

COSMO: Using Python and Databases to Monitor HST's Cosmic Origins Spectrograph

Joanna M. Taylor, Justin Ely, and Mees Fix

Space Telescope Science Institute, Baltimore, MD, USA

Abstract. The Cosmic Origins Spectrograph (COS) was installed on the Hubble Space Telescope (HST) in May 2009. COS is designed to perform high-sensitivity medium- and low-resolution spectroscopy of astronomical objects in the far-ultraviolet and near-ultraviolet wavelength regimes. We present an overview of a new comprehensive data repository, COSMO (COS Monitoring) that is capable of automatic data retrieval and reduction as well as monitoring the instrument through analysis of observations and telemetry. COSMO relies on a database that allows for dynamic querying across a wide array of gathered data metrics. Calibration programs that monitor the instrument's stability and behavior have been integrated with COSMO to provide timely and accurate analysis of the COS detectors.

1. Data Retrieval and Calibration

COSMO¹ can be roughly divided into three components: data retrieval and calibration, ingestion of data into the database, and instrument monitoring. COSMO utilizes the Sphinx documentation generator for code documentation.² COSMO also uses Travis CI, a continuous integration service that automatically builds and tests code under different environments.³

1.1. Data Retrieval and Calibration

All HST data are hosted on the Mikulski Archive for Space Telescope (MAST). Using SQLAlchemy, COSMO automatically determines if there are new COS data to retrieve by comparing the COSMO database (MySQL) to the MAST database (TSQL) (`cos_monitoring.retrieval.find_new_cos_data`). New data can also be specified as any datasets missing from the local directory when compared to the MAST database. Once missing data are identified, they are retrieved by submitting a formatted XML file to a MAST server `cos_monitoring.retrieval.request_data`. Data are transferred to a specified local directory via SFTP. Once all data have been successfully transferred locally, they are calibrated in parallel using the CalCOS pipeline `cos_monitoring.retrieval.manualabor`. This process results in a local, up-to-date repository of data that can be easily accessed by users within STScI. Individ-

¹See https://github.com/justincely/cos_monitoring.

²See <http://cosmo.readthedocs.io>.

³See https://travis-ci.org/justincely/cos_monitoring.

ual datasets, or the entire local repository can be re-retrieved and reprocessed at will (`cos_monitoring.retrieval.reprocess`).

2. Database Population

COSMO uses a MySQL database server with SQLAlchemy, an object-relational mapper. The database is broken up into different relational tables, which include header keyword and monitoring information. Individual classes define the structure of each table, and a general connection scheme allows for easy access to each table (`cos_monitoring.database.db_tables`). The bulk of the heavy lifting is performed in `cos_monitoring.database.database`, which finds new data in the local directory and ingests it into all applicable database tables. The database used by COSMO can also be completely wiped, allowing for a clean ingest of all data.

3. COS Monitors

In addition to providing dynamic records of all COS data, COSMO also automatically uses the database information to perform routine checks of the COS detectors. Using packages such as `astropy` (Astropy Collaboration et al. 2013), `numpy` (Stefan van der Walt & Varoquaux 2011), `scipy` (Jones et al. 01), and `matplotlib` (Hunter 2007), we can track the evolution of the COS detectors. This includes monitoring the gain degradation of the instrument over time, the drift of grating wheels (optics selection mechanism), the evolution of the dark rate, and the X and Y positions of the stim pulses which correct for any shifts and stretches of the detector (see `cos_monitoring.database.database`). These monitoring scripts result in reference files used during calibration. Timely instrument monitoring is crucial for optimizing the performance of the instrument and the science.

4. COSMO By The Numbers

There are currently **12** tables in the COSMO database. Wiping the database and ingesting the total number of COS datasets to date (**341,000+**) takes less than **24** hours. COSMO formally entered this world **July 10, 2015** and has **318** commits to date with over **10,000** lines of code and counting.

Acknowledgments. We would like to acknowledge the use of the python packages SQLAlchemy and CalCOS. This work would not have been possible without help from the STScI COS team as well as the STScI OED division.

Orbital Variation in Darkrate for FUVB

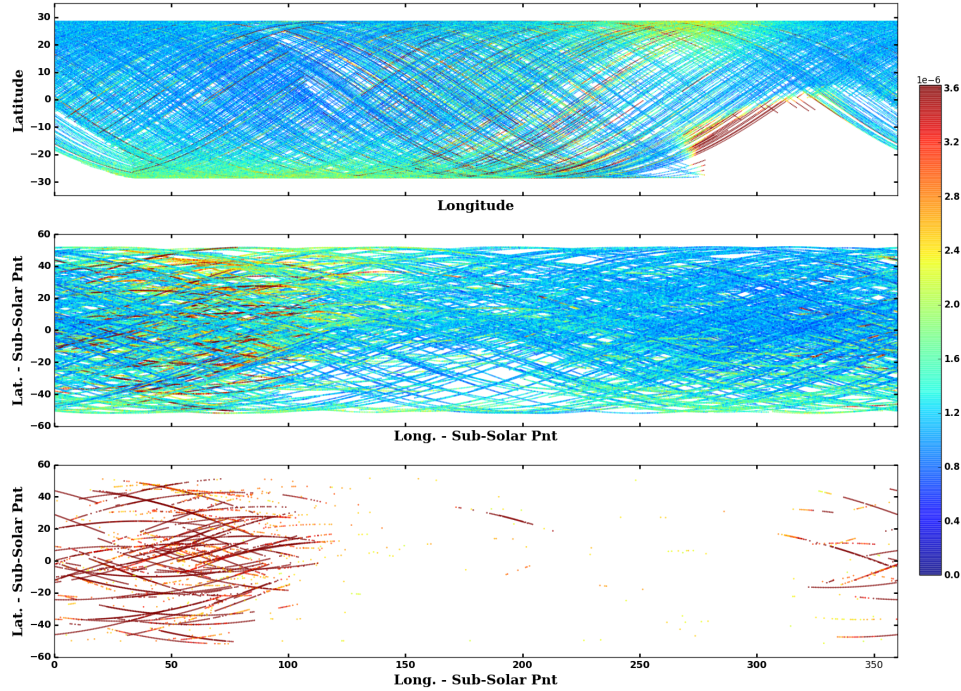


Figure 1. The COS/FUV channel dark rate as a function of HST orbital location. The area near (-20, 310) is the South Atlantic Anomaly.

References

- Astropy Collaboration, et al. 2013, *A&A*, 558, A33. 1307.6212
 Hunter, J. D. 2007, *Computing in Science and Engineering*, 9, 90. URL <http://scitation.aip.org/content/aip/journal/cise/9/3/10.1109/MCSE.2007.55>
 Jones, E., Oliphant, T., Peterson, P., et al. 2001–, *SciPy: Open source scientific tools for Python*. [Online; accessed 2016-10-12], URL <http://www.scipy.org/>
 Stefan van der Walt, S. C. C., & Varoquaux, G. 2011, *Computing in Science and Engineering*, 13, 22. URL <http://scitation.aip.org/content/aip/journal/cise/13/2/10.1109/MCSE.2011.37>