Jet Propulsion Laboratory
California Institute of Technology

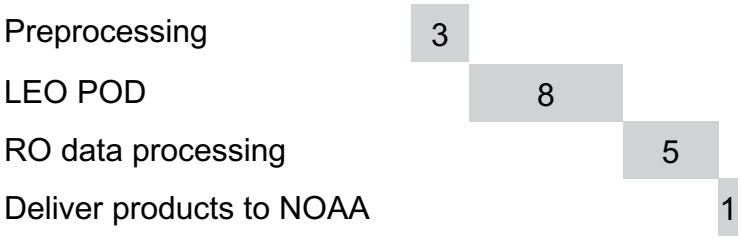
Towards near-real-time radio occultation processing operations for weather forecasting applications

Chad Galley, Byron Iijima, Yoaz Bar-Sever
Jet Propulsion Laboratory, California Institute of Technology

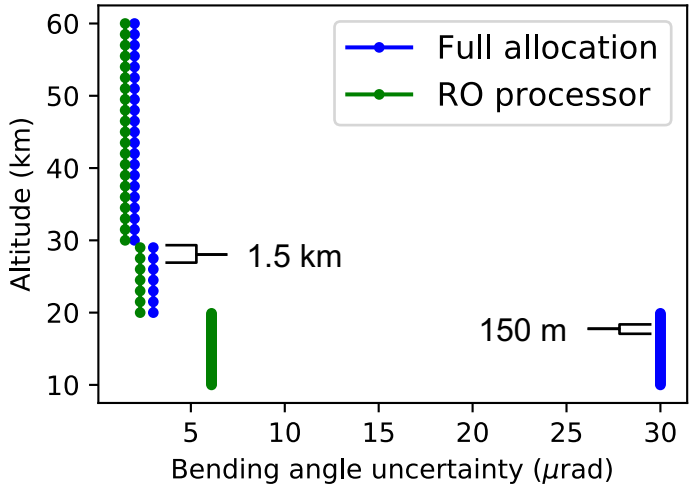
IROWG-7 Helsingor, Denmark, September 19-25, 2019

Sentinel-6/Jason-CS RO NRT processing at JPL

- JPL is responsible for processing Sentinel-6 RO occultations in near real-time (NRT)
  - EUMETSAT and ROM SAF are primary producers of the official non-time-critical (NTC) product
  - JPL will produce a secondary/validation NTC product
- Important requirements to be met:
  - 17 min from time that instrument data is received to availability of NRT BUFR product on WMO GTS



  - LEO POD accuracy:
    - Position: 10 cm/axis, RMS
    - Velocity: 0.1 mm/s, RMS, along track
  - 770 profiles per day (post-QC)
  - 94% availability over any one-month period
  - Bending angle uncertainties:



JPL GPS Occultation Analysis Software (GOAS)

- Used by JPL RO processing group to make science products for atmosphere/climate research

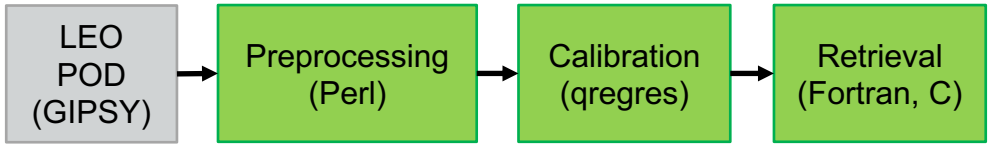
- Implements some of the original RO processing algorithms [e.g., see Hajj et al (2001)]
  - Fjeldbo's geometric optics retrieval algorithm
  - Standard Abel inversion
  - No statistical optimization used for bending angle retrieval

- Processes multi-mission data including (but not limited to):
  - CHAMP
  - SAC-C
  - COSMIC
  - GRACE
  - TerraSAR-X
  - TanDEM-X
  - KOMPSAT-5
  - PAZ
  - GRACE-FO

- Features:
  - Handles occultations tracked with closed-loop, open-loop, or both
  - Calibrates links using zero-, single-, or double-differencing
  - Implements the canonical transform for low-altitude bending angle retrieval

JPL GPS Occultation Analysis Software (GOAS)

- Architecture:
  - Perl
  - Fortran 77 & 90
  - A tiny bit of C



- Dependencies:
  - JPL GIPSY libraries and POD software
  - JPL's qregres program to calibrate occultation and, if needed, clock and ground reference links
  - fftw2

- Limitations of GOAS for Sentinel-6 NRT processing
  - GPS only
  - LEO POD solutions provided in separate process
  - Serial processing
  - File-based processing
  - Depends on GIPSY software libraries – GIPSY is no longer maintained nor supported

- Conclusion: A new RO processing software is needed at JPL to meet Sentinel-6 NRT processing requirements

JPL GNSS Radio Occultation Atmospheric Retrieval Software (ROARS)

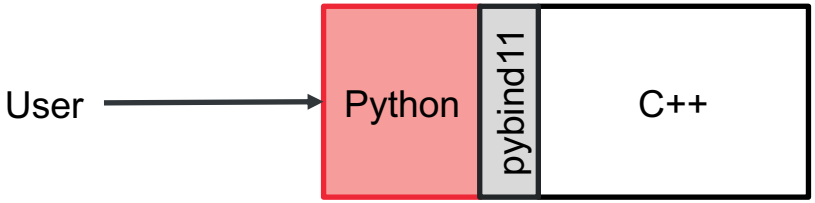
- Implements GOAS algorithms with flexibility to include more (e.g., phase matching)

- Supports multi-mission processing of current and future missions including:
  - Daily processing of GeoOptics data
  - NRT and daily processing of Sentinel-6 data
  - Daily processing of COSMIC-2 once data is publicly available

- Features:
  - Processes occultations from multiple GNSS constellations
  - Parallel processing of retrievals
  - Currently handles occultations tracked with open-loop; others planned for inclusion
  - Currently calibrates links using zero-differencing; others planned for inclusion
  - Implements the canonical transform for low-altitude bending angle retrieval
  - Interactive and scriptable to handle operational processing, trouble-shooting, debugging, algorithm development, receiver software assessments,...

JPL GNSS Radio Occultation Atmospheric Retrieval Software (ROARS)

- Architecture:
  - C++ for speed and organization of data into classes/types
  - Python3 for user interface (interactive and scripts)
  - pybind11 library for exposing C++ functionality to Python



- Dependencies:
  - JPL GCORE software libraries
  - JPL's RTGx software for LEO and GNSS POD
  - automate – JPL Python package for date/time handling, automation & system tasks (incl. with ROARS)
  - pybind11 (incl. with ROARS)
  - fftw3

- Easy to install -- uses CMake build system from a Python setup.py script

JPL GNSS ROARS: Objects

- Classes to hold and manipulate data:

`Earth`	`Spacecraft`	`CanonicalTransform`
`Orbit`	`Link`	`Refractivity`
`Clock`	`BendingAngle`	`NavBits`
`Attitude`	`GeometricOptics`	`...`

- Classes to facilitate mathematical operations on data (`roars.mathlib`)
  - Smoothers: `LocalPolyRegression, RunningMean, RunningMedian,...`
  - Interpolators: `LinearPoly, CubicSpline, PiecewiseConstant, Poly,...`
  - Fitters: `LinearLeastSquares, PolyLeastSquares,...`
  - Differentiators: `FiniteDifference, NoiseRobust,...`
  - Integrators: `Riemann, Trapezoidal, Simpson, GaussChebyshev,...`
- Classes to manage processing:

  `Olog, OccultationManager, ProcessingManager`

- Python classes to facilitate easy file I/0, object manipulations, and visualization:

  `LinkObjects, SpacecraftObjects, MeasObjects, BendingAngleObjects,...`

- Highly configurable
  - Accommodates all GNSS signal frequencies
  - Multiple reference ellipsoids available (WGS84, GRS80, user-defined,...)
  - Multiple gravity models available (EGM2010, OSU91a,...)
  - Processing strategy customized by an input tree

JPL GNSS ROARS: Flexible tree-based processing

- Tree-driven automated processing
  - One input file containing all input parameters, configurations, server/directory information, etc

```
BendingAngle:
    L1:
        FineGrained:
            Smoother:
                Name:                LocalPolyRegression
                Smooth:
                    Degree:          3
                    Window:          1.0
                Decimate:
                    Interval:        0.32
                    Begin:           500  # Number of points, not seconds
            StaticSpacecraft:        Transmitter
            GeometricOptics:
                MaxImpactParameter:  120e3   # [m]
                MaxIterations:       30
            CanonicalTransform:
                LimbHeight:          0.0
                ScreenInterval:      1.0  # [m]
        CoarseGrained:
            Smoother:
                Name:                LocalPolyRegression
                Smooth:
                    Degree:          3
                    Window:          2.0
                Decimate:
                    Interval:        0.32
                    Begin:           500  # Number of points, not seconds
            StaticSpacecraft:        <BendingAngle.L1.FineGrained.StaticSpacecraft>
            GeometricOptics:         <BendingAngle.L1.FineGrained.GeometricOptics>
    L2:
        CoarseGrained:               <BendingAngle.L1.CoarseGrained>
```

JPL GNSS ROARS: Flexible tree-based processing

- Tree-driven automated processing
  - One input file containing all input parameters, configurations, server/directory information, etc

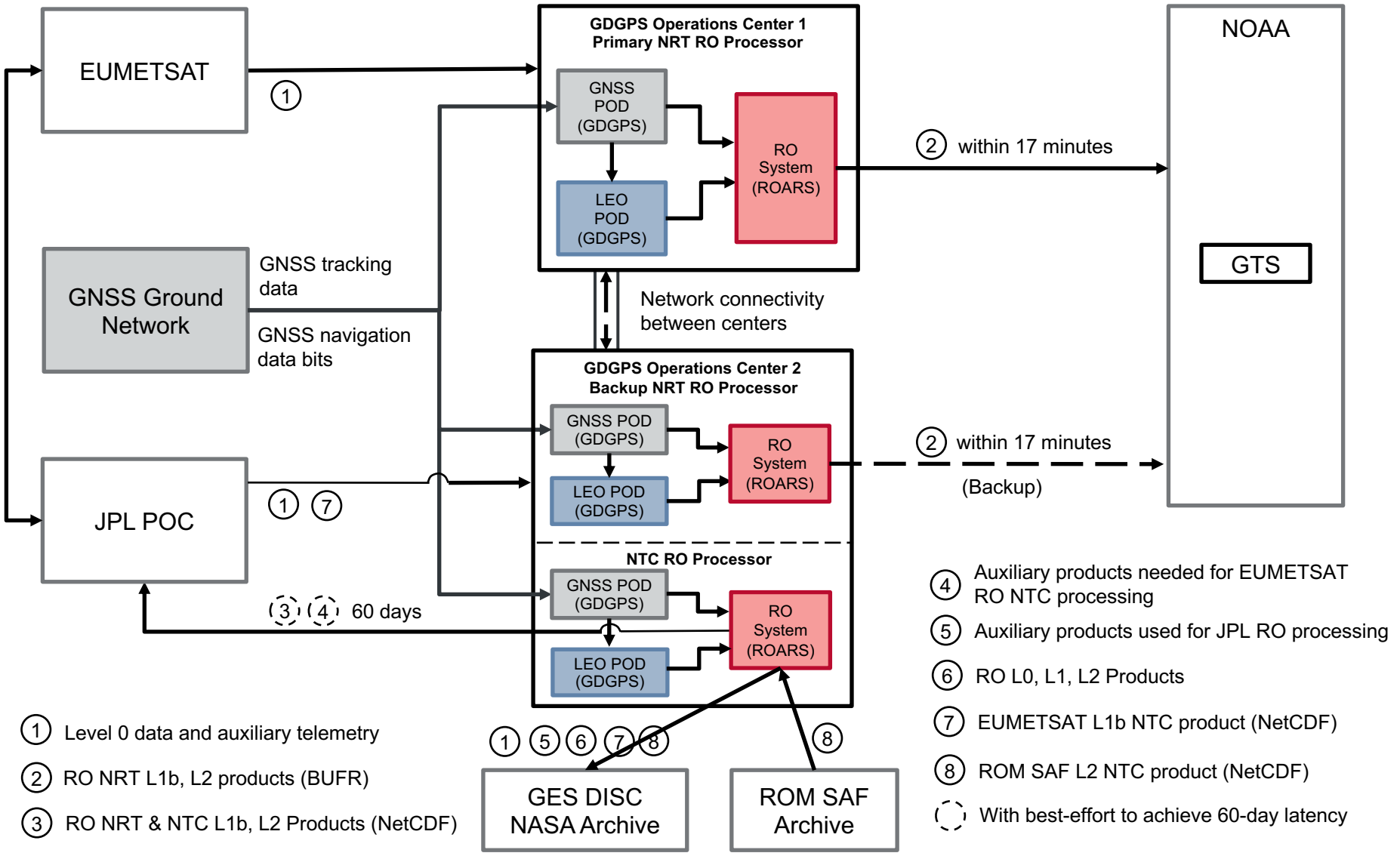
```
BendingAngle:
    L1:
        FineGrained:
            Smoother:
                Name:                LocalPolyRegression
                Smooth:
                    Degree:          3
                    Window:          1.0
                Decimate:
                    Interval:        0.32
                    Begin:           500  # Number of points, not seconds
            StaticSpacecraft:        Transmitter
            GeometricOptics:
                MaxImpactParameter:  120e3   # [m]
                MaxIterations:       30



        CoarseGrained:
            Smoother:
                Name:                LocalPolyRegression
                Smooth:
                    Degree:          3
                    Window:          2.0
                Decimate:
                    Interval:        0.32
                    Begin:           500  # Number of points, not seconds
            StaticSpacecraft:        <BendingAngle.L1.FineGrained.StaticSpacecraft>
            GeometricOptics:         <BendingAngle.L1.FineGrained.GeometricOptics>
    L2:
        CoarseGrained:               <BendingAngle.L1.CoarseGrained>
```

JPL GNSS ROARS: Flexible tree-based processing

- Tree-driven automated processing
  - One input file containing all input parameters, configurations, server/directory information, etc

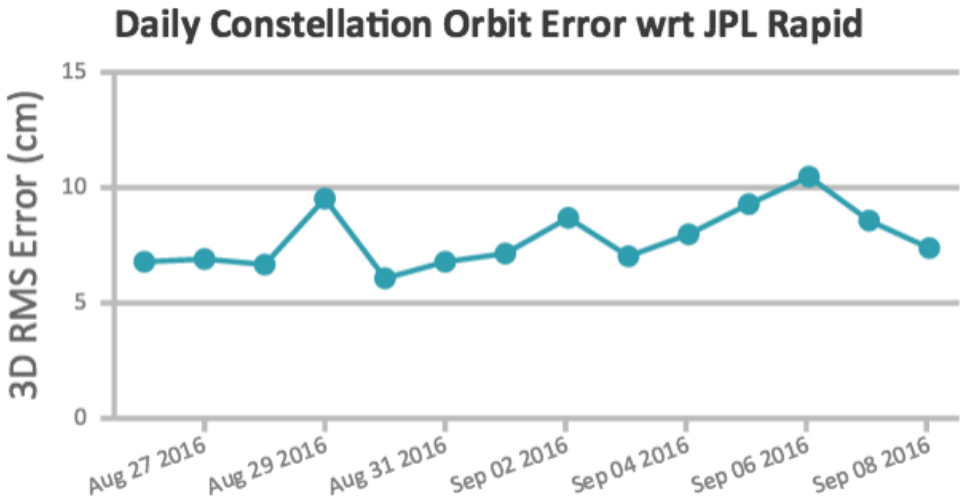
```
BendingAngle:
    L1:
        FineGrained:
            Smoother:
                Name:               LocalPolyRegression
                Smooth:
                    Degree:         3
                    Window:         1.0
                Decimate:
                    Interval:       0.32
                    Begin:          500  # Number of points, not seconds
            StaticSpacecraft:       Transmitter



            CanonicalTransform:
                LimbHeight:         0.0
                ScreenInterval:     1.0  # [m]
        CoarseGrained:
            Smoother:
                Name:               LocalPolyRegression
                Smooth:
                    Degree:         3
                    Window:         2.0
                Decimate:
                    Interval:       0.32
                    Begin:          500  # Number of points, not seconds
            StaticSpacecraft:       <BendingAngle.L1.FineGrained.StaticSpacecraft>
            GeometricOptics:        <BendingAngle.L1.FineGrained.GeometricOptics>
    L2:
        CoarseGrained:              <BendingAngle.L1.CoarseGrained>
```

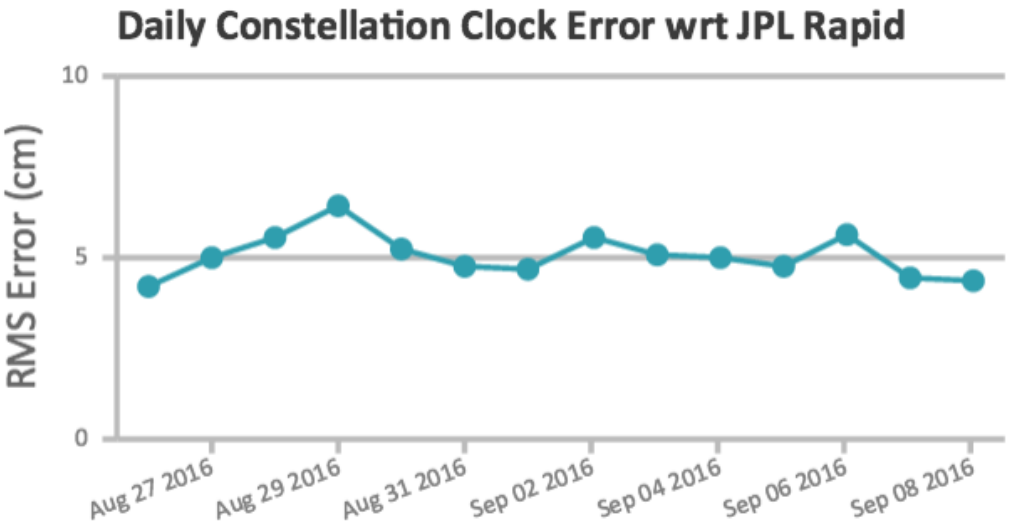
S-6/J-CS RO NRT processing architecture & data flow



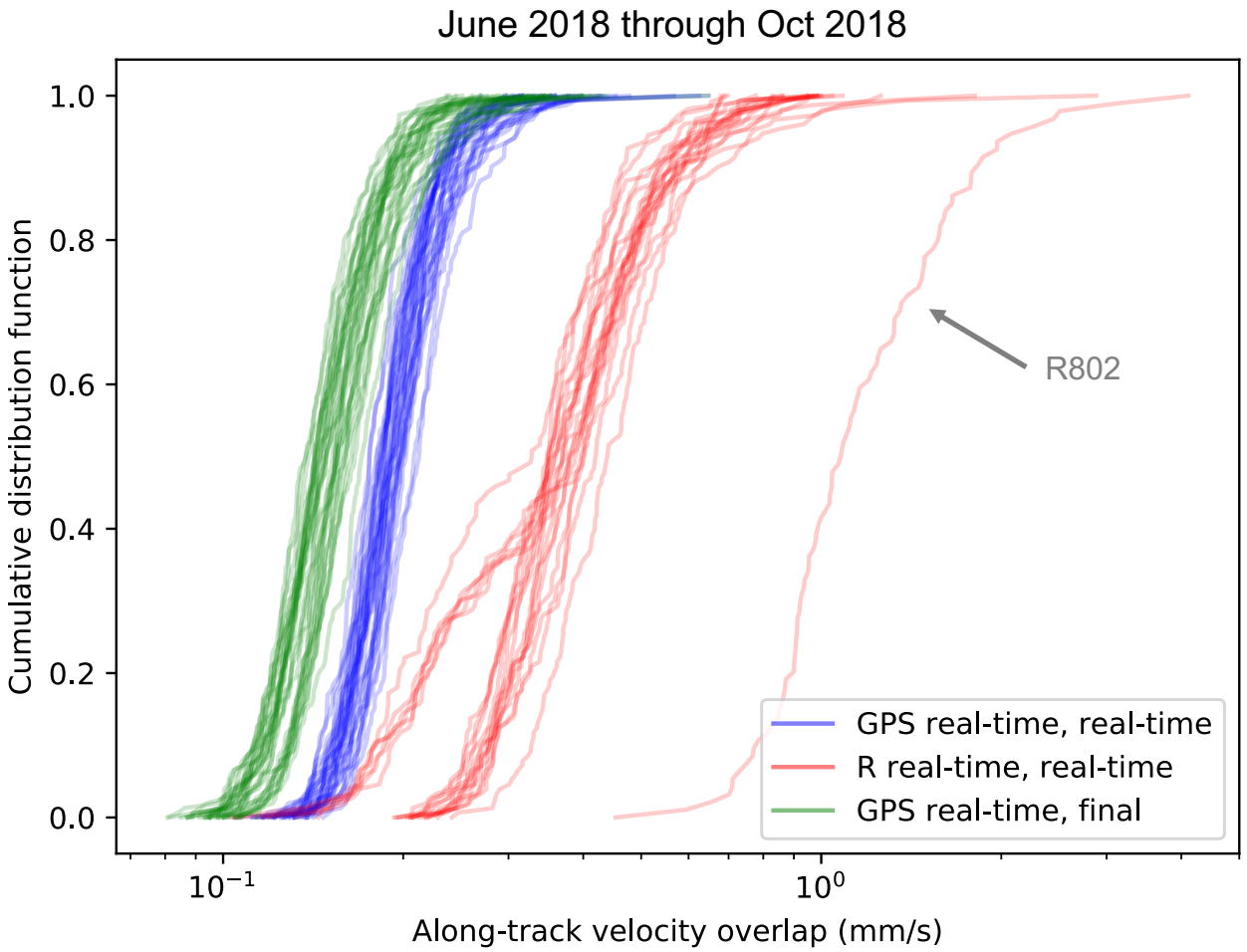
Real-time GPS POD at JPL GDGPS Operations Centers

GDGPS automated performance metrics from http://www.gdgps.net/products/orbit-clock-states.html

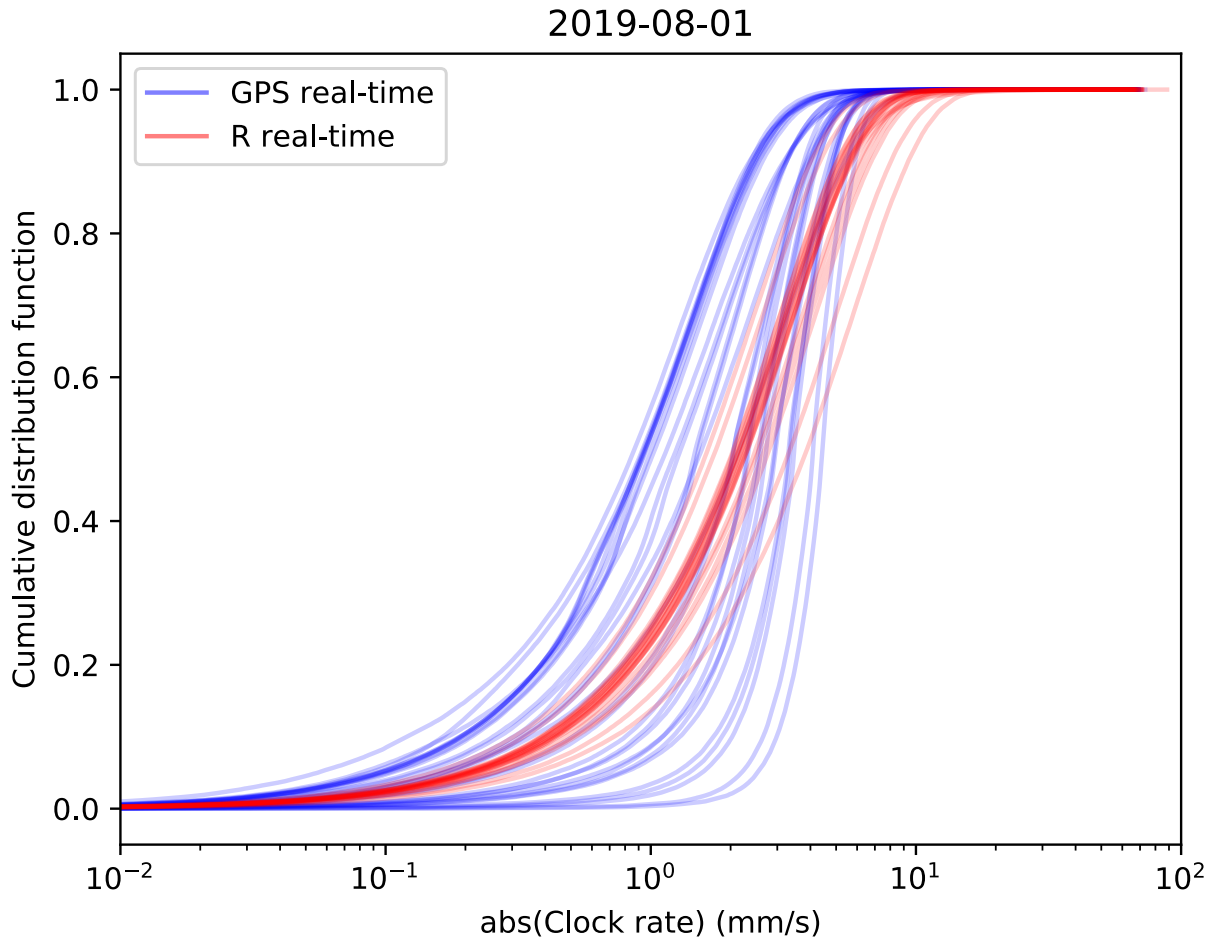




GNSS orbital velocity overlaps

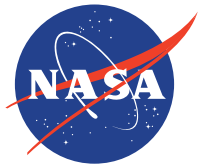


GNSS real-time clock bias rates



Summary

- New GNSS RO processing software (ROARS) being developed at JPL
  - Leverages time-tested algorithms from JPL's GOAS with the flexibility to add more
- ROARS:
  - supports both NRT processing for weather applications and daily processing for science applications
  - is flexible for building mission-specific processing packages
  - processes occultations from multiple GNSS constellations
  - will be implemented with parallelization in a hot-redundant operating environment to support Sentinel-6 high-availability and timeliness requirements
- RO NRT processing at JPL utilizes:
  - JPL's RTGx software for LEO POD
  - JPL's GDGPS operational service for real-time GNSS POD products, navigation data bits, and high-availability operational environment
- ROARS tested operationally processing GeoOptics data
- Sentinel-6/Jason-CS RO products generated by the JPL ROARS system:
  - NRT L1b and L2 products distributed on GTS in BUFR format
  - NTC L1 and L2 products will be available from ROM SAF and NASA's GES DISC archive
  - See the "Sentinel-6 Project GNSS-RO Product Description Document"



Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov