

다목적실용위성 2호 HIL 시뮬레이터 개발방법 연구

이상욱, 조성기*, 이호진, 이성팔

한국전자통신연구원 위성통신시스템연구부

1. 개요

다목적 실용위성 1호는 1999년 12월에 발사되어 3년의 수명기간동안 한반도의 지도제작, 해양관측 및 우주탐사에 활용되고 있다. 이러한 다목적 실용위성을 운용하기 위해서는 그림 1에서 표시된 바와 같이 위성운용시스템(SOS: atellite Operation Subsystem)[1], 임무분석 및 계획서브시스템(MAPS:Mission Analysis and Planning Subsystem)[2], 위성추적 원격측정 및 명령서브시스템(TTCS: Tracking Telemetry and Command Subsystem), 그리고 위성시뮬레이터(SIM : SIMulator)으로 구성된 위성 관제시스템이 필요하다. 관제시스템을 구성하는 한 서브시스템인 위성시뮬레이터를 한국전자통신연구원이 대우중공업과 함께 공동개발하였다[3]. 또한 한국전자통신연구원은 무궁화위성시뮬레이터를 실험실모델 무궁화위성관제시스템 개발의 일부로 개발한 바 있다.[4]

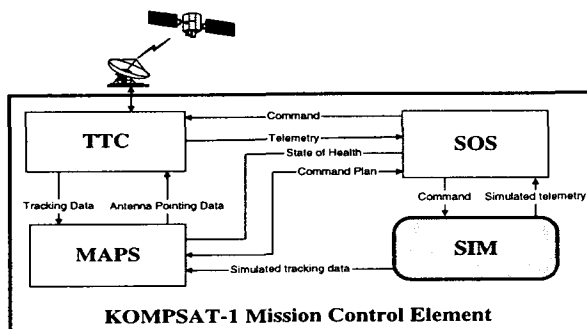


그림 1. 다목적실용위성 관제시스템 기능블록도

이러한 시뮬레이터는 관제시스템의 한 서브시스템으로 원격명령검증, 위성 운용자 교육, 원격명령절차확인 및 검증, 새로운 제어로직의 검증, 위성이 발사되기 전까지 위성운용시스템의 검증 그리고 미리 정의된 이상상태에 대한 복구작업을 수행하는데 활용할 수 있는 소프트웨어 시스템이다.

본 논문에서는 다목적실용위성 1호 시뮬레이터의 구조 및 기능을 소개하고 다목적 실용위성 2호 위성시뮬레이터를 HIL(Hardware In the Loop)로 구현하기 위한 방법을 제안하고자 한다.

2. 다목적실용위성 1호 S/W 시뮬레이터

다목적실용위성 1호 시뮬레이터는 그림 2에 나타난 바와 같이 하드웨어적인 환경으로 HP J210 Workstation 에서 개발되었고 RS232C 로 직접 연결된 PC에 World Tool Kit이라는 VR Tool을 사용하여 실시간으로 위성의 운동상태를 표시할 수 있도록 하였다. 관제시스템의 다른 서브시스템과는 LAN을 이용하여 접속하도록 하였다. 소프트웨어적인 환경으로는 HP-UX 10.X 운영체제에서 구현했으며 사용언어는 ANSI C이며 다중프로세스구조로 되어 있다.

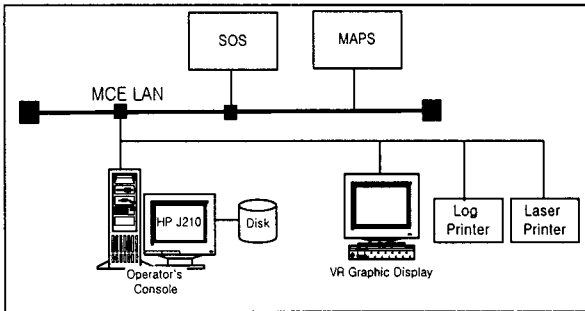


그림 2. 다목적실용위성시물레이터 H/W 구성도

그림 3 은 위성시물레이터의 소프트웨어적인 Hierarchy 를 보여주고 있다. 상위 Level 은 기능구조로 Block 으로 구분하였고 구현상의 독립프로세스인 Task 는 두번째 Level 로 구성하고 Function 의 모임으로 구성된 독립적인 기능의 Unit 으로 표시하였다.

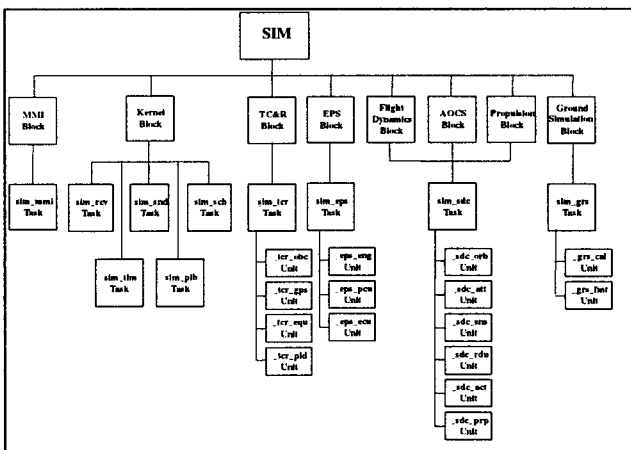


그림 3. 다목적실용위성 시물레이터 S/W Hierarchy

그림 4 는 위성시물레이터의 기능을 데이터 흐름을 통하여 기술하고 있다. 위성시물레이터는 위성의 기능과 유사하게 SOS 로부터 원격명령 (TC:TeleCommand) 를 수신하여 이를 처리하고 이에 상응하는 시물레이션 반응을 원격측정

(TM:TeleMetry) 형태로 SOS 에 출력하게 된다. 이와 유사한 기능은 시물레이터의 자체 GUI(Graphic User Interface)를 통하여 수행할 수도 있다. 다목적 실용위성의 정밀한 위성체 각 서브시스템 모델, 우주환경 모델 및 비행역학모델이 필요하며 특히 위성체 개발회사로부터 제공되는 위성체탑재 비행 소프트웨어(FSW:Flight SoftWare)[5] 는 시물레이터 개발환경에 이식작업을 거쳐 포함시켰다. 즉, 그림 3 과 4 에서 tcr(tracking commanding & ranging), sdc(spacecraft dynamics & control), eps(electric power subsystem) Task 는 각각 위성체의 세 프로세서, OBC(OnBoard Computer), RDU(Remote Drive Unit), ECU(Electric Control Unit)에 탑재된 FSW 를 포함하고 있다.

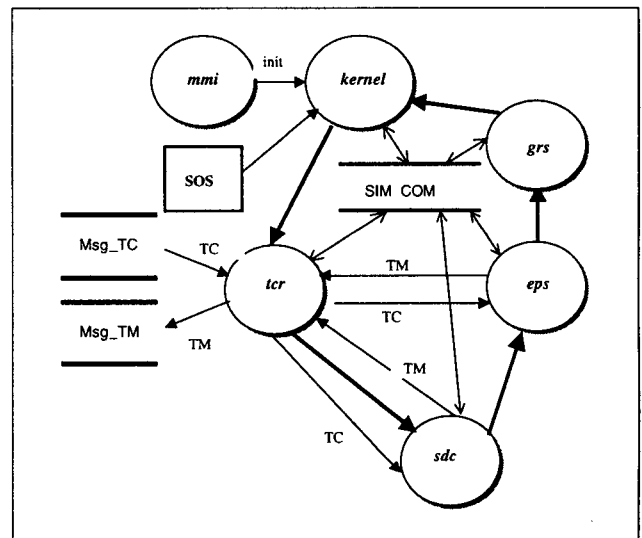


그림 4. 다목적실용위성시물레이터 데이터흐름도

데이터의 흐름상에서 시물레이터의 구동과 시물레이터의 스케줄링, 데이터처리 및 통신은 knl(kernel)에서 수행하고 수신된 TC 는 tcr 의 OBC 에서 sdc(satellite dynamics and control)의 RDU 및

eps(electric power subsystem)의 ECU 에 해당 명령을 분류하여 1553B 를 Emulation 하는 Shared Memory 를 통하여 분배한다[6]. 시뮬레이션에 의해 생성된 TM 은 각각의 비행소프트웨어에서 수집하여 Packet 화하여 TC 처리의 역순으로 OBC 에서 모은 후, knl 을 통하여 SOS 에 전송한다[7]. 각각의 프로세스간의 통신은 Shared Memory 와 Message Queue 를 이용하였다. 위성서브시스템 모델과 FSW 간의 인터페이스는 참고문헌[8]에 상세히 기술되어 있다.

3. 다목적실용위성 2 호 HIL 시뮬레이터

3.1 HIL 시뮬레이터의 구성

이미 기술된 바와 같이 다목적실용위성 1 호 시뮬레이터는 HP Workstation 의 HPUX 상에서 구현된 순수 소프트웨어로서이며 구현시 FSW 를 이식하는데 여러가지 문제점, 즉 위성의 80186 온보드 프로세서 및 VRTX OS 와 HP Workstation 및 HPUX 의 차이로 인한 데이터 Type 및 Swap 등의 문제로 어려움을 겪게 되었다. 또한 개발과정에서 FSW 의 변경에 대한 대응에도 상당한 노력을 필요로 하였다. 이러한 제약사항을 극복하기 위해 다목적실용위성 2 호의 시뮬레이터를 HIL(Hardware In the Loop) 로 개발하는 것을 고려하게 했다.

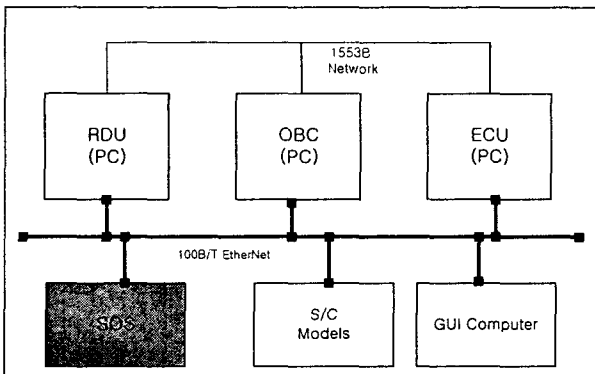


그림 5. 제안시스템의 구성도

즉, 온보드 프로세서와 1553B 을 Hardware 로 구성하고 인터페이스 및 위성체의 각종 하드웨어는 소프트웨어 모델로 구성하는 HIL 시뮬레이터를 제안하고자 한다.

그림 5 는 제안시스템의 구성도이다. 위성 온보드 프로세서 역할을 하는 3 대의 PC 에 VRTX RTOS 및 FSW 를 탑재시켜 1553B Network 으로 위성의 온보드 컴퓨터 환경을 구성하고 이를 시뮬레이터 GUI, 비행역학, S/C Subsystem 모델을 탑재한 컴퓨터 및 SOS 와 100B/T LAN 을 통하여 인터페이스를 갖도록 구성한다. 표 1 은 HIL 의 구성 사양을 나타내고 있다.

표 1. HIL 시뮬레이터의 사양

구성요소	사양
PC	CPU : Intel Pentium III 600MHz 이상 Memory : 128MB 이상 Monitor : 19" HDD : 20GB LAN Card : 100/BT BUS : PCI & ISA
LAN	100B/T, HUB
1553B	MIL-STD-1553
RTOS	VRTX x86

3.2 HIL 시뮬레이터의 구현방법

시스템 구성

그림 6 은 제안시스템의 데이터 흐름관점에서의 구조도이다. 실시간운영체계인 VRTX 를 탑재한 상용 PC 3 대를 1553B 데이터 버스를 통하여 인터페

이스를 구성한다. 이러한 3 대의 컴퓨터는 위성의 온보드 컴퓨터에 탑재되는 OBC, RDU, ECU 의 FSW 를 각각 탑재시켜 위성 탑재 컴퓨터의 기능을 수행하도록 한다. 이렇게 위성의 프로세서인 Intel 계열의 마이크로프로세서를 사용하여 구성함으로써 기존 시물레이터에서 발생했던 데이터의 Type 및 Swap 문제를 해결할 수 있다. 이 컴퓨터는 100B/T 고속 LAN 을 통하여 위성비행역학을 포함한 S/C(SpaceCraft) 모델과 위성의 운동상태를 디스플레이하는 VR 을 포함하는 컴퓨터, 시물레이터를 구동시키고 TC 을 생성, 전송하고 TM 을 수신하여 처리하고 분석하는 도구를 제공하는 GUI 컴퓨터와 LAN 을 통하여 인터페이스를 갖는다.

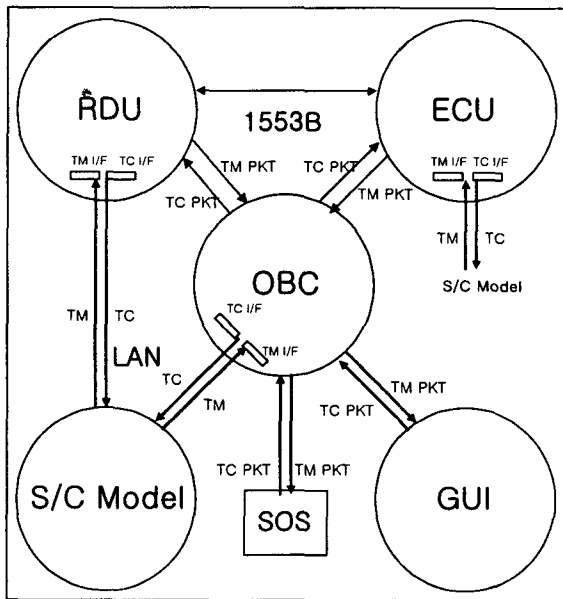


그림 6. 제안시스템의 데이터 흐름도

TC 처리

위성의 주어진 임무를 수행하기 위해서는 실시간 명령(RTC:Real-Time Command), 절대시간명령

(ATC:Absolute Time Command), 및 일련의 상대시간 명령(RTCS:Relative Time Command Sequence)를 이용하여 위성을 운용하게 되는데 일반적으로 이러한 명령은 SOS 에서 생성하여 Packet 으로 구성되어 일정한 형태, 예컨대 CCSDS Format 과 같은 형태로 생성되어 위성에 전송하게 된다. 제안 시물레이터에서는 이러한 Packet 형태의 TC 를 SOS 혹은 시물레이터 GUI 컴퓨터로부터 수신하게 된다. 이러한 TC Packet 은 OBC 에서 Validation 과정을 거친 후, RDU 및 ECU 의 TC Packet 을 분리하여 각각에 1553B 를 통하여 전송한다. 이렇게 전송된 TC Packet 은 종류에 따라 RTC 는 즉시 수행되고 ATC 는 해당시간에 RTCS 는 Trigger 명령이 수행되고 해당 상대시간이 경과하면 RTC 로 전환되어 수행된다.

TM 처리

위성의 OBC, RDU, ECU 에는 각각 TM 데이터를 수집하여 이를 Packet 화 하여 RDU 및 ECU 에서 생성된 Packet 은 1553B 를 통하여 OBC 에 수집되어 CCSDS Format 으로 SOS 혹은 GUI 컴퓨터에 전송된다.

S/C Model

위성체 모델은 우주환경 및 위성의 비행역학 모델 FDS(Flight Dynamics Subsystem), 위성의 자세 및 궤도제어를 위한 각종센서 및 추진시스템을 포함하는 각종 구동기 모델을 포함하는 AOCS (Attitude Orbit Control Subsystem), 전력계 모델을 포함하는 EPS(Electric Power Subsystem), 열제어시스템 모델의 TCS(Thermal Control Subsystem) 위성의

TC&R(Telemetry Commanding & Ranging) 하드웨어 모델의 TC&R 그리고 위성의 탑재체인 MSC(Multi-Spectral Camera) 모델의 PLD 로 구성 한다. 이러한 위성체 서브시스템은 TC 와 제어로직에 의해 생성된 명령을 OBC RDU ECU 로부터 수신하여 동작하고 그 결과를 OBC RDU ECU 로 전송되어 제어로직의 입력데이터 및 TM Packet 작성에 사용된다.

S/C Model 과 OBC RDU ECU 간의 인터페이스

실제위성에서는 위성의 세 프로세서와 위성체 각 서브시스템과 하드웨어적으로 인터페이스를 가진다. 제안시스템에서는 이러한 하드웨어적인 인터페이스를 소프트웨어적으로 구성하고자 한다. 먼저 TC 및 제어로직에서 생성된 명령을 위한 인터페이스는 FSW 에서 위성체의 각종 하드웨어에 명령을 전송할 때, `outp(address/channel, command)` 및 `outpw(address/channel, command)`함수를 사용한다. 즉, 해당 인터페이스의 Address 에 해당 명령을 전달하는 함수이다. 제안시스템의 경우에는 위 함수가 호출되면 이를 Ethernet LAN 을 통하여 S/C 모델 컴퓨터로 전송하고 이를 수신한 S/C 모델 컴퓨터에서 해당 Address 및 Channel 에 대응하는 명령 상태변수 및 Argument 를 변경시켜준다. 여기서 명령 Patch 처리루틴과 명령상태변수와 명령인터페이스 Address 및 Channel 간의 대응 Table 관리를 S/C 모델에서 수행한다. TM 의 경우에는, 시물레이션된 데이터를 매 1/4, 1, 16 초마다 수집하여 제어로직의 입력데이터로 사용하거나 매초마다 TM Packet 을 생성하는데 사용한다. 시물레이션데이터를 위한 인터페이스는 FSW 에서 위성체의 각종 하드웨어에 동작상태 및 운용결과를 수신할 때,

`inp(address/channel, command)` 및 `inpw(address/channel, command)`함수를 사용한다. 시물레이션 데이터를 수집주기에 따라 Packet 형태로 만들어 매초마다 Ethernet LAN 을 통하여 해당 프로세서로 전송한다. 각각의 프로세서에서는 해당 데이터수집 인터페이스의 Address 및 Channel 과 S/C 모델에서 수신된 Data Packet 의 상대위치와의 대응 Table 에 의해 데이터수집이 가능하도록 함수를 재지정하면 된다. 이러한 대응 Table 은 TC 와는 달리 각각의 프로세서에서 유지관리 하도록 한다. 또한 `inp, inpw, outp,` 및 `outpw` 등의 함수는 지금까지 기술한 내용을 수행하는 기능을 하는 Function Overloading 을 사용하면 FSW 의 변경을 최소화하여 개발할 수 있고 계속되는 FSW 의 수정 및 관리에도 쉽게 대응할 수 있다.

4. 결론

다목적실용위성 1 호 시물레이터의 시스템 소개와 개발과정에서 제기된 기술적인 문제점을 극복하기 위해 HIL 시물레이터를 제안하였다. 제안시스템은 일반적인 의미의 H/W 인터페이스를 포함하는 HIL 시스템은 아니지만 S/W 만으로 개발된 위성시물레이터의 한계를 극복하고 위성의 FSW 를 크게 변경하지 않고 이식할 수 있고 CPU 의 차이로 인한 데이터 Type 및 swap 문제를 해결할 수 있는 장점을 가진다.

참고문헌

- [1] Hee-Sook Mo, Ho-Jin Lee, and Seong-Pal Lee, "Development and Testing of Satellite Operation System for Korea Multipurpose Satellite-I", ETRI

- Journal, Vol. 22, No. 1, March 2000.
- [2] C. H. Won, J. S. Lee, B. S. Lee, and J. W. Eun,"
Mission Analysis and Planning System for Korea
Multipurpose Satellite-I," ETRI Journal, Vol. 21, No.
3, September 1999,
- [3] 최완식, 이상욱, 김신희, 최한준, 채동석, "다목
적실용위성 시뮬레이터 개발," 항공우주학회 추
계학술대회 초록집, pp. 572-575, 1998.
- [4] J. Y. Kang, J. M. Kim, and S. J. Chung,"Design and
Development of an Advanced Real-Time Satellite
Simulator,"ETRI Journal, Vol. 17, No. 3, October
1995
- [5] TRW Space & Electronics Group, KOMPSAT
Software Critical Design Audit, 1996.
- [6] TRW Space & Electronics Group, KOMPSAT
Command Allocation Document, 1997.
- [7] TRW Space & Electronics Group, KOMPSAT
Telemetry Allocation Document, 1997.
- [8] 이상욱, 최완식, 채동석, "다목적 실용위성 시뮬
레이터상의 Flight Software 와 위성 서브시스템
모델간의 인터페이스 구현," 항공우주학회지,
Vol. 27, No. 3, pp. 145-150, 1999.