telescope can be achieved in space, infrared (IR) obervations have been tried through small missions in Korea. Based upon the previous technical development for infrared spectro-photometric instrument, NISS (Near-infrared Imaging Spectrometer for Star formation history) onboard NEXTSat-1, we participated in the all-sky infrared spectro-photometric survey mission, SPHEREx.

The SPEHREx (Spectro-Photometer for the History of the Universe, Epoch of Reionization, and Ices Explorer) was selected as the NASA MIDEX (Medium-class Explorer) mission (PI Institute: Caltech) in this February. As an international partner, KASI will take part in the hardware development, the operation and the science for the SPHEREx. The SPHEREx will perform the first all-sky infrared spectro-photometric survey to probe the origin of our Universe, to explore the origin and evolution of galaxies, and to explore whether planets around other stars could harbor life. For the purpose of the all-sky survey, the SPHEREx is designed to have a wide FoV of 3.5 × 11.3 deg. as well as wide spectral range from 0.75 to 5.0μm. Here, we report the status of the SPHEREx project and the progress in the Korean participation.

[구 AT-02] Demonstration of the KPLO operation visualization using Cosmographia

Joo Hyeon Kim, Jo Ryeong Yim, Young-Rok Kim, Dong-Gyu Kim
Korea Aerospace Research Institute

우리나라 최초의 우주탐사 사업인 달 탐사선(KPLO) 개발 사업에는 국내에서 개발하는 과학 탑재체 3기, 기술 검증탑재체 1기, 고해상도 카메라 1기, 국제협력의 일환으로 NASA의 과학 탑재체 1기도 함께 개발되고 있다. KPLO와 이들 탑재체의 운영을 수행하게 될 KPLO 심우주 지상시스템은 달 탐사선 운영에 필요한 궤도, 임무계획 등의 정보를 생성하고, KPLO의 기동명령과 상태정보를 송, 수신하는 역할을 주요 임무로 수행한다. 또한 이들 정보를 기반으로 궤도임무를 수행하고 있는 KPLO의 임무운영 상태를 시각화하여 운영자로 하여금 KPLO 운영을 용이하게 하고, 공공에게 이를 제공하는 역할도 함께 수행한다.

KPLO 심우주 지상시스템은 AGI사의 STK와 NASA/JPL에서 개발한 Cosmographia를 활용하여 각각 특성에 맞는 KPLO 운영 시각화 정보를 제공할 것이다.

본 발표에서는 Cosmographia의 작동 및 활용 개념을 설명하고, KPLO의 가상 임무를 적용한 SPICE Kernels을 활용하여 고해상도 카메라인 LUTI의 지향, 달 중심 표준 좌표를 적용한 KPLO의 궤도 등을 시각화 시연을 한다. 또한 고해상도 달 표면 영상 적용, 심우주 네트워크 안테나의 위치 정보표출 등 Cosmographia에서 기본적으로 제공하던 시각화 정보를 개선한 내용에 대해서도 함께 시연한다.

[구 AT-03] Overview of Modeling Process using Giant Magellan Telescope Software Development Kit

Jimin Han[1],ChanggonKim[1], Martí Pi[2], Josema Filgueira[2], Marianne Cox[2], Chien Peng[2], Alfonso Román[2], Jordi Molgó[2], William Schoenell[2], Hector Swett[2], Divya Thanasekaran[2], Pierre Kurkdjian[2], Charles plubell[2], Tae-Geun Ji[1],Hye-InLee[1], Soojong Pak[1]
[1]Kyung Hee University. [2]Giant Magellan Telescope Organization.

The software group in Giant Magellan Telescope Organization (GMTO) is developing the GMT Software Development Kit (SDK) for the device control and the telescope operations. The SDK is dived into the modeling and the operation defining. In the modeling process, Domain Specific Language (DSL) can validate the availability of a model and generate a skeleton code automatically. After the modeling, the developer can simply define the device operation. All devices are connected via EtherCAT, and the SDK simplifies the network connection. This presentation will give an overview of the modeling process and development examples using the GMT SDK.

[구 AT-04] Development Process on the Control Software for Camera and Grating Articulation System Prototype (CGAS-P) of the Giant Magellan Telescope Multi-Object Astronomical and cosmological Spectrograph (GMACS)

Tae-Geun Ji[1], Erika Cook[2], Evan Kelly[2], Darren L. DePoy[2], Jennifer Marshall[2], Hye-In Lee[1], Soojong Pak[1]
[1]School of Space Research, Kyung Hee University,
[2]Department of Physics & Astronomy, Texas A&M University

We present the control software and its development process for a prototype of the Camera and Grating Articulation System (CGAS) for GMACS, a wide-field, multi-object, moderate-resolution optical spectrograph for the Giant Magellan Telescope (GMT). The CGAS prototype is currently designed for the camera articulation controller as a miniature model of the GMACS. The camera articulation package (CAP) is a software that controls two stepper motors to adjust the camera angle. The package is developed using Visual C++ and runs on Windows 10. We discuss the architectural design and communication route between the high-end user software and the electronics hardware.