Haystack AeroVista REU

Hirani Sattenapalli, Aparna Rajesh, T. Lucas Briggs
“Aurora Touching Sunrise” from NASA archives
AERO VISTA Mission Introduction

- AERO & VISTA satellites will collect radio frequency (RF) data from the auroral regions
- Data will be used to accomplish science and tech goals
  - Study emissions such as Auroral Kilometric Radiation (AKR), Medium Frequency Burst (MFB), Auroral Roar, and Auroral Hiss
  - Validate usage of Vector Sensor Interferometry and RFI survey
Monopole, Horizontal Loop, and Rectangular Dipoles correspond to channels on spectrogram.
AERO-VISTA Interactive Spectrogram Display

Hirani Sattenapalli
AVIS Display & Objectives

- Provide a tool for the science team to visualize metadata
- Present spectrogram data in plotly graphs
- Allow science team to perform computation on channel data and send commands for in-flight processing
Libraries Used

- **Plotly**
  - Graphing utility used for telemetry maps and spectrogram plots
- **Redis**
  - In-memory data structure used to store metadata
- **Digital RF**
  - Software used for reading and writing spectrogram metadata into digital RF format
Libraries Used

- Dash
  - Python framework to build web pages
  - Used to build and style layout and components of the dashboard

- Numpy
  - Python library to work with arrays and matrices

- Xarray
  - Python package that adds dimensions and coordinates to numpy arrays
  - Used to organize metadata to place into redis
Existing AVIS Display
Goals for Dashboard Version 2.0

- Faster Loading of Data
  - Updating the spectrogram by retrieving and processing the summary data files presents a high computational load

- Display of Telemetry Data
  - Spacecraft speed, location, and altitude
  - Used to provide context for science team

- Generation of Subplots to do computation between channels

- Overall Design Changes to increase visual & user interactibility
File Structure

AVIS Version 1

- Metareader used to read summary data file
- Metatime provides timestamp data
- Specmeta creates spectrogram plot
- Util - creates dash components for the display
File Structure

AVIS Version 2.0

- Specmeta and Metareader files replaced with Specdata file
- Specdata:
  - used to enter summary data into redis for in-memory storage
- App.py files
  - Split into index.py, app.py, & app pages to accommodate multi-page dash app
  - Easier for future additions
Summary of New Layout & Results

- **App.py structure & index.py**
  - Allows for easy addition of future improvements

- **Redis interface & backend structure**
  - Enters all spectrogram and time data into redis
  - Meant for future collaboration between Lucas and Aparna’s work
  - Access files in redis and send uplink files into redis

- **Speed**
  - In memory storage of data did not speed up spectrogram plot generation as desired
  - Data input to redis makes it easier for data to be accessed and exported in the future
Summary of New Layout & Results Cont.

- Telemetry data page
  - Provides a data table of satellite speed, position/ location, & altitude
  - Provides context for science team when analyzing spectrogram data
- Subplots page
  - Continuous regeneration of spectrogram plots to use for computation between different channels (ex. sum(loops), mult(dipoles), division, linear combinations)
  - Used to classify if data is showing electrostatic or electromagnetic phenomena & type of emission
Future Work

- Computation between channels
  - Currently there is subplot generation; computation for channel math needs to be developed
- Improvements on redis structure
  - Data organization in redis and improvements on file access
- Satellite video/ display in home page
  - Future satellite data to be presented on the home page


