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

AERO VISTA Interactive Spectrogram

Display Controls
- Data Scaling: Log, Linear
- Y Axis Scale: Linear, Log
- Color Map: Winds, Map
- Plot Format: Linear, Log
- X Limits: 0%, 100%
- Y Limits: 0%, 100%

Metadata Info
- File Path: [Path to file]
- Date: 11/17/2020
- Sample Length: 109,937 seconds
- Sample Rate: 10,024 Hz
- Channel: Dipole A

Selection Bounds
- X min, X max
- Y min, Y max
- No selection

Selection Specs
- Event Type: Unknown
- Processing Type: Full

Selection Table
- Optional File Name: [File name]

Selection Resolution Grid
- Frequency Resolution: [Resolution]
- Time Resolution: [Resolution]

Start date: 11/17/2020

Frequency [MHz]:
- 0.5
- 2.5
- 3.5
- 4.5

Time [UTC]:
- 21:39:30
- 21:40:00
- 21:40:30
- 21:41:00
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
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


Satellite Commanding API

Aparna Rajesh
Commanding?

- Once in orbit, satellites must maintain a connection with ground systems
- Example tasks:
  - Ping
  - File transfer
  - Data collection
- Commands are sent up via the uplink & a response is sent back via the downlink

person at desk image: https://media.istockphoto.com/vectors/thinking-and-typing-vector-id1154317230?k=6&m=1154317230&s=612x612&w=0&h=ui0CK04fYmDjyNZDE1YV-gjiVvADc7RJNC_jhg=
Goals

- **Develop an API** (Application Programming Interface, an intermediary software that takes data from a different software and then sends it to another API or program) **that will:**
  - neatly package command data and metadata
  - serialize all the information pertaining to a command or command schedule in a predetermined format
    - human readable format
    - hex format for the spacecraft
  - pass this byte stream to another program
AV-Command-Schedule-API

- The AV-Command-Schedule-API:
  - creates a command object with using instance-specific parameters and other data stored in config files
  - establishes a command “connection” using a connection modifier
    - *where will this command be sent?*
  - generates a human-readable component of the command object
  - generates a hex stream that is to be read by the spacecraft
test = Connection(ping_aurora(5, 4, 2, 1, 0), "aurora_payload_computer")
test = Connection(ping_aurora(5, 4, 2, 1, 0), "aurora_payload_computer")
Next Steps

- Command schedules
- Hex headers
- Command dictionary parameter data types
- Config files
Ground Station Radio Communications

T. Lucas Briggs
What is a Ground Station?

- Enables communication between spacecraft and mission operations
- Handles all radio operation, command scheduling, and data verification tasks

Source: https://www.gim-international.com/content/news/pasco-provides-rental-service-for-satellite-ground-station-facilities
For AERO/VISTA

- Multiple possible Ground Stations in different locations with a wide range of radio configurations

- Many different types of data with varying sizes of payload
  - **Commands**: small, carries data necessary for spacecraft to execute desired action.
  - **Acknowledgements**: very small, carries data necessary to say “command received”
  - **Telemetry**: large, carries as much information as possible about spacecraft state
  - **Science**: very large, carries a full time-series of spectrum data from auroral emissions
Digital Communications over Radio

**UPLINK**

Data I want to send: 11010011

"frame" for verification: frame |11010011| frame

"modulate" for transmission:

"demodulate": frame |11010011| frame

"deframe" and verify: 11010011

**DOWNLINK**

"deframe" and verify: 00010001

"demodulate": frame | 00010001 | frame

"modulate" for transmission:

"frame" for verification: frame | 00010001 | frame

Data I want to receive: 00010001
Summary of Requirements

Need a software package that...

- Enables framing/deframing and mod./demod. on uplink/downlink.
- Allows for a wide range of data payload lengths.
- “Drops in”. Can be imported, started, and maintained anywhere.
- Provides a way to access data remotely and asynchronously.
- Is as configurable as possible
- Is as extendable as possible
Implementation - GNURadio Flowgraph
Implementation - Software (simplified)

- **Database** provides asynchronous remote access.
- **Ground Station** provides configuration, handles top-level operation.
- **Radio Interface** manages framing/deframing data, uplinking/downlinking frames.
- **Flowgraph** operates the Radio Transceiver.
Implementation - Software (simplified)

- **Database** provides asynchronous remote access.

- **Ground Station** provides configuration, handles top-level operation.

- **Radio Interface** manages framing/deframing data, uplinking/downlinking frames.

- **Flowgraph** operates the Radio Transceiver.

---

MIT
HAYSTACK
OBSERVATORY
Live Demo - PNG File Transfer
Next Steps

- Interaction of the Ground Station with the Database needs to be expanded
  - Operation commands, logging, configuration, and satellite command scheduling, all remote-asynchronous
- GNU Radio Flowgraph needs to be expanded
  - More/better signal processing for higher SNR (configurable)
- Develop a “mock satellite” version of the Ground Station
  - Responds to known commands with dummy data
  - Potentially applies a channel model to the signal to test non-ideal conditions
- Integrate with a real transceiver and satellite engineering model
backup
Implementation - Software (expanded)

- Radio Interface exists as 3 important blocks
- Each has a single responsibility, promotes extendability
- See report for more details
Modulation and Framing for A/V

- Modulation using Gaussian Frequency Shift Keying (GFSK)
  - Looks a bit like this:

- Flexible data framing with the following template:
Reliable Data Transfer: Framing

- **Preamble**: recognize symbols, “lock on” to binary data
- **Syncword**: find the start of the data packet
- **Length**: how many bytes to expect in the data
- **Address**: where should this data go

- **CRC (Cyclical Redundancy Check)**: Verify that the data has not changed
- **Whitening**: Deterministically randomize the data
Configuration Through YAML Files

Configurable parameters are categorized into “packet”, “radio”, and “redis” groupings.
Digital Communications: Modulation

- Binary data representation on a “constant wave”, accomplished through keying
  - Amplitude-Shift Keying (ASK)
  - Frequency-Shift Keying (FSK)
  - Phase-Shift Keying (PSK)
  - ...many more
- Our solution: Gaussian FSK (GFSK)
  - FSK with no sudden jumps
Final Group Demo
Thank you to our mentors

Mary Knapp, Ryan Volz, John Swoboda, Frank Lind, Phil Erickson, Toby Gedenk, and Geoff Crew