技術論文

정지궤도위성 실시간 동역학 시뮬레이터 개발 및 연동시험을 통한 검증

박영웅*, 구철회**, 박봉규**, 구자춘*, 최재동*

Development of VDS for Geosynchronous Satellite and Verification using PILS & HILS

Youngwoong Park*, Cheolhea Koo**, Bongkyu Park*, Jachun Koo* and Jaedong Choi*

ABSTRACT

In this paper, VDS(Vehicle Dynamics Simulator) and ACS(Attitude Control Simulator) are developed and are verified using PILS(Process In-the Loop Simulation) between VDS and ACS. VDS is including the AOCS(Attitude & Orbit Control Subsystem) hardware modeling of geosynchronous satellite and consists of modulation concept. ACS performs the attitude determination using sensor data and generates the attitude control commands. In order to transfer the data between VDS and PCDU(Power Control & Distribution Unit), data acquisition boards were mounted. VDS performance is verified using HILS(Hardware In-the Loop Simulation) between VDS and PCDU.

초 록

본 논문에서 VDS(동역학 시뮬레이터)와 ACS(자세제어 시뮬레이터)를 개발하고 VDS를 ACS와 연동한 PILS를 통해 검증하였다. VDS는 정지궤도위성의 자세제어계 하드웨어 특성을 고려한 모델링 및 모듈화로 구성하였다. ACS는 센서 데이터로부터 자세결정을 수행한 후 구동기에 적절한 명령을 생성하는 역할을 한다. 또한, VDS와 PCDU(전력제어분배장치) 하드웨어와의 데이터 송수신을 위해 데이터획득보드를 장착하였다. VDS와 PCDU와의 HILS 연동시험을 통해서도 VDS의 성능검증을 수행하였다.

Key Words : AOCS(자세제어계), VDS(동역학 시뮬레이터), ACS(자세제어 시뮬레이터), Geosynchronous Satellite(정지궤도위성), PILS(프로세서 연동시험), HILS(하드웨어 연동시험), Verification(검증), PCDU(전력제어분배장치)

Ⅰ. 서 론

인공위성의 개발 단계에서 설계된 시스템의 성능평가는 필수적인 사항이며, 특히 heritage를 갖지 못한 새로운 제어 시스템 및 단품(부품 포함)을 설계하거나 개발하는 경우에 있어서 실제 위성체에 탑재하기까지는 여러 단계의 성능시험 및 평가를 요한다. 자세제어계에서의 성능시험으로 가장 보편적인 방법인 엔지니어링 시뮬레이션이 있는데 이 방법은 한 대의 컴퓨터를 이용해서 성능을 검증하는 것인데 이 방법은 실제 하드웨어의 특성 및 실시간 처리에 따른 시간 동기화 문제를 충분히 고려하지 못하는 단점이 있다.

이에 반해 보다 복잡한 방법으로 별도의 프로세서를 이용하여 위성체 모델을 처리하고 제어로직을 담당하는 부분을 분리하여 프로세싱 동기화 및 데이터 전송을 실제 시스템과 일치되게 시험을 수행하는 방법인 PILS(Processor In- the-Loop

† 2005년 3월 7일 접수 ~ 2005년 8월 22일 심사완료
* 정회원, 한국항공우주연구원 통신해양기상위성
연락저자, E-mail : ywpark@kari.re.kr
대전시 유성우체국 사서함 113호

Simulation) 방식이 있고, 실제 하드웨어를 일부 또는 전부와 연계해서 시험을 수행하는 방법인 HILS(Hardware In-the-Loop Simulation) 방식이 있다. 국내에서도 인공위성 자세제어계의 성능시험으로 PILS 및 HILS에 관한 연구를 진행하고 있다[1-4].

본 논문에서는 먼저 하나의 프로세서에 동역학 모델과 자세제어 모델을 같이 구현하는 엔지니어링 시뮬레이션을 통해 위성체 모델과 각종 자세제어계 구성품 모델 및 비행소프트웨어에서의 자세제어계 알고리즘 성능을 검증하고, 동역학 모델과 자세제어 모델을 별도의 프로세서로 분리한 PILS 연동시험과 개발된 전력제어분배장치 하드웨어와 데이터 송수신을 할 수 있는 HILS 연동시험을 통해서 동역학 시뮬레이터의 모든 성능을 검증하였다. 특히, 각 모델 및 제어 알고리즘은 독립된 기능별로 모듈화하였기 때문에 별도의 모델을 추가하거나 기존의 모델을 제거할 때 용이성을 최적화하였다.

본 논문은 2장에 동역학 시뮬레이터 모듈 구성을 3장에 자세제어 시뮬레이터 모듈 구성을 각각 설명하고, 4장에 엔지니어링 시뮬레이션과 1553B 통신을 이용한 프로세서 연동시험 및 하드웨어 연동시험을 수행한 결과를 정리하였고, 마지막 5장의 결론으로 구성되었다.

II. 동역학 시뮬레이션 모듈 구성

아래의 Fig. 1은 동역학 시뮬레이터를 구성하는 각 모듈과 그 기능 및 상호 입출력 흐름을 보여주고 있다. Fig. 1에 있는 각 모델들에 대한 자세한 모델링은 참고문헌[5]~[7]을 참조하고, 본 장에서는 추가로 보충할 내용만 정리한다.

먼저 위성과 휠의 동역학 관계를 표현하면 식 (1)과 같다. 여기서 $\vec{T}$는 위성에 가해지는 토크이고, $\vec{H}$는 위성 전체의 모멘텀이고, $\vec{\omega}_{M/F}$는 고정좌표계에 대한 회전좌표계의 각속도이고, $\vec{h}_w$는 휠만의 모멘텀이고, I는 위성 전체의 관성모멘트이고, J는 휠의 관성모멘트이고, $\vec{\omega}_w$는 휠속도이며, $|_M$는 회전좌표계를 의미한다.

$$\begin{aligned}\vec{T} &= \frac{d}{dt}\vec{H}|_M + \vec{\omega}_{M/F}\times\vec{H} \\ &= I\,\vec{\omega}_{M/F} + \vec{h}_w + \vec{\omega}_{M/F}\times(I\,\vec{\omega}_{M/F} + \vec{h}_w) \\ where,\ \vec{H} &= I\,\vec{\omega}_{M/F} + \vec{h}_w \\ \vec{\omega}_{M/F} &= (\omega_x, \omega_y, \omega_z)|_F\ ,\quad \vec{h}_w = J\,\vec{\omega}_w\end{aligned} \tag{1}$$

반작용휠의 경우 우주용으로 사용되는 종류는 크게 전압제어 방식과 전류제어(토크제어) 방식이 있는데 본 논문에서 다룬 모델은 전압제어 방

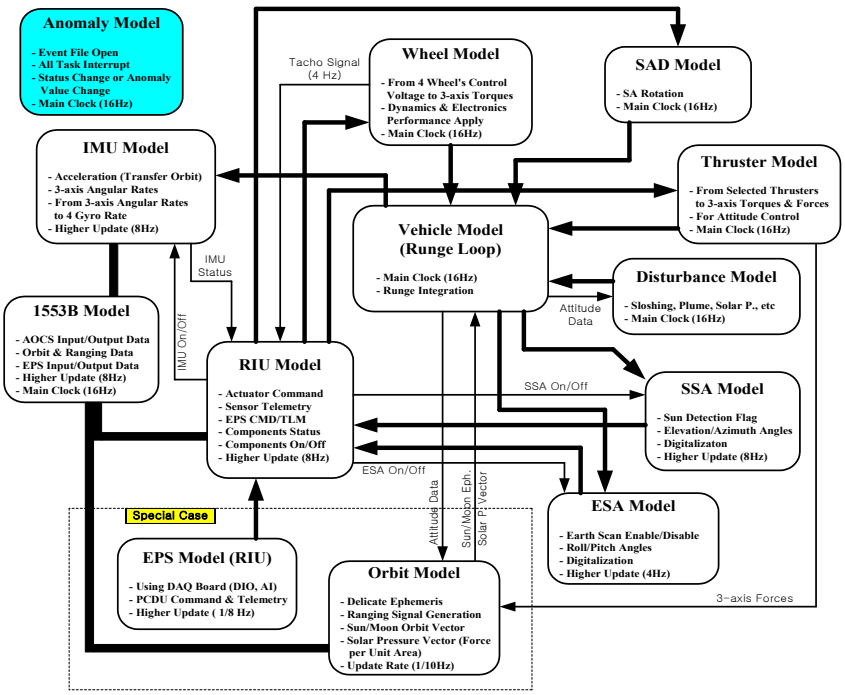

Fig. 1. 동역학 시뮬레이터 모듈 구성

식으로 일정한 휠속도를 유지하기 위해 해당 전압을 인가하는 방식이며 토크를 얻기 위해서는 시간에 비례한 전압 변화를 사용하게 된다. 이러한 특성으로 인해 참고문헌[6]에서 보여주듯이 내부에 복잡한 회로 구성을 갖게 되고 본 논문에서는 이러한 하드웨어 특성을 고려하여 모델링을 수행하였다.

또한, 휠속도를 측정하는 타코미터가 4 [Hz] 샘플링 주기와 18 [펄스/회전]의 하드웨어 특성을 갖고 있으므로 모델링은 식 (2)와 같다.

$$\begin{aligned} &Tacho\ pulses \\ &= Current\ speed\left[\frac{rad}{\sec}\right] = \frac{1}{2\pi}\frac{rev}{\sec}\Big] \\ &\quad \times 18\left[\frac{pulses}{rev}\right] \times \frac{1}{4}\ [\sec] \\ &= \left\{Current\ speed\left[\frac{rad}{\sec}\right] \times \frac{1}{2\pi}\right\}\left[\frac{rev}{\sec}\right] \\ &\quad \times \frac{9}{2}\left[\frac{pulses \cdot \sec}{rev}\right] \\ &\therefore\ Tacho\ pulses[pulses] \\ &= \left\{Current\ speed\left[\frac{rad}{\sec}\right]\right\} \times \frac{9}{4\pi} \end{aligned} \tag{2}$$

추력기의 경우 명령으로 주어지는 매우 작은 On-Time에 대해 실제 위성은 별도의 RIU 프로세서 클럭이 담당하지만, 본 논문에서는 다른 모델들의 동작시간과 전체적인 실시간 시뮬레이션을 보장하기 위해 추력기 전담 클럭을 설정하기가 어려워 Fig. 2와 같이 수정하여 적용하였다.

태양센서의 경우는 궤도시뮬레이션을 통해 관성좌표계의 태양위치를 궤도좌표계 성분으로 얻을 수 있다. 그리고 궤도좌표계 성분을 자세오차를 포함한 몸체좌표계 성분으로 표현할 때 3-2-1 (Z-Y-X) 오일러 변환으로부터 식 (3)을 얻을 수 있다. 이 변환은 피치(Z), 롤(Y), 요(X)각 순서로 회전하는 것을 의미하며 각각 Θ, ϕ, Ψ이다.

그리고 참고문헌[6]에서 알 수 있듯이 위성에 장착된 태양센서로부터 식 (4)와 같이 방위각(Θ_{AZ})과 고도각(Θ_{EL})을 수치적으로 얻는다.

$$\begin{aligned} X_B &= \cos\Theta\cos\phi X_O + \sin\Theta\cos\phi Y_O \\ &\quad - \sin\phi Z_O \\ Y_B &= (\cos\Theta\sin\phi\sin\Psi - \sin\Theta\cos\Psi)X_O \\ &\quad + (\sin\Theta\sin\phi\sin\Psi + \cos\Theta\cos\Psi)Y_O \\ &\quad + \cos\phi\sin\Psi Z_O \\ Z_B &= (\cos\Theta\sin\phi\cos\Psi + \sin\Theta\sin\Psi)X_O \\ &\quad + (\sin\Theta\sin\phi\cos\Psi - \cos\Theta\sin\Psi)Y_O \\ &\quad + \cos\phi\cos\Psi Z_O \end{aligned} \tag{3}$$

$$\begin{aligned} &\Theta_{AZ} = \cos^{-1}(Y_B)\ ,\ \Theta_{EL} = \sin^{-1}(Z_B) \\ &where,\ -\pi < \Theta_{AZ} \le \pi \\ &\qquad -\pi/2 < \Theta_{AZ} \le \pi/2 \end{aligned} \tag{4}$$

PCDU와의 데이터 송수신을 위해 Acromag사의 DIO(Digital Input/Output) IP(Industrial Pack) 보드와 AI(Analog Input) IP 보드를 동역학 시뮬레이터에 탑재하였다. Binary 변수는 DIO로, 아날로그 변수는 AI로 송수신 하면서 Fig. 3과 같이 DIO의 5채널을 할당하여 많은 양의 정보(Binary, 아날로그 모두 포함)를 송수신하는 RS-422 시리얼 통신을 하였다.

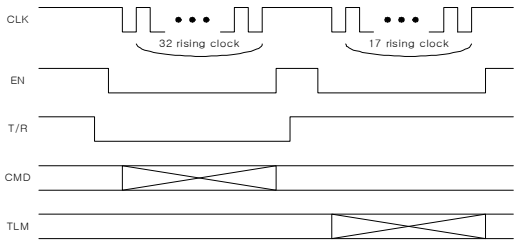

Fig. 3. DIO 보드를 이용한 RS-422 시리얼 통신

III. 자세제어 시뮬레이터 모듈 구성

Fig. 4는 자세제어 시뮬레이터를 구성하는 각 모듈과 그 기능을 보여주고 있다. 확장칼만필터 모듈에 대한 것은 참고문헌[5], 자세제어 모듈에 관련된 것은 참고문헌[7]에 소개되었고, 전반적인

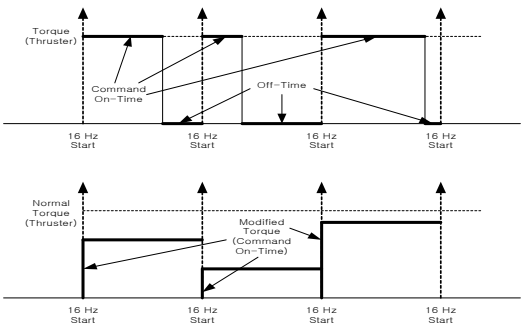

Fig. 2. 수정된 추력기 모델

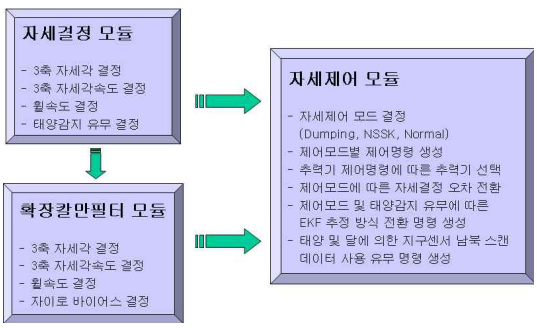

Fig. 4. 자세제어 시뮬레이터 모듈 구성

설명은 참고문헌[6]에 소개되었다.

이 장에서는 자세결정 모듈 중에서 태양센서에 의한 요각 결정 과정을 설명한다. 휠속도는 식 (2)를 역으로 처리하면 되고, 롤(Φ)과 피치각(Θ)은 지구센서로부터 간단히 얻을 수 있다.

식 (3)으로부터 식 (5)를 쉽게 얻을 수 있고 식 (6)과 같이 새로운 변수를 설정하면 식 (7)로부터 요각을 얻을 수 있다. 이때, Y_B와 Z_B는 식 (4)에서 얻을 수 있다. 또한, 태양위치의 실제 궤도좌표계 성분은 자세제어 시뮬레이터에서도 생성하거나 동역학 시뮬레이터와 공유해야 한다.

$$\begin{aligned} Y_B &= (\cos\Theta\sin\Phi X_O + \sin\Theta\sin\Phi Y_O + \cos\Phi Z_O)\sin\Psi - (\sin\Theta X_O - \cos\Theta Y_O)\cos\Psi \\ Z_B &= (\cos\Theta\sin\Phi X_O + \sin\Theta\sin\Phi Y_O + \cos\Phi Z_O)\cos\Psi + (\sin\Theta X_O - \cos\Theta Y_O)\sin\Psi \end{aligned} \quad (5)$$

$$\begin{aligned} p &= (\cos\Theta\sin\Phi X_O + \sin\Theta\sin\Phi Y_O + \cos\Phi Z_O) \\ q &= (\sin\Theta X_O - \cos\Theta Y_O) \end{aligned} \quad (6)$$

$$\tan\Psi = \frac{pY_B + qZ_B}{pZ_B - qY_B} \quad (7)$$

4개의 자이로 데이터로부터 3축 각속도를 얻기 위한 자세결정 식은 식 (8)과 같다. 이때, 변수들은 참고문헌 [5]와 동일하다.

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 & A & -A & 0 \\ -B & \frac{B}{2} & \frac{B}{2} & 0 \\ -C & -D & -D & -C \end{bmatrix} \begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \\ \omega_d \end{bmatrix}$$

$$where,\ A = \frac{1}{2\sin\beta\sin\alpha},\quad B = \frac{1}{\sin\alpha(\cos\beta+1)},\quad C = \frac{1}{\cos\alpha},\quad D = \frac{1}{2\cos\alpha\cos\beta} \quad (8)$$

IV. Simulation

시뮬레이션을 위한 정지궤도 위성체의 관성모멘트와 제어기 변수를 Table 1에 정리하였다.

Table 1. 관성모멘트 및 제어기 이득 변수

관성모멘트 (in-lbf-sec^2)			
변수	값	변수	값
I_{xx}	28618	I_{xy}	0
I_{yy}	20424	I_{xz}	0
I_{zz}	5214	I_{yz}	0

PID 이득 변수			
변수 / 축	K_F	K_P	K_I
x	0.5	127.75	0.975
y	0.5	83.81	0.639
z	2.0	15.27	0.1732

4.1 엔지니어링 시뮬레이션

이 절에서는 프로세서 연동시험의 성능을 비교하기 위해 태양 감지 전후의 엔지니어링 시뮬레이션(ES) 결과만을 나타내었고 자세한 것은 참고문헌 [6]에 정리되어 있다.

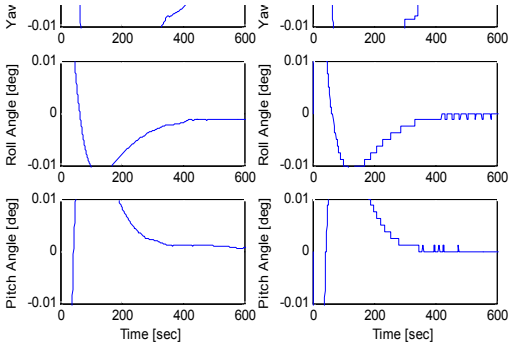

Fig. 5. 자세결정 결과 (태양 감지 상태, ES)

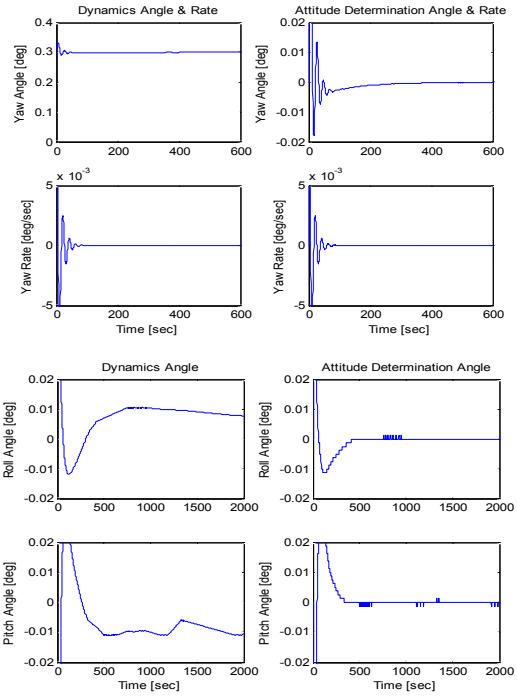

Fig. 6. 자세결정 결과 (태양 미감지 상태, ES)

시뮬레이션으로 궤도 Ephemeris를 변경하여 하나는 초기에 태양감지 상태를 다른 하나는 태양미감지 상태를 고려하고, 초기 (요, 롤, 피치) 자세 오차를 (0.3, 0.1, -0.2) [deg], 자세각 오차를 (0.01, -0.01, 0.01) [deg/sec]으로 동일하게 설정하였다. 결과는 Fig. 5와 Fig. 6과 같다. 수렴 이후의 실제 자세오차와 자세결정 차이는 센서의 디지털 변환 오차이다.

Fig. 6에서 롤과 피치의 경우는 요각이 정확히 제어되지 못하면서 발생되는 오차이며, 이러한 이유로 실제 위성에서 태양 미감지인 경우 오차 허용범위가 정상상태보다 높게 설정된다. 무궁화 위성 3호의 경우 ±0.02 [deg]를 허용한다.

4.2 프로세서 연동시험

엔지니어링 시뮬레이션 초기 오차와 다르게 본 시뮬레이션은 초기 (요, 롤, 피치) 자세오차를 (0.5, 0.02, -0.5) [deg], 자세각 오차는 없는 것으로 설정하였다. 이때, 궤도운동에 의해 태양 미감지 상태에서 감지상태로 변화하는 시점(14440 [초])을 중심으로 결과를 출력하였다.

결과는 Fig. 7 ~ Fig. 9와 같고, Fig. 7에서는 동일 출력에 대해 아래쪽이 자세결정 결과이며,

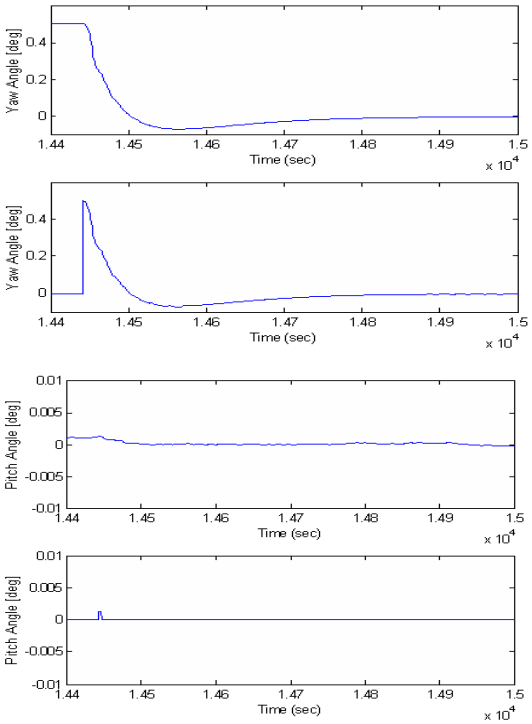

Fig. 7. 자세결정 결과 (태양 미감지에서 감지상태로 전환, PILS)

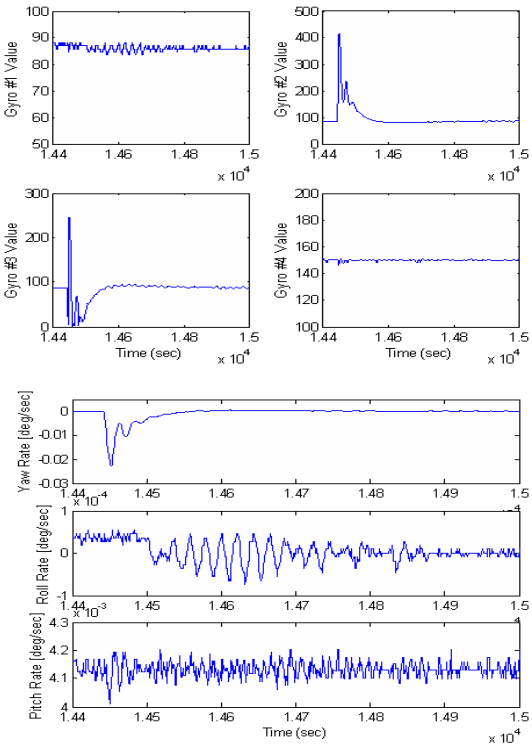

Fig. 8. 각속도 결정 및 변환 결과 (PILS)

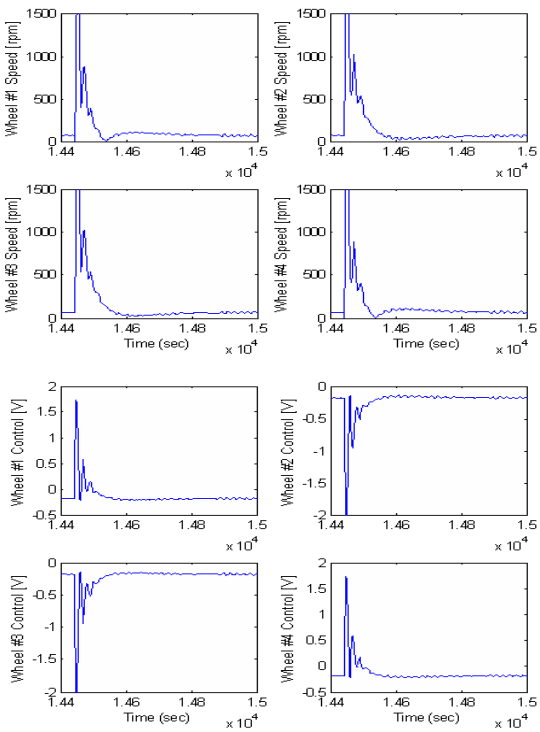

Fig. 9. 휠속도 결정 및 휠제어 명령 (PILS)

Fig. 8에서는 4개의 자이로 각속도를 결정한 후 3축 각속도로 변환한 결과를 보여준다.

이것은 엔지니어링 시뮬레이션 결과와 동일한 성향을 나타내므로 프로세서 연동시험이 원활하게 수행되고 있음을 알 수 있다. Fig. 8에서 피치 각속도는 궤도이동에 따라 위성이 지구를 정확하게 지향하기 위한 궤도 회전율(4.167e-3 [deg/sec]=제어기의 피치 각속도 기준 입력값)을 유지하고 있음을 볼 수 있다. 즉, PILS를 통한 데이터 전송 및 자세제어가 잘 수행됨을 알 수 있다.

Fig. 9는 태양 미감지 상태에서 감지 상태로 전환될 때 태양센서로부터 실제의 요각이 정확히 측정되면서 새로운 제어명령이 생성되는 과정을 휠속도 결정과 연동해서 얻은 결과를 보여준다.

이 절에서 얻은 결과들은 엔지니어링 시뮬레이션과 달리 실시간으로 수행하였으므로 시뮬레이션 마지막 시간인 15000 [초]의 경우 4시간 10분을 시뮬레이션한 것이다. 또한, 1553B 통신을 통해 16 [Hz]로 데이터가 교환되지만 센서 데이터는 4 [Hz] 및 8 [Hz]로 갱신되고 제어명령은 2 [Hz]로 갱신되기 때문에 어느 변수들은 항상 유지해 주어야 하는 것이 있는 반면 추력기의 동작시간은 일회성이 되어야 하는 복잡한 처리과정이 필요하게 된다.

4.3 하드웨어 연동시험

본 절에서는 동역학 시뮬레이터가 자세제어계 모듈에 영향을 주지 않고(Multi Tasking), 약 100여개에 해당하는 전력제어분배장치의 획득 데이터 및 동작을 위한 명령 데이터를 송수신하는 일련의 과정을 데이터획득보드를 통해 1/8 [Hz]로 원활히 수행할 수 있음을 검증하였다.

Fig. 10은 전력제어분배장치에 있는 릴레이를 구동하기 위해 약 30 [msec] 동안의 negative 펄스를 생성해야 하고 연속 구동을 위해 최소 100 [msec] 간격을 유지해야 하는 사양을 만족한 결과를 오실로스코프를 통해 얻은 것이다.

Fig. 11은 DIO 보드를 통해 RS-422 시리얼 통신을 수행한 결과로 실제 전력제어분배장치에 명령을 전송한 후 그 결과 값을 출력한 것이 아니고 데이터획득보드의 명령 채널과 획득데이터 채널을 동시에 연결하여 32 [bits]의 명령 스트림을 주어지는 상승 클럭에서 획득데이터 채널이 원활하게 읽는지를 오실로스코프로 얻은 것이다.

Fig. 12는 여러 개의 시리얼 통신을 1/8 [Hz] 동안 항상 수행해야 하므로 연속한 시리얼 통신이 원활하게 수행되는지 확인한 결과로 시험에 사용된 오실로스코프가 여러 개의 연속 데이터를 획득할 수 없어 최대로 클럭을 보여줄 수 있도록 하기 위해 연속으로 2개만 시리얼 통신한 결과를 보여준다.

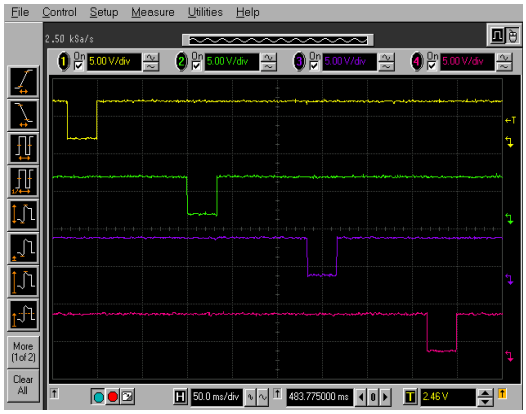

Fig. 10. 릴레이 구동 신호 생성 결과 (HILS)

Fig. 11. 시리얼 통신에 의한 명령전송 및 데이터 획득 시험 결과 (HILS)

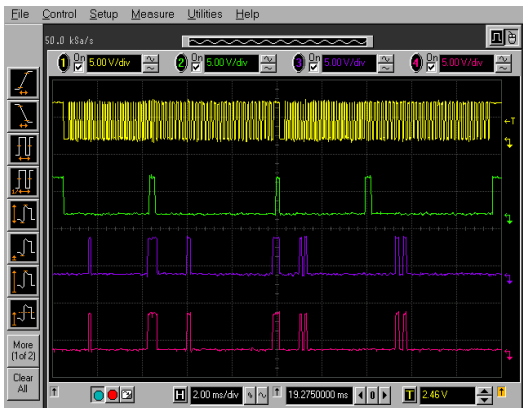

Fig. 12. 연속 시리얼 통신 결과 (HILS)

Table 2는 전력제어분배장치의 배터리 Enable과 Disable 명령(DIO 보드) 및 충전율 명령과 충전전류(AI 보드)를 보여주고 있다. 여기에서 'CH Mode'와 'Status'는 6비트 16진수 값을 나타내며, '3E'의 경우 첫 번째 비트가 'Off' 되어진 상태를 의미한다. 그러므로 좌측 열은 'bpc phase' A 6개와 B 6개 중 A의 첫 번째 것만 'Enable' 되어 있고, 우측 열은 A와 B의 첫 번째 것이 'Enable' 되어 있음을 의미한다.

Table 2. 배터리 명령에 따른 충전 상태 결과

```
------------------------------------------------------------
VDS start...
BPC A (CH mode) = 3F            BPC B (CH mode) = 3F

charge rate = 2.000000[A]
bpc_phs_A_EN_status = 1         bpc_phs_A_EN_status = 1
bpc_phs_B_DIS_status = 3F       bpc_phs_B_DIS_status = 3E
Battery [0/0] (CH) = 1.049648   Battery [0/0] (CH) = 2.031094
Battery [1/0] (CH) = 0.000000   Battery [1/0] (CH) = 0.000000
Battery [2/0] (CH) = 0.018633   Battery [2/0] (CH) = 0.000000

charge rate = 4.000000[A]
bpc_phs_A_EN_status = 1         bpc_phs_A_EN_status = 1
bpc_phs_B_DIS_status = 3F       bpc_phs_B_DIS_status = 3E
Battery [0/0] (CH) = 1.078945   Battery [0/0] (CH) = 1.694180
Battery [1/0] (CH) = 0.000000   Battery [1/0] (CH) = 0.000000
Battery [2/0] (CH) = 0.000000   Battery [2/0] (CH) = 0.018633

charge rate = 6.000000[A]
bpc_phs_A_EN_status = 1         bpc_phs_A_EN_status = 1
bpc_phs_B_DIS_status = 3F       bpc_phs_B_DIS_status = 3E
Battery [0/0] (CH) = 1.474453   Battery [0/0] (CH) = 3.027188
Battery [1/0] (CH) = 0.000000   Battery [1/0] (CH) = 0.000000
Battery [2/0] (CH) = 0.003984   Battery [2/0] (CH) = 0.033281

charge rate = 8.000000[A]
bpc_phs_A_EN_status = 1         bpc_phs_A_EN_status = 1
bpc_phs_B_DIS_status = 3F       bpc_phs_B_DIS_status = 3E
Battery [0/0] (CH) = 1.459805   Battery [0/0] (CH) = 3.041836
Battery [1/0] (CH) = 0.000000   Battery [1/0] (CH) = 0.000000
Battery [2/0] (CH) = 0.003984   Battery [2/0] (CH) = 0.000000

charge rate = 10.000000[A]
bpc_phs_A_EN_status = 1         bpc_phs_A_EN_status = 1
bpc_phs_B_DIS_status = 3F       bpc_phs_B_DIS_status = 3E
Battery [0/0] (CH) = 1.489102   Battery [0/0] (CH) = 3.041836
Battery [1/0] (CH) = 0.000000   Battery [1/0] (CH) = 0.000000
Battery [2/0] (CH) = 0.003984   Battery [2/0] (CH) = 0.003984
------------------------------------------------------------
```

V. 결 론

본 연구를 통해 정지궤도위성의 자세제어계 단품들의 하드웨어 특성을 모듈화하여 코딩한 동역학 시뮬레이터와 비행소프트웨어의 자세제어계를 모듈화하여 코딩한 자세제어 시뮬레이터가 전력제어분배장치 하드웨어와의 데이터 송수신 부하를 견디면서도 주어진 태스크 수행 주기에 문제없이 정상적으로 연동되고 있음을 검증하였다.

PILS 및 HILS 연동시험은 실시간으로 데이터를 처리해야 하는 하드웨어 특성을 자세제어 알고리즘이 충분히 수용할 수 있는지 검증할 수 있는 저비용 고효율의 시험 환경임을 본 연구를 통해서 다시 한번 확인할 수 있었다.

본 연구를 통해 획득된 동역학 시뮬레이터 및 자세제어 시뮬레이터 그리고 PILS 연동 환경 등을 활용하여 실제 위성에 탑재할 수 있는 자세제어 알고리즘을 적용하고 검증할 수 있을 것이다.

향후 동역학 시뮬레이터에 외란 모델이 추가되면 위성개발을 위한 자세제어계 분야에 대한 지상에서의 해석 및 검증 환경은 선진 개발업체와 대등한 기술을 보유하게 될 것으로 예상된다. 물론, 보다 정확한 데이터 및 반복된 검증 과정을 통해 보다 완벽한 시뮬레이션 환경을 구축해야 하는 노력이 더욱 필요할 것이다.

참고문헌

1) 한정엽, 박영웅, 방효충, 황보한, "1축 회전 시뮬레이터를 이용한 인공위성 지상 자세제어 실험 연구", 한국항공우주학회지, 제26권 제4호, 1998.

2) 박영웅, 한정엽, 방효충, 황보한, "다중 프로세서를 이용한 인공위성 실시간 자세제어 시뮬레이터 개발 : PILS(Processor In-the- Loop Simulation) 시스템 구조", 한국항공우주학회지, 제26권 제5호, 1998.

3) 한정엽, 박영웅, 방효충, "인공위성 3축 자세제어 지상 검증 Testbed 연구 - II. 실험검증연구", 한국항공우주학회지, 제27권 제6호, 1999.

4) Choongsuk Oh, Hyochoong Bang, "Ground Experiment of Spacecraft Attitude Control Using Hardware Testbed", Korean Society for Aeronautical and Space Sciences, Vol, 4, No. 1, 2003.

5) 박영웅, 박봉규, "확장칼만필터를 이용한 정지궤도위성 자세결정 연구", 한국항공우주학회 추계학술발표회 논문집, 2001.

6) 박영웅, 방효충, "정지궤도위성 동역학 시뮬레이터 개발 연구", 한국항공우주학회 추계학술발표회 논문집, 2003.

7) 박영웅, 남문경, 방효충, "정지궤도위성의 휠모멘텀 관리 로직 연구", 한국항공우주학회지, 제31호 제3권, 2003.