll, Yaw 및 Bounce(Heave) 운동에 대하여 각 실린더의 길이 L1 ... L6에 대한 명령으로 Real-time 으로 계산하여 D/A 포트를 통해 Servo 제어기로 보내게 된다. Real-time 제어를 위해 Intel 80386 CPU를 사용하는 Motion 제어기와 Analog I/O를 위한 A/D, D/A 인터페이스를 함께 사용한다. 이때 Motion 제어기의 처리 속도를 향상시키기 위해 Array Processor 보드를 함께 사용할 수도 있다. 이렇게 출력된 Analog Output 신호(L1..L6)는 Electro-hydraulic Servo 증폭기를 통해 전류 신호로 바뀌어져 유압 Jack을 구동한다. 또한 Motion Computer는 Motion 구조물의 상태를 수시로 점검하여 위험 상황 발생시 즉각 경고 신호를 발생시킨다.

3.2. Visual System

Visual System은 컴퓨터 터미널 화면을 통해 노면 상태를 만들어 주고 이 화면을 부사기를 통해 시뮬레이터 전면에 있는 대형 화면에 부사시켜 준다. 이때 노면의 영상은 차량에 장착된 카메라를 통해 관측되며 카메라의 영상은 Video 화면으로 Console 모니터에 보내준다. 모든 Video 화면은 real-time 영상처리에 의해 컴퓨터에 기록되어 차후 시험에 대한 분석 자료로 사용되고 필요하면 LAN을 통하여 Host Computer에 전송되어진다.

3.3. Data Acquisition System(DAS)

DAS는 모든 시험 수행중 발생하는 Data를 수집, 처리, 수록한다. A/D 변환 채널, D/A 변환 채널, D/I 및 D/O 채널을 겸비하고 있는 I/O 모듈을 HP Workstation이 제어하며 이 Workstation은 시험 Data에 대한 Real-time Local Processing에 의하여 분석 결과를 그래픽 모니터에 Display 하며 수집된 각종 Data와 정보들을 LAN을 통하여 Host Computer에 전송한다. Workstation의 운영체제로서는 UNIX의 Real-time Version인 'HP-UX'를 사용하였다. DAS S/W의 개략적인 구조를 그림 3.에 나타내었다.

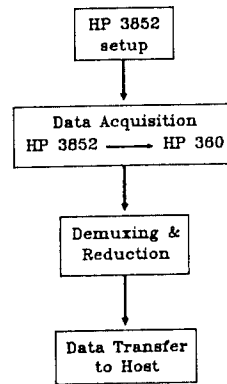


그림 3. DAS S/W 구조

3.4. Main Operator's Console

6-DOF 시뮬레이터 운용시 각 Sub-system의 상태를 Color CRT를 통하여 Monitoring 하는 장치로서 Monitoring은 각 Sub-system으로부터 MOC 컴퓨터에 입력되는 전기적인 신호가 Data Processing 되고, CRT에 각 기능에 따라 정해진 그래픽 형태로 Display 되는 정보를 운용자가 감시하게 된다.

MOC 컴퓨터는 A/D 변환기와 Digital Input 단자로 입력된 Data를 처리하고 Key Panel은 시뮬레이터 운용시 각 Sub-system의 Power와 모니터 상의 메뉴를 제어한다. 또한 MOC의 Key Panel에는 Emergency Stop 스위치가 있어서 위험 상태 발생시 전체 시스템의 동작을 멈출 수 있도록 되어있다.

3.5. Host Computer System

Host는 각종 시험 Schedule 대로 전체 시험을 조정 및 감시하며 각종 시험 Data를 분석 및 처리할 수 있는 S/W를 준비하여 시험 중이나 종료 후 필요한 분석을 수행하게 된다. 전체 시스템의 상태를 수시로 점검하고 비상상태 발생시 전체 시스템을 정지시키는 역할도 담당한다. Host Computer S/W 구조를 그림 4.에 나타내었다.

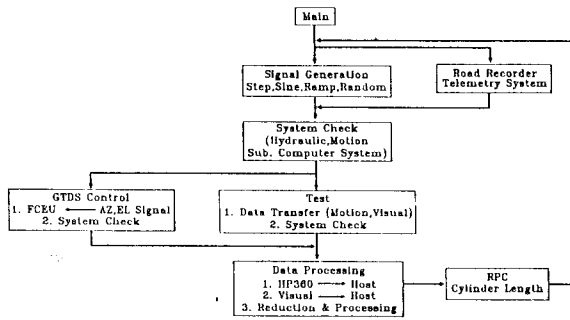


그림 4. Host Computer S/W 구조

3.6. CAD System

Workstation에 각종 CAD 시스템과 MDS 시스템 등을 적재하여 각종 제어기 보드 등의 설계 및 개발에 사용하며 설계시의 시뮬레이션 결과물 실제 시험에서 획득한 Data와 비교 분석을 할 수 있도록 한다.

4. 결과 및 검토

Real-time 환경에서는 시한성이 관건인 만큼 각 Sub-system 간의 동기화(Synchronization)가 요구된다. 따라서 각 Sub-system은 소정의 시간 내에서 필요한 처리를 완료해야 하고 그 결과를 다른 Sub-system으로 전송해야 한다.

6-DOF 시뮬레이터의 설계에 있어서 초기 단계에서는 Host Computer 하나로써 Sub-system Computer들의 기능들을 모두 수행하는 것이었으나 그렇게 하면 Real-time 처리를 위하여 Host Computer는 대용량, 고성능 Computer가 필요하게 되고 많은 비용을 Host Computer에 부자해야 하는 난점이 있었다. 이를 해결하는 방안으로서 LAN을 사용한 분산처리 시스템으로 구성하였고, 각 Sub-system에 장착된 Computer 혹은 주변장치들을 독립적으로 사용 가능함에 따라 모듈별 Testing, Local processing이 이루어짐으로써 자원을 효율적으로 이용할 수 있게 되었다. 비교적 저렴한 가격에 빠른 처리속도를 가진 RISC-based Computer를 Host로 선정하여 요구되는 성능을 얻을

수 있고 비용 대 효과면에서 뿐만아니라 장기적으로 볼때 시스템의 기능 확장이 용이하게 되어 전체 시스템의 부자 가치를 높이게 되었다. 또한 최근에 와서 급속히 여러 분야에 널리 도입되고 있으나 Real-time 응용에는 약한 것으로 인식되어 있는 UNIX 운영체제를 사용하여 시스템을 구성함으로써 UNIX 시스템의 Real-time 응용의 가능성을 보여주었다.

5. 결론

시뮬레이터 시스템에 있어서 각각의 Sub-system들이 서로 유기적인 관계를 유지하면서 하나의 큰 시스템을 구성하기 위해서는 전체 시스템의 Real-time 제어가 이루어져야 하며, 빠른 전송 속도를 얻을 수 있고 다양한 이기종 컴퓨터 간의 통신을 비교적 간편하게 실현할 수 있을 뿐만아니라 장기적인 안목으로 볼때 시스템의 확장이 용이한 LAN 시스템을 사용하여, 각 Sub-system 별 Real-time 처리에 바탕을 둔 분산처리 시스템으로써 전체 시스템의 Real-time 제어를 구현하였다. LAN을 설치함으로써 각 Sub-system들 간에 정보 교환 및 자원 공유의 효과도 아울러 얻을 수 있으므로 시스템 자원의 효율적 이용을 도모할 수 있게 되었다.

참고 문헌

- [1] C.J. Jansen, "Present and Future of the NLR Moving Base Research Flight Simulator", AIAA Flight Simulation Tech. Conf., Atlanta, Georgia, September 7-9 1988.
- [2] R. Levi and L. Hayashigawa, "Specification Consideration for Small Motion Base", AIAA Flight Simulation Tech. Conf., Atlanta, Georgia, September 7-9 1988.
- [3] Clement T. Cole and John Sundman, "UNIX in Real time: What it takes to make the grade", UNIX Review, 1985.

[4] Networks and Communications Buyer's Guide, pp.

1.1 - 2.77, digital, January-June 1989.