AERO-VISTA Ground Station Radiocommunications

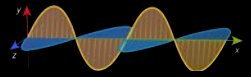
Michael Gutierrez
California Institute of Technology

Maximilian Riccioli
University of Texas at Austin

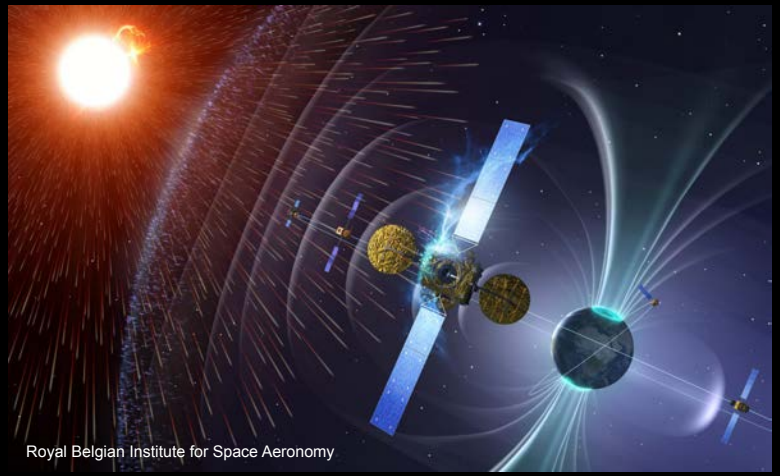
MIT HAYSTACK OBSERVATORY


AERO/VISTA Overview
Communications Background
The Master Plan: Haystack Satcom
Full Stack Walkthrough
What’s Next
Reflections

Crazy things happen when high-energy solar radiation finds its way to the poles!

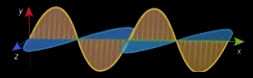
- Lots of electric charge + a strong magnetic field = 

Visible Light

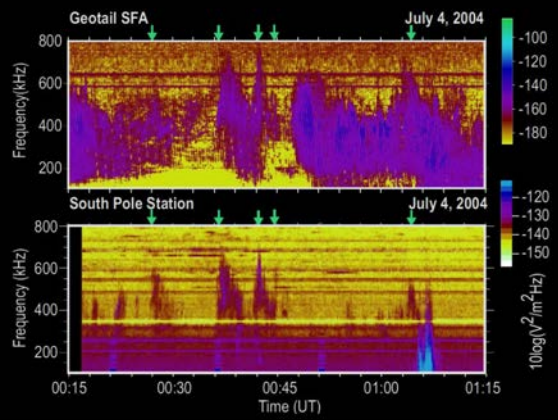

Royal Belgian Institute for Space Aeronomy


Getty Images

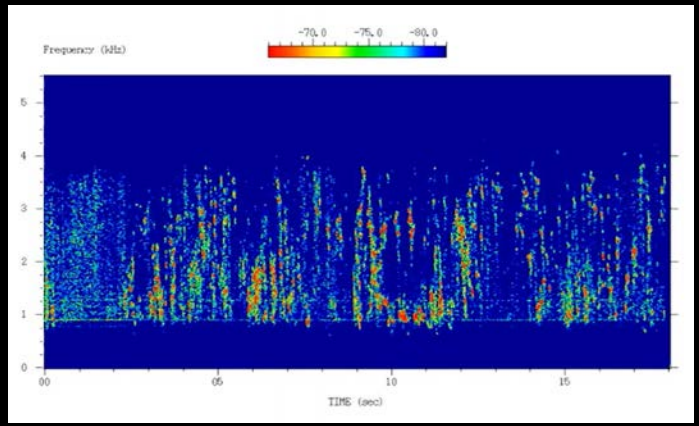
Crazy things happen when high-energy solar radiation finds its way to the poles!

- Lots of electric charge + a strong magnetic field = 

Radio Waves



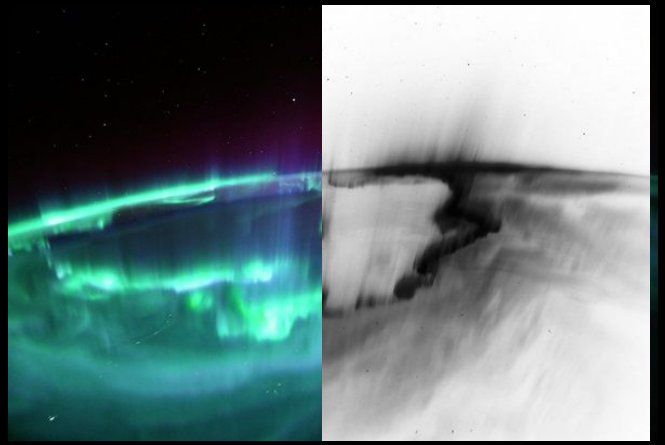
LaBelle, J., and Anderson, R.R., Ground-level detection of Auroral Kilometric Radiation



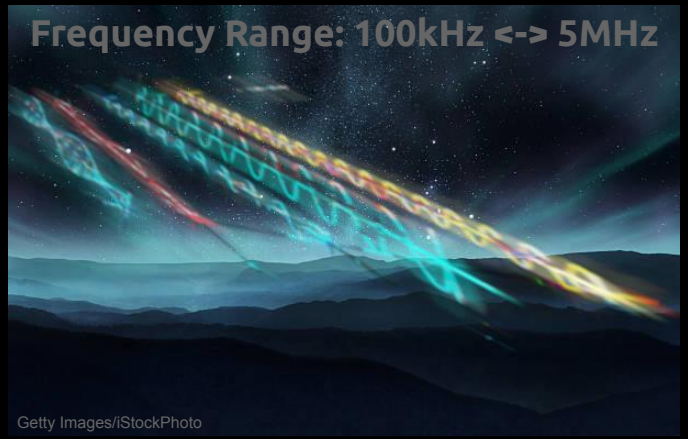
https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1977-102A

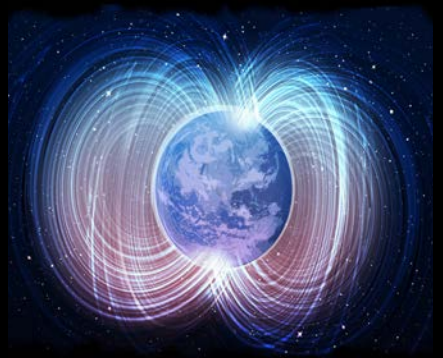
How to study these phenomena?

- What do these radio waves look like, and where do they come from?
- How do we observe them through the kilometer-wave-opaque atmosphere?



https://astronomy.com/news/2021/11/recent-aurora-captured-from-iss

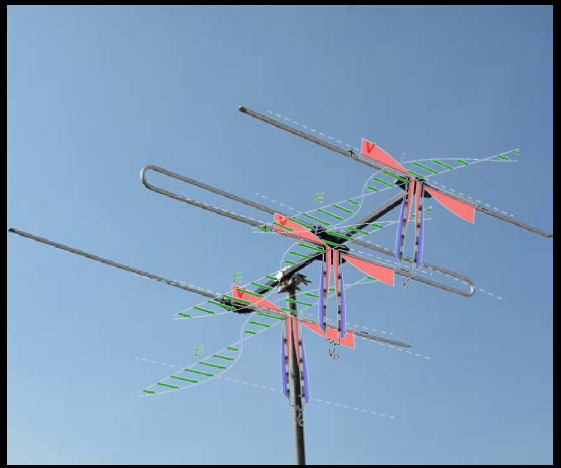




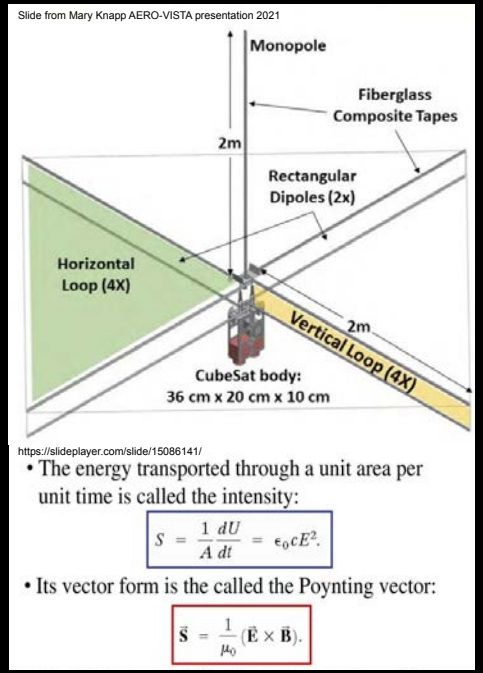
https://www.earth.com/news/earths-magnetic-poles-flip/

Radio waves, antennas, electromagnetism, oh my!

- Satellites above the ionosphere can get us physical access
- We still need to think in 3D to design the optimal science gear



Dipole/loop antenna = 1 dimension of data 😔



Vector sensor antenna = 6 dimensions of data 🤩

MIT HAYSTACK OBSERVATORY

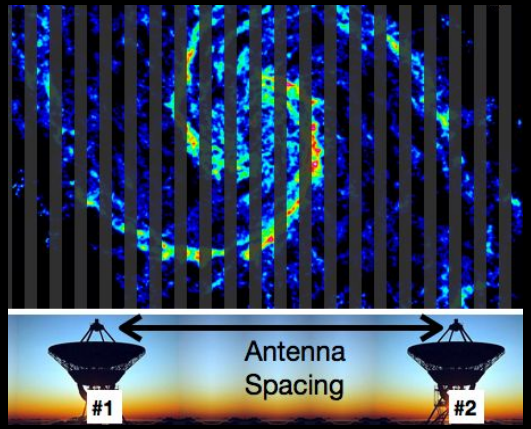
Auroral
Emissions
Radio
Observatory

Vector
Interferometry
Space
Technology using
AERO

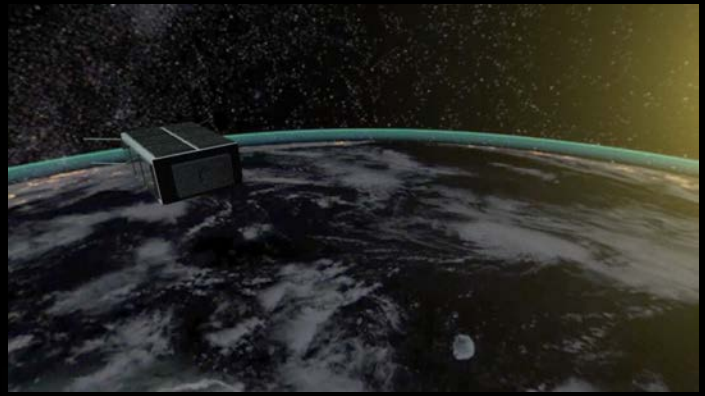
MIT HAYSTACK OBSERVATORY

The spacecraft

- Twin 6U CubeSats to more-than-double the data output with interferometry



https://astrobites.org/2013/04/17/the-whirlpool-galaxy-like-youve-never-seen-it-before/




AERO/VISTA Overview
Communications Background
The Master Plan: Haystack Satcom
Full Stack Walkthrough
What's Next
Reflections

OBJECTIVE #1: We need to talk to the spacecraft.

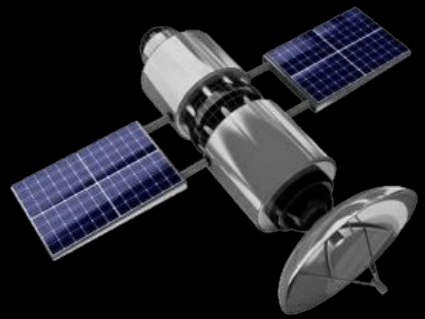
- Uplink (data to sat): Pings, commands, data acknowledgements
- Downlink (data from sat): Health/status telemetry, experiment data, command acknowledgements

SOLUTION: Use a ground station:

Westford Radio Telescope

Uplink

Downlink

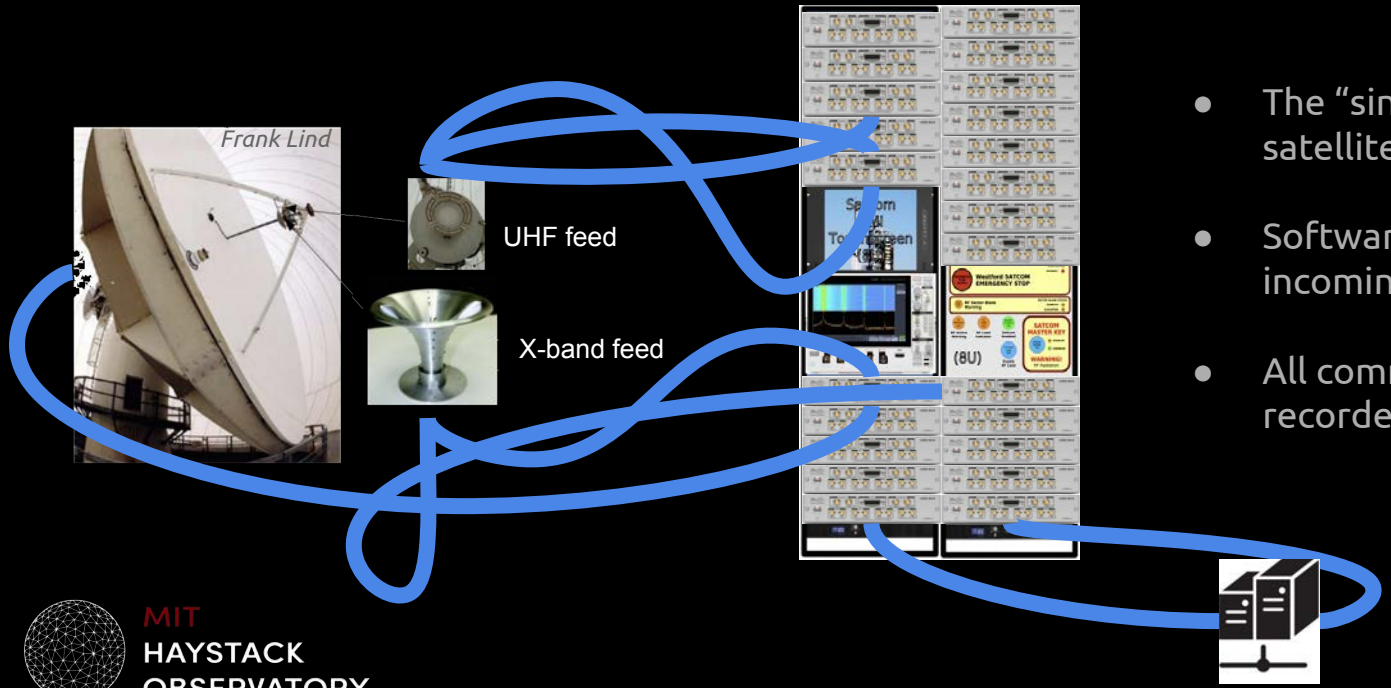


Satellite

https://png.pngitem.com/pimgs/s/116-1166977_satellite-png-transparent-png.png

MIT HAYSTACK OBSERVATORY

"Ground Zero" -- the Westford antenna



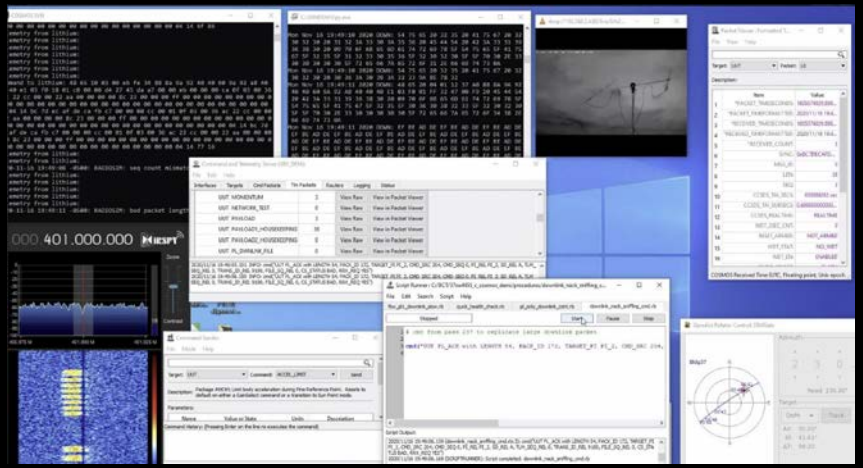
- The "single point of failure" for satellite communications
- Software-defined radios process incoming signals in real time
- All communications are also recorded for later reference

MIT HAYSTACK OBSERVATORY

OBJECTIVE #2: We need to NOT do everything by hand.

- LEO satellite passes only last from 5 to 15 minutes - very limited time to exchange data & send commands
- It's inefficient and unnecessary to have a human manually control low-level communication processes.

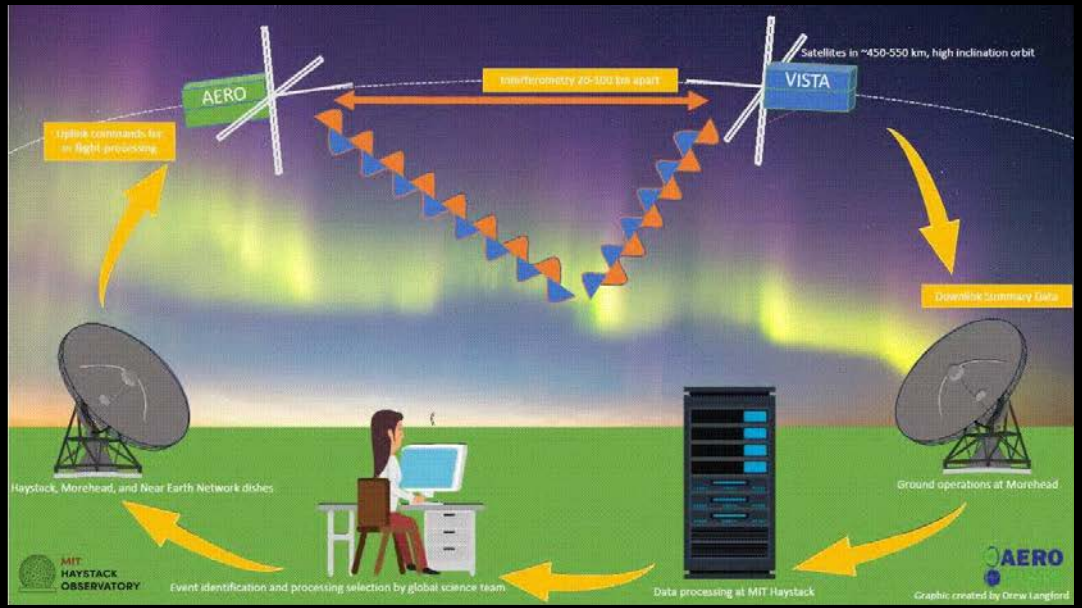
SOLUTION: Use a scheduler.




AERO/VISTA Overview
Communications Background
The Master Plan: Haystack Satcom
Full Stack Walkthrough
What’s Next
Reflections

Haystack Satcom

- Open source software package
- Full stack: from signal to spreadsheet
- Configurable for any satellite or mission
- Needs to remain human-accessible



Drew Langford

AERO/VISTA Overview

Communications Background

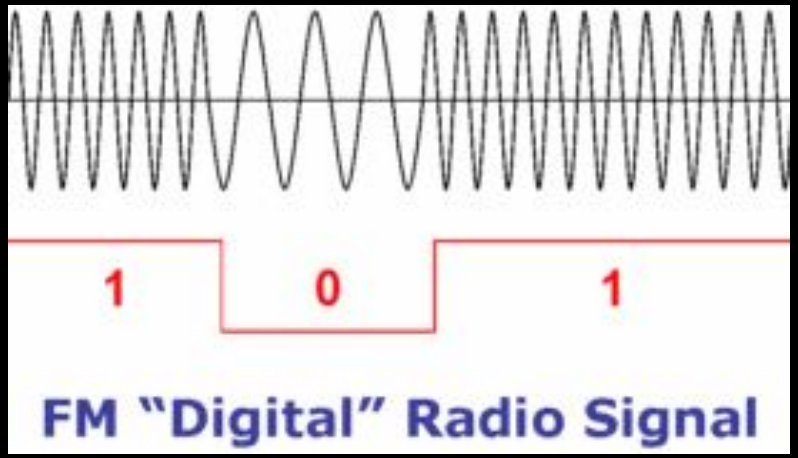
The Master Plan: Haystack Satcom

Full Stack Walkthrough

What’s Next

Reflections

GNURadio: Open Source Signal Processing

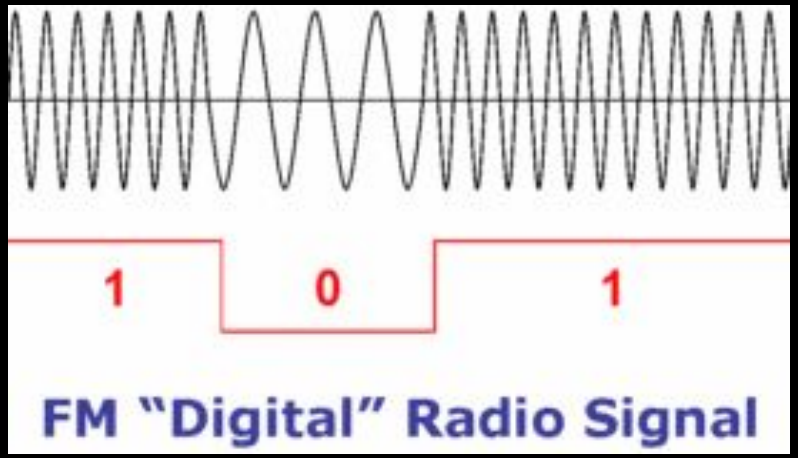


https://atlantahiddendogfence.com/the-truth-about-digital-modulation/

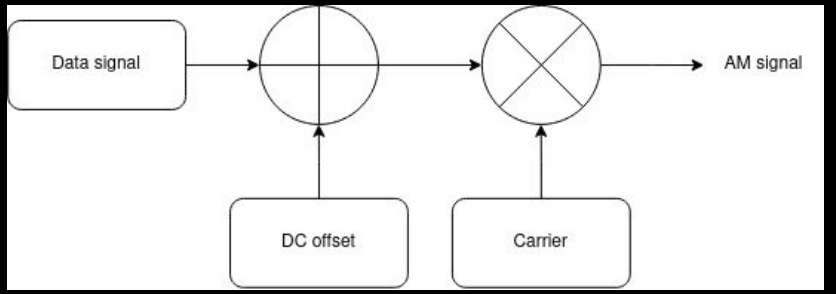
MIT HAYSTACK OBSERVATORY

GNURadio THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

GNURadio: Open Source Signal Processing



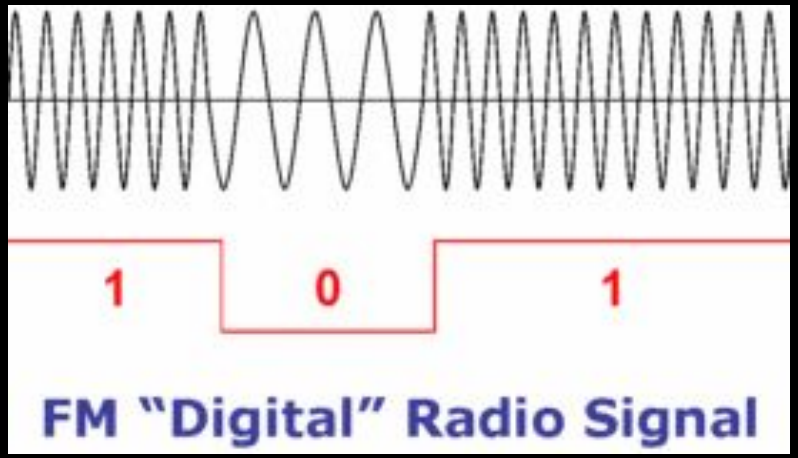
https://atlantahiddendogfence.com/the-truth-about-digital-modulation/



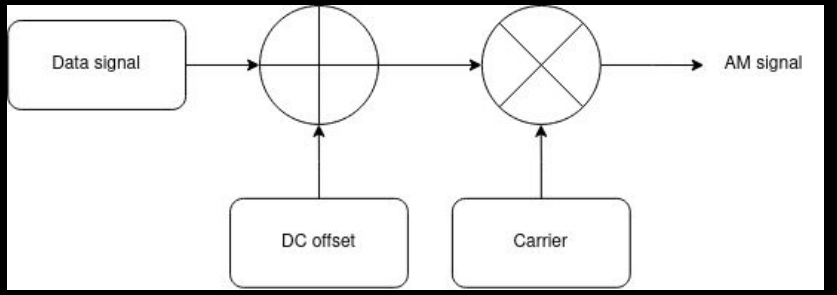
MIT HAYSTACK OBSERVATORY

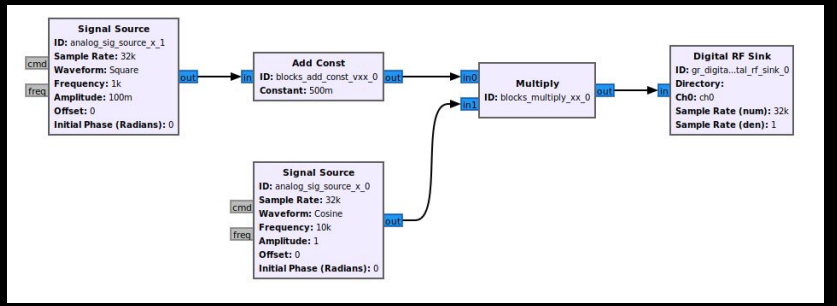
GNURadio THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

GNURadio: Open Source Signal Processing



https://atlantahiddendogfence.com/the-truth-about-digital-modulation/

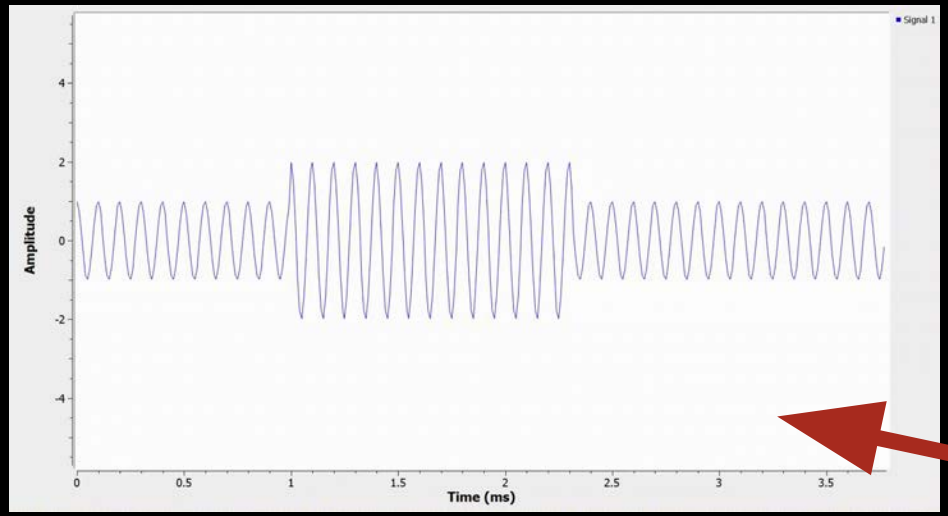


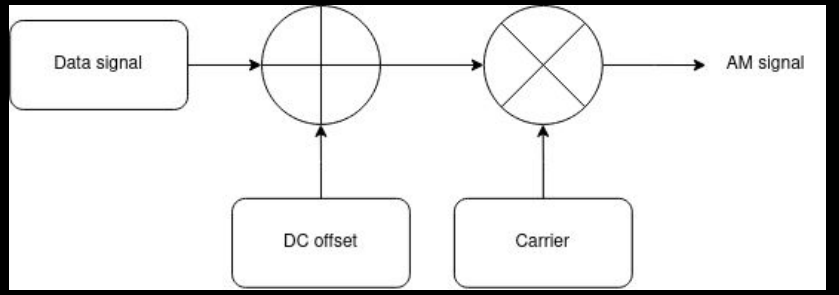


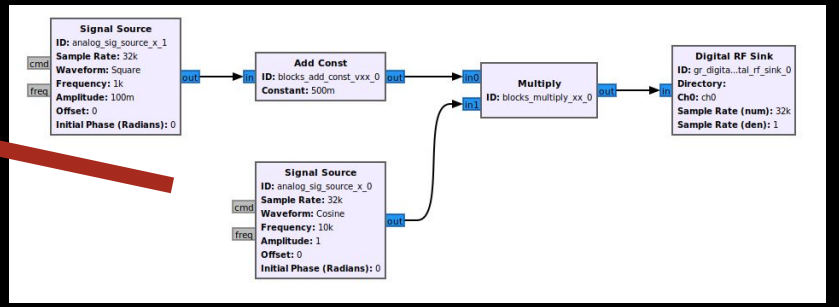
MIT HAYSTACK OBSERVATORY

GNURadio THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

GNURadio: Open Source Signal Processing





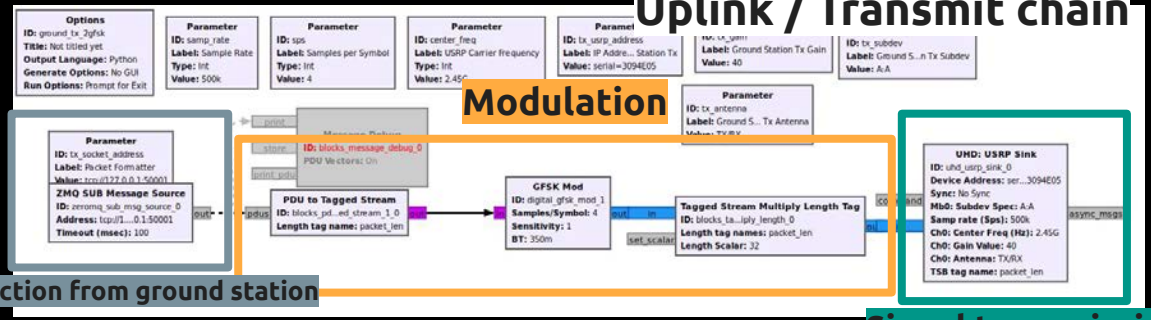


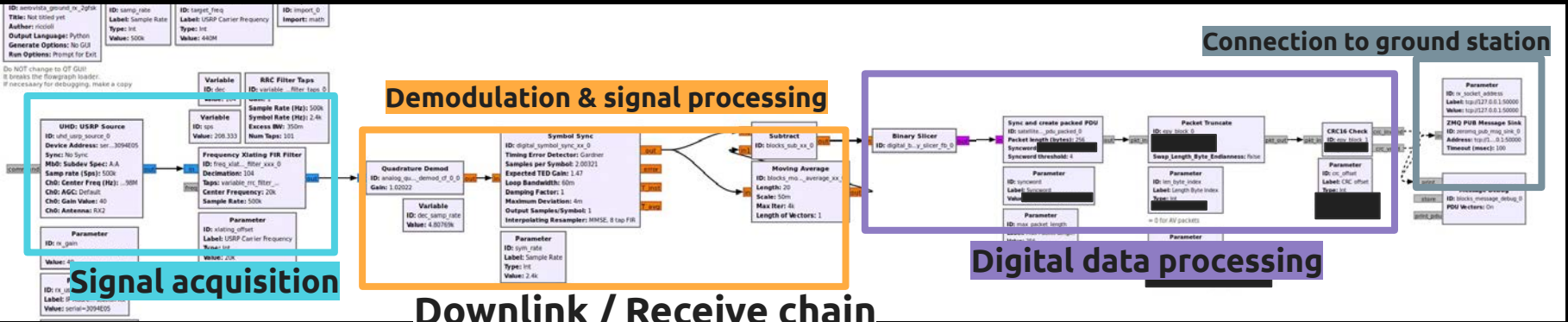
MIT HAYSTACK OBSERVATORY

GNURadio
THE FREE & OPEN SOFTWARE RADIO ECOSYSTEM

Current GNURadio Implementations

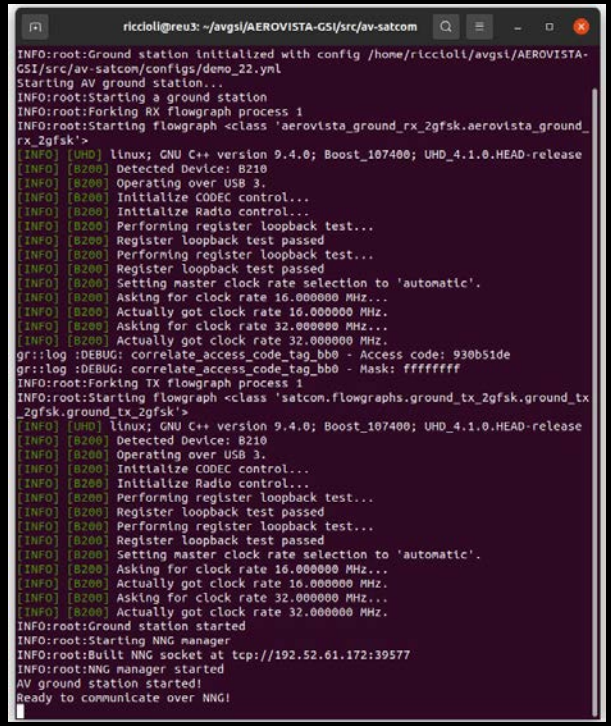
Uplink / Transmit chain





Ground Station Software

- Extension of development of REU 2021 project
- Needs to…
  - Transmit and modulate
  - Receive and demodulate
  - Packetize and depacketize
  - Accept user input or preprogrammed commands asynchronously
  - Facilitate communication between several programs and data sources, over multiple protocols
  - Be as configurable and flexible as possible

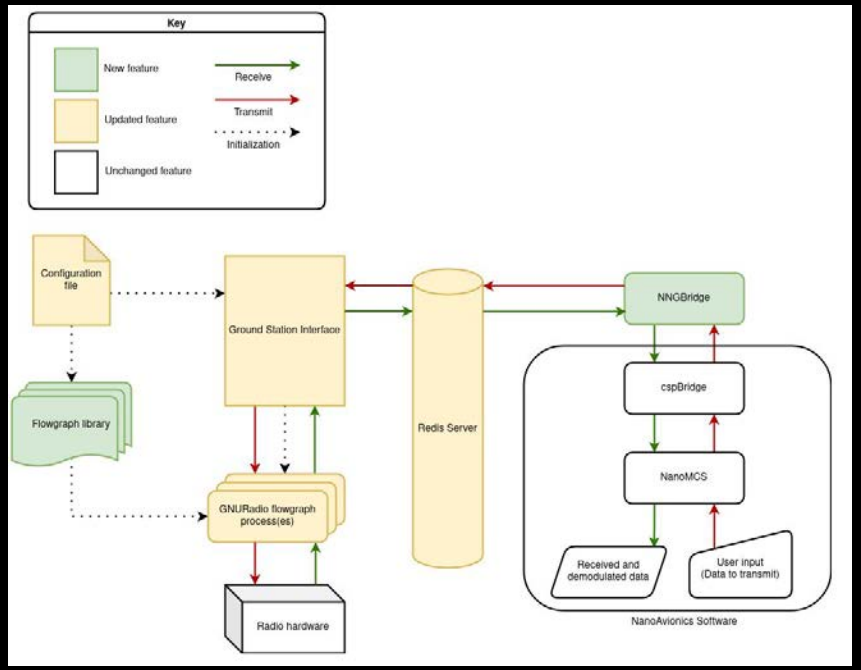


AERO-VISTA-GS startup readout

MIT HAYSTACK OBSERVATORY

Ground Station Architecture

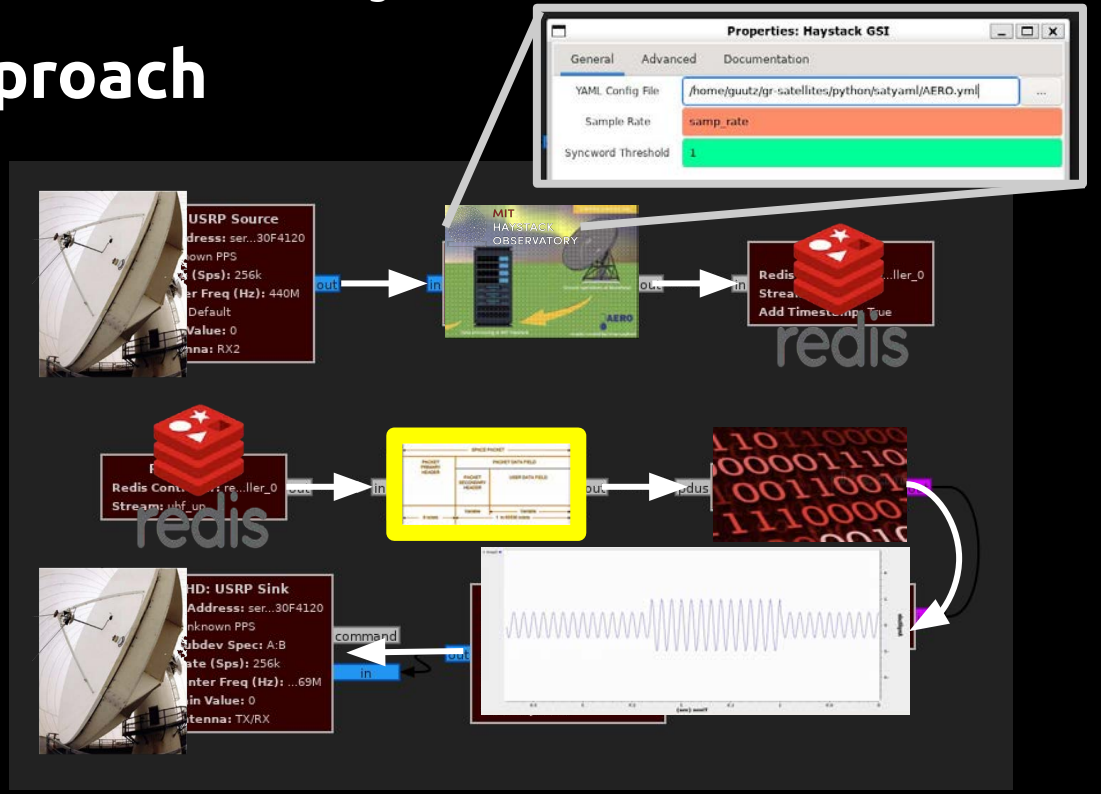
- Config file lists interface and flowgraph initial conditions
- Redis server provides remote access to ground station & allows data stream queuing
- Interface starts and handles GNURadio flowgraphs
- Flowgraph processes perform mod/demod, operate radio hardware
- NNGBridge facilitates communication between Redis server and mission-level software




AERO/VISTA Overview
Communications Background
The Master Plan: Haystack Satcom
Full Stack Walkthrough
What’s Next
Reflections

A Flowgraph-centric Approach

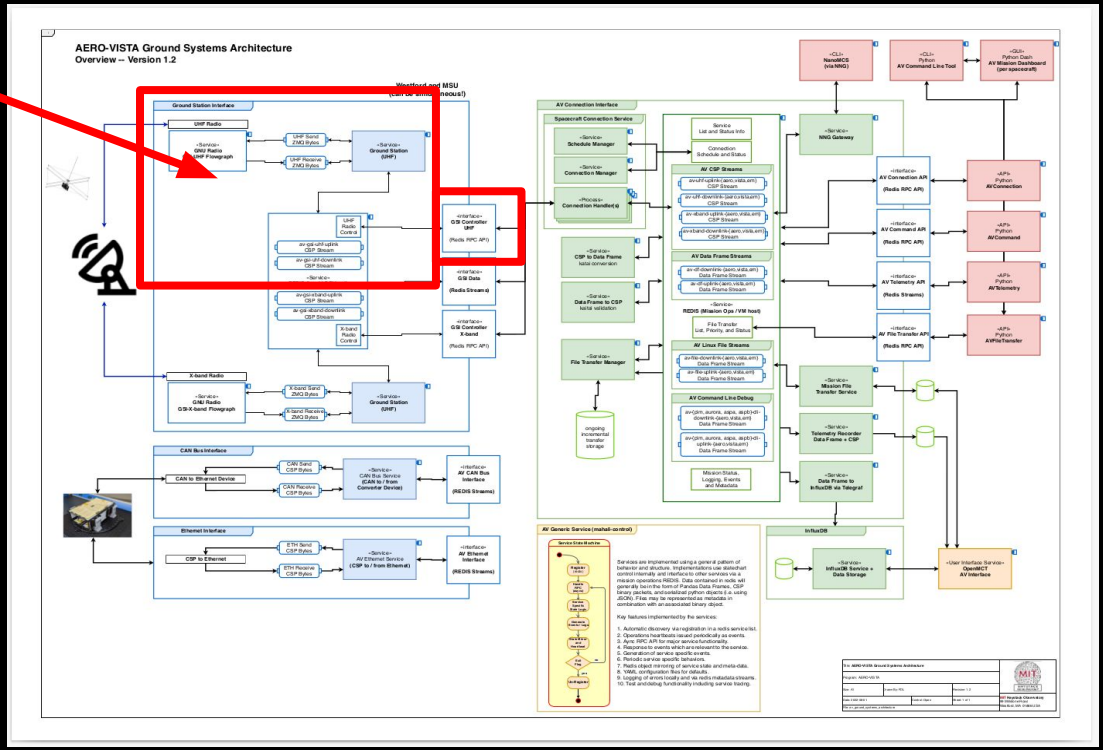
- Pros:
  - Utilizes an array of pre-written communications software
  - No need to reimplement atomic-level processes
- Cons:
  - Configurable parameters must be hard-coded before deployment
  - Less quality control of each line of code



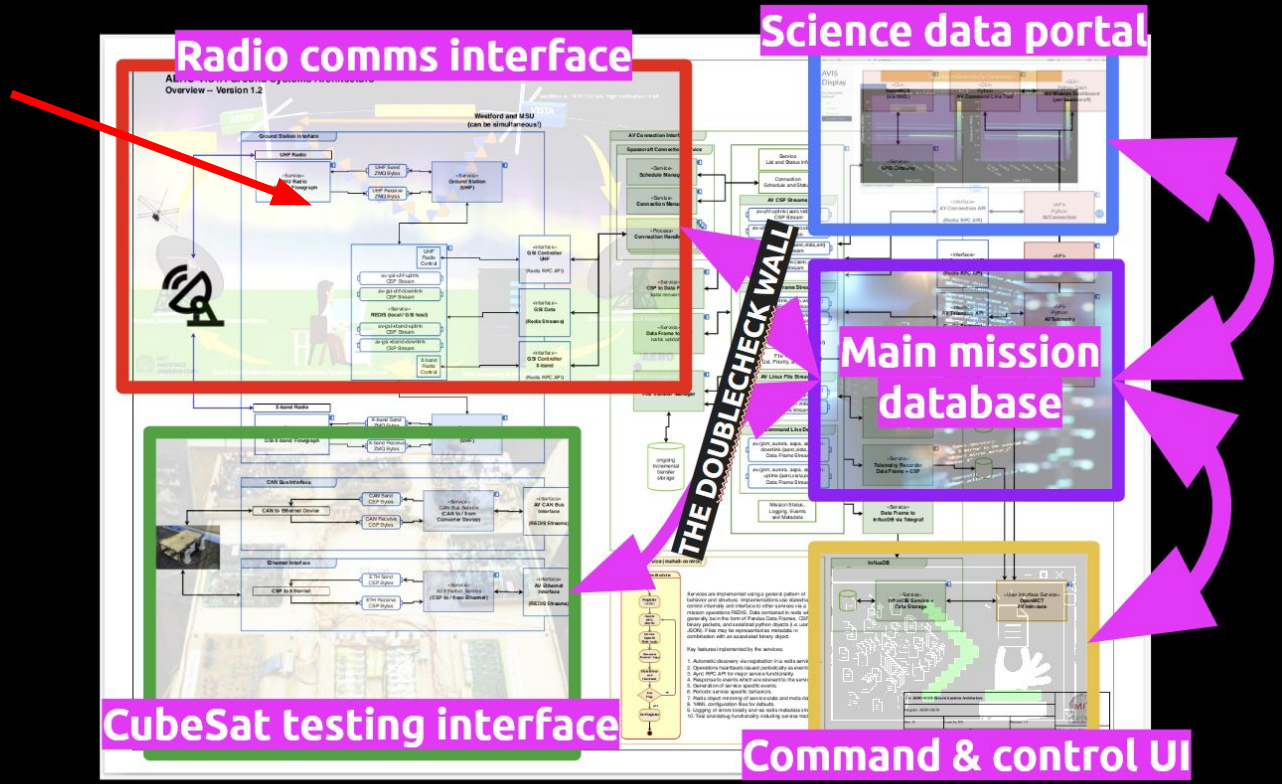
MIT HAYSTACK OBSERVATORY

The "Big Picture" at a glance

That's us!

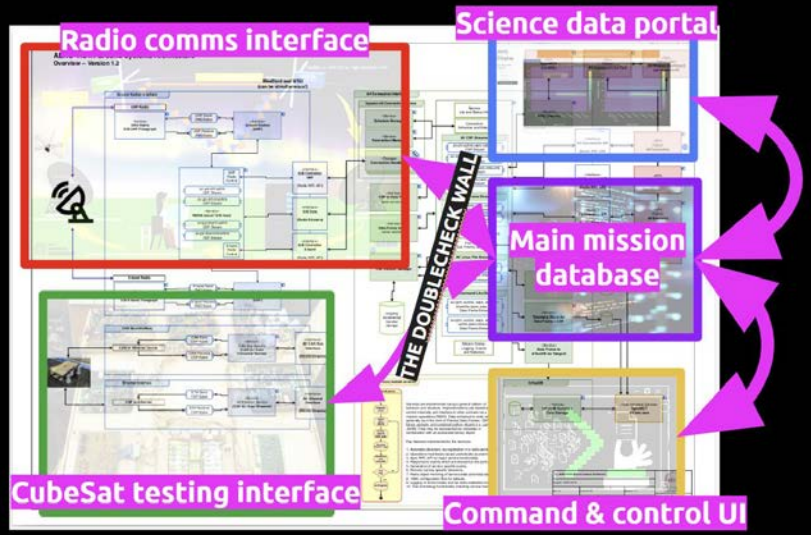


The "Big Picture" at a glance (simplified)



Next Steps

- Immediate:
  - Flowgraph vs Python architecture
  - Test GS implementation on Westford hardware
  - Introduce GNURadio signal processing for LEO
- Future:
  - Expand flowgraph library
  - Acknowledgement (ACK) signal detection / response
  - Full UX experience
  - Porting our GNURadio blocks to C++



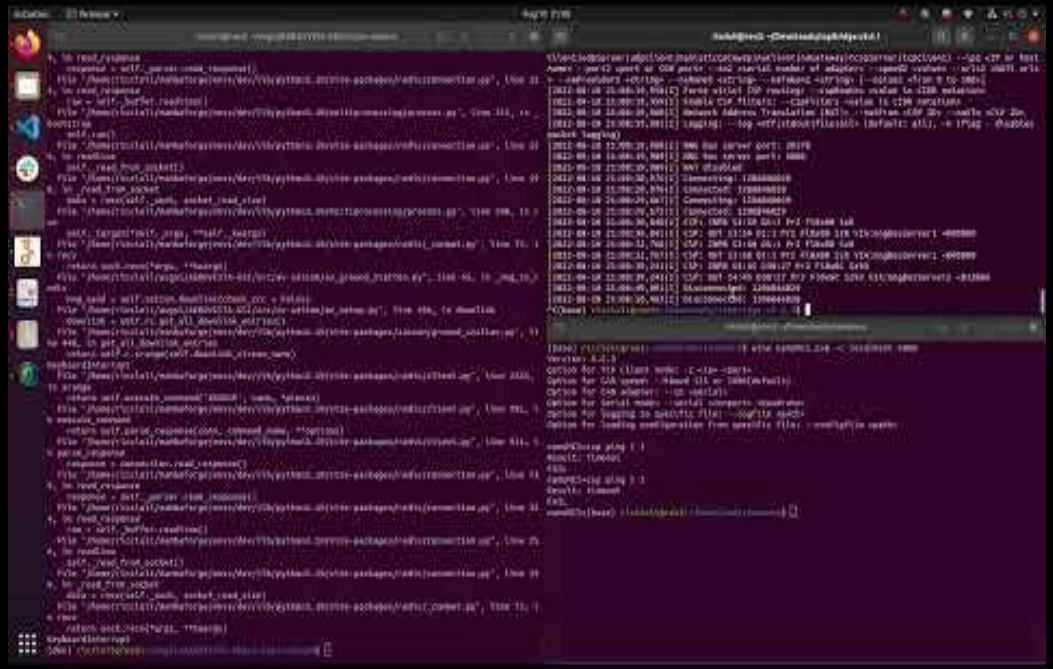
Frank Lind

Demo

Ground Station →

Watch for:

- Flowgraph traffic
- System spin-up



← cspBridge

Watch for:

- Packet traffic

← NanoMCS

Watch for:

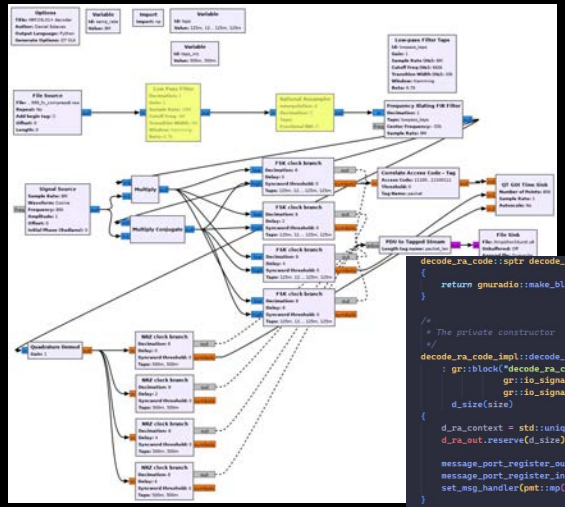
- Packet data input

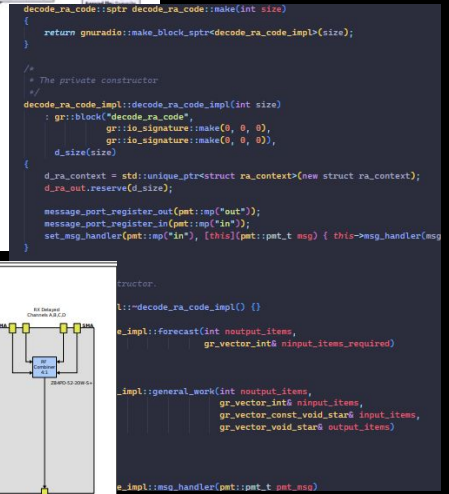
MIT
HAYSTACK
OBSERVATORY

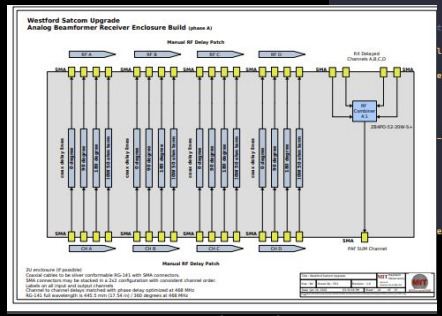

AERO/VISTA Overview
Communications Background
The Master Plan: Haystack Satcom
Full Stack Walkthrough
What’s Next
Reflections

Project Challenges

- Radio
  - Digital modulation
  - IQ & complex wave representation
- Programming
  - Linux
  - OOP
  - GNURadio
- Lab time
  - Couldn't verify GS software on actual satellite/Westford hardware







Frank Lind

MIT HAYSTACK OBSERVATORY

Thank you!

Questions?

A-V REU 2022 mentors: Mary Knapp, John Swoboda, Ryan Volz, Tobias Gedenk
A-V REU 2022 student collaborators: Allen Chang, Alexis Lupo

MIT
HAYSTACK
OBSERVATORY

Resources and Further Reading

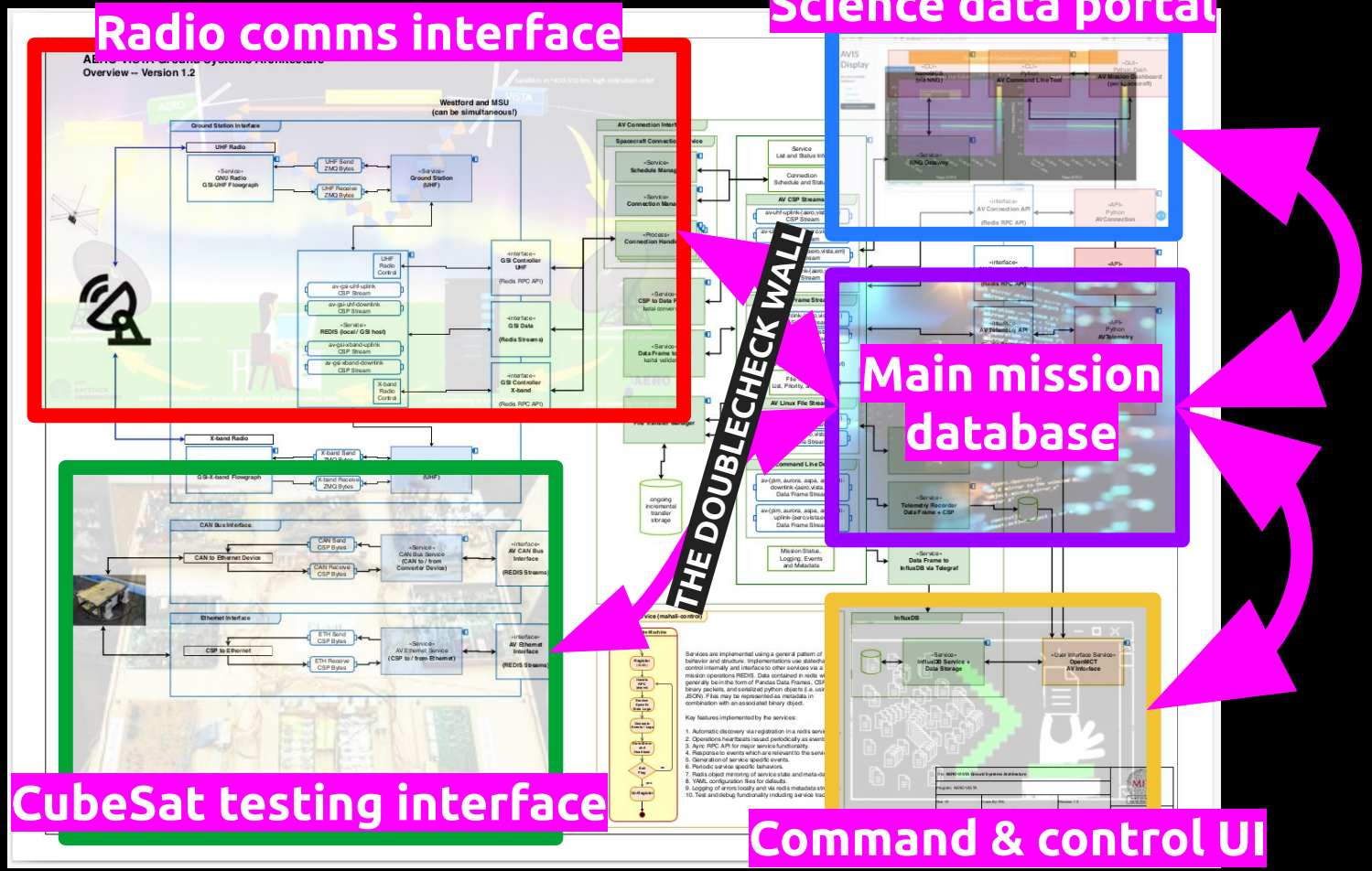
Main code repository: https://github.mit.edu/AEROVISTA/Haystack-GSI

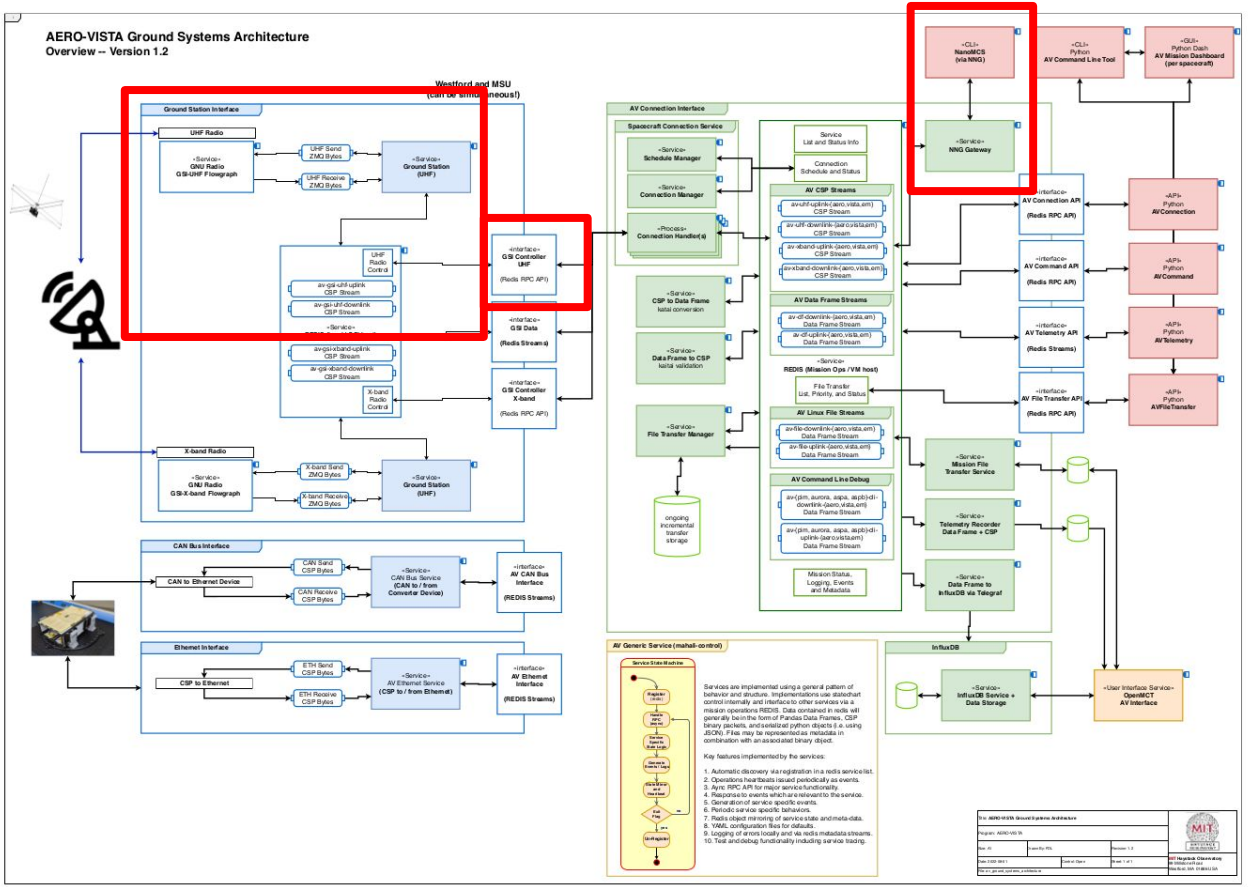
Haystack GNURadio plugin: https://github.mit.edu/AEROVISTA/gr-haystack [internal]

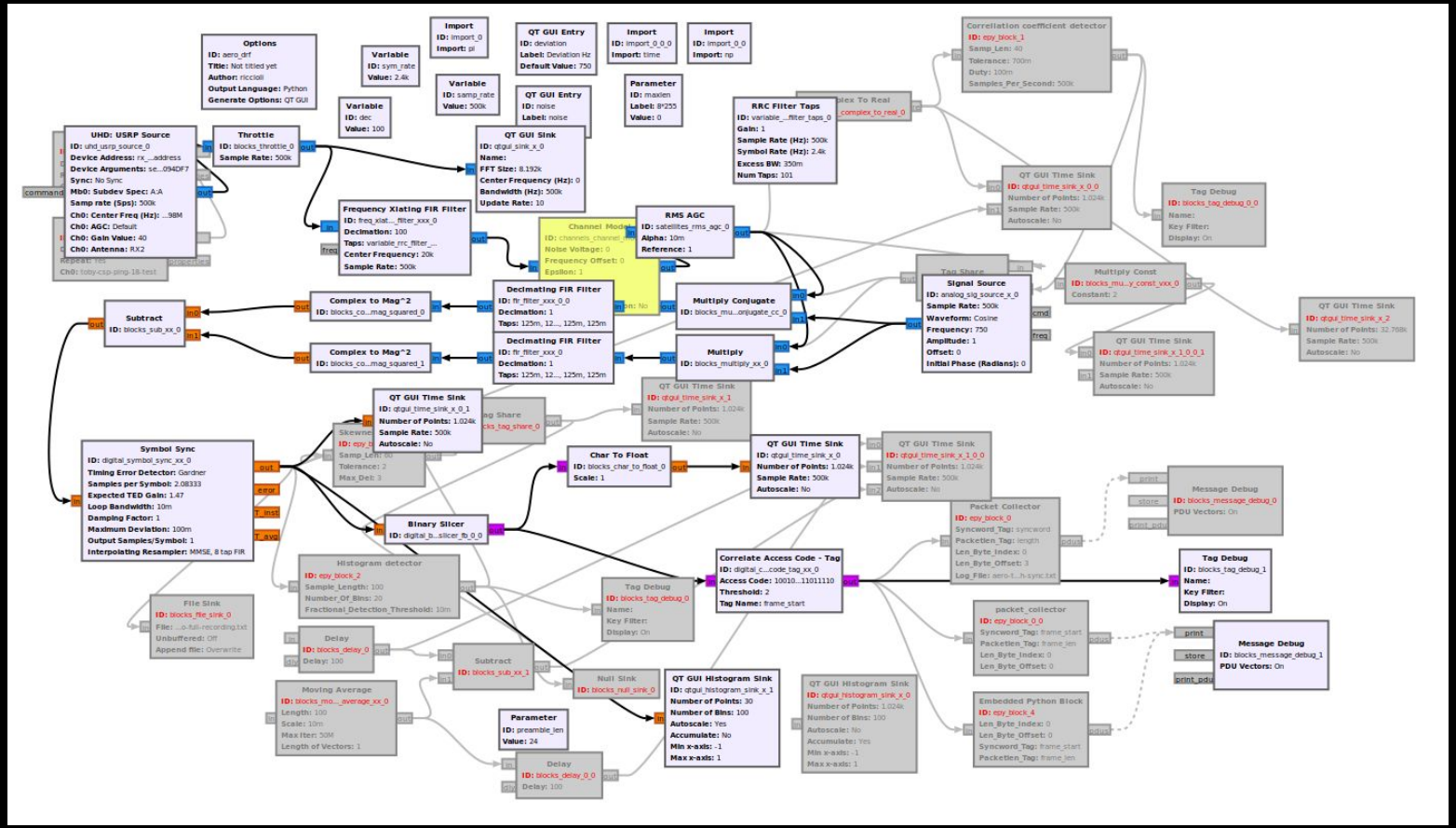
AERO: Auroral Emission Radio Observer (2018 Erickson et al.)

AERO & VISTA: Demonstrating HF Radio Interferometry with Vector Sensors (2019 Lind et al.)

Using GNURadio for CubeSat ground station ops: Decoding images from AMICal Sat – Daniel Estévez


Radio comms interface
Science data portal
Main mission database
THE DOUBLECHECK WALL
CubeSat testing interface
Command & control UI


AERO-VISTA Ground Systems Architecture
Overview -- Version 1.2
Westford and MSU
(can be simultaneous!)
Ground Station Interface
UHF Radio
«Service»
GNU Radio
GSI-UHF Flowgraph
UHF Send
ZMQ Bytes
UHF Receive
ZMQ Bytes
«Service»
Ground Station
(UHF)
UHF
Radio
Control
av-gsi-uhf-uplink
CSP Stream
av-gsi-uhf-downlink
CSP Stream
«Service»
av-gsi-xband-uplink
CSP Stream
av-gsi-xband-downlink
CSP Stream
X-band
Radio
Control
«interface»
GSI Controller
UHF
(Redis RPC API)
«interface»
GSI Data
(Redis Streams)
«interface»
GSI Controller
X-band
(Redis RPC API)
X-band Radio
«Service»
GNU Radio
GSI-X-band Flowgraph
X-band Send
ZMQ Bytes
X-band Receive
ZMQ Bytes
«Service»
Ground Station
(UHF)
CAN Bus Interface
CAN to Ethernet Device
CAN Send
CSP Bytes
CAN Receive
CSP Bytes
«Service»
CAN Bus Service
(CAN to / from
Comestar Device)
«interface»
AV CAN Bus
Interface
(REDIS Streams)
Ethernet Interface
CSP to Ethernet
ETH Send
CSP Bytes
ETH Receive
CSP Bytes
«Service»
AV Ethernet Service
(CSP to / from Ethernet)
«interface»
AV Ethernet
Interface
(REDIS Streams)
AV Connection Interface
Spacecraft Connection Service
«Service»
Schedule Manager
«Service»
Connection Manager
«Process»
Connection Handler(s)
Service
List and Status Info
Connection
Schedule and Status
AV CSP Streams
av-uhf-uplink-(aero,vista,em)
CSP Stream
av-uhf-downlink-(aero/vista/em)
CSP Stream
av-xband-uplink-(aero,vista,em)
CSP Stream
av-xband-downlink-(aero,vista,em)
CSP Stream
«Service»
CSP to Data Frame
kaitai conversion
«Service»
Data Frame to CSP
kaitai validation
AV Data Frame Streams
av-df-downlink-(aero,vista,em)
Data Frame Stream
av-df-uplink-(aero,vista,em)
Data Frame Stream
«Service»
REDIS (Mission Ops /VM host)
File Transfer
List, Priority, and Status
«Service»
File Transfer Manager
ongoing
incremental
transfer
storage
AV Linux File Streams
av-file-downlink-(aero,vista,em)
Data Frame Stream
av-file-uplink-(aero,vista,em)
Data Frame Stream
AV Command Line Debug
av-(pim, aurora, aapa, aspb)-idl-
downlink-(aero,vista,em)
Data Frame Stream
av-(pim, aurora, aapa, aspb)-idl-
uplink-(aero/vista/em)
Data Frame Stream
Mission Status,
Logging, Events
and Metadata
«CLI»
NanoMCS
(via NNG)
«Service»
NNG Gateway
«CLI»
Python
AV Command Line Tool
«GUI»
Python Dash
AV Mission Dashboard
(per spacecraft)
«interface»
AV Connection API
(Redis RPC API)
«interface»
AV Command API
(Redis RPC API)
«interface»
AV Telemetry API
(Redis Streams)
«interface»
AV File Transfer API
(Redis RPC API)
«API»
Python
AVConnection
«API»
Python
AVCommand
«API»
Python
AVTelemetry
«API»
Python
AVFileTransfer
«Service»
Mission File
Transfer Service
«Service»
Telemetry Recorder
Data Frame + CSP
«Service»
Data Frame to
InfluxDB via Telegraf
InfluxDB
«Service»
InfluxDB Service +
Data Storage
«User Interface Service»
OpenMCT
AV Interface
AV Generic Service (mahali-control)
Service State Machine
Services are implemented using a general pattern of
behavior and structure. Implementations use statechart
control internally and interface to other services via a
mission operations REDIS. Data contained in redis will
generally be in the form of Pandas Data Frames, CSP
binary packets, and serialized python objects (i.e. using
JSON). Files may be represented as metadata in
combination with an associated binary object.
Key features implemented by the services:
1. Automatic discovery via registration in a redis service list.
2. Operations heartbeats issued periodically as events.
3. Async RPC API for major service functionality.
4. Responses to events which are relevant to the service.
5. Generation of service specific events.
6. Periodic service specific behaviors.
7. Redis object mirroring of service state and meta-data.
8. YAML configuration files for defaults.
9. Logging of errors locally and via redis metadata streams.
10. Test and debug functionality including service tracing.



MIT
HAYSTACK
OBSERVATORY

Specific changes

- Flowgraph library
  - Brought out more config options
- NNGBridge
  - Main pathway for hardware testing
- Improved packet parsing
  - Overlapping packet detection
  - Packet CRC check
- Software health logging
- Flowgraph-level signal processing improvements

MIT
HAYSTACK
OBSERVATORY