Extending VLBI Station Capabilities for Delta-DOR and Navigation via Open-Source RDEF Software Compliant with CCSDS Standards

Andrea Togni[1], David Mayo[1], Bradford Arnold[2], Sami Asmar[2], Marc Sanchez-Net[2], Benjamin Malphrus[1]

[1] Space Science Center, Morehead State University, Morehead, KY, USA
[2] Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

1 MOTIVATION

Delta-DOR (Delta Differential One-way Ranging) applies the VLBI technique to spacecraft navigation: two stations simultaneously record narrowband tones transmitted by a spacecraft and broadband emission from a nearby quasar, then difference the measured delays to determine the spacecraft's angular position in the plane of the sky.

Interagency Delta-DOR requires raw data exchange between stations and correlators from different agencies. The CCSDS Raw Data Exchange Format [1] standardizes this: each product file carries one second-resolution records containing complex baseband samples, down-conversion polynomials, and time-tags – making it self-coherent for both cross-station correlation and single-dish radiometric tracking.

Deep-space navigation demand is growing beyond current DSN capacity [2], and VLBI stations already operate the required infrastructure: large antennas, cryogenic receivers, and hydrogen masers. VLBI stations have already contributed single-dish Doppler observables to spacecraft navigation [3], and validated CCSDS-compliant software would enable the next step: their direct contribution of Delta-DOR angular measurements to the global tracking capacity.

RDEFpy is an open-source Python library that provides performance-optimized functions to read, process, and produce CCSDS-compliant Delta-DOR products from standard VLBI operations. It bridges the gap between VLBI data acquisition and the navigation data pipeline, enabling any station to generate interoperable RDEF products.

RDEFpy is internally and externally validated through three independent tiers:

1. Data fidelity: Synthetic signals are roundtripped and tested through all supported bit depths.
2. Format compliance: Binary output is verified byte-by-byte independently of RDEFpy's own reader.
3. Interoperability: A real S-band RDEF file from JPL's SPR at DSS-17 is read, written, and read back. All header fields and I/Q samples are bit-identical to the original.

2 RDEF Structure

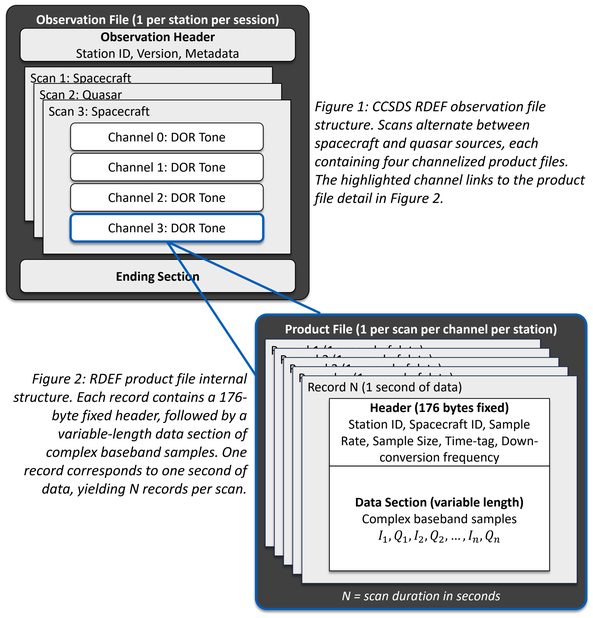

Figure 1: CCSDS RDEF observation file structure. Scans alternate between spacecraft and quasar sources, each containing four channelized product files. The highlighted channel links to the product file detail in Figure 2.

Figure 2: RDEF product file internal structure. Each record contains a 176-byte fixed header, followed by a variable-length data section of complex baseband samples. One record corresponds to one second of data, yielding N records per scan.

3 USAGE EXAMPLES

Inspect all Files in the Experiment

```python
from rdefpy import read_observation, read_product

obs = read_observation("observation_file.obs")
for scan in obs:
    for entry in scan.products:
        pf = read_product(entry.datafile)
        df = pf.to_dataframe(include_frequency=True)
```

Listing 1: Querying a multi-scan observation file. The observation file indexes all product files by scan and channel: each product file can be loaded individually and exported to a DataFrame for downstream analysis.

Extract Carrier or Tone

```python
from rdefpy import read_product

pf = read_product("product_file.prd")
freq, pc_n0 = pf.get_carrier(pc_n0=True)
```

Listing 2: Reading a product file and extracting carrier observables. The get_carrier method returns time tags, baseband and sky frequencies, and carrier power-to-noise spectral density (Pc/N0) at the requested integration time.

Conversion from Baseband Stream

```python
import numpy as np
from rdefpy import ComplexSignal, write_product

i, q = np.loadtxt("iq_samples.csv", delimiter=",", unpack=True)

sig = ComplexSignal(data=i + 1j * q, sample_rate=2e6,
                    rf_to_if_downconv=8.4e9,
                    if_to_channel_downconv=20.432e6)
pf = sig.to_product_file(sample_size=8, station_id=24,
                         spacecraft_id=85, time_tag_year=2026,
                         time_tag_doy=77, time_tag_start_second=43200.0)
write_product("product_file.prd", pf)
```

Listing 3: Writing an RDEF product file from in-phase and quadrature samples. Raw I/Q data are wrapped in a ComplexSignal with the station down-conversion chain, then serialized to a CCSDS 506.1-B-1 compliant product file.

4 DEPLOYMENT IN RADIOMETRIC TRACKING

RDEFpy is operationally deployed at DSS-17 (Morehead State University 21-m antenna) for X- and S-band spacecraft tracking. Carrier frequency estimates of LRO from RDEF product files agree with TT&C modem measurements. This enables open-loop recording stations to extract navigation-grade Doppler observables using standard Python tools.

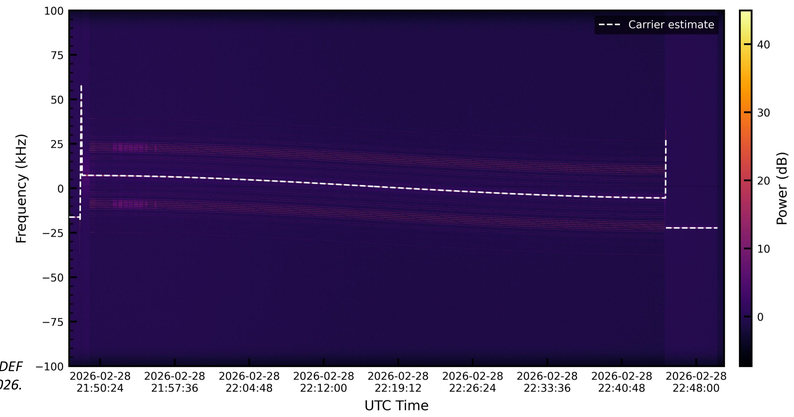

Figure 3: Spectrogram and carrier estimate extracted from RDEF product files of LRO S-Band signal recorded at DSS-17 on 28 Feb 2026.

5 PERFORMANCE

For navigation and radio-science bandwidths (1-100 kHz), file reading and carrier estimation complete in seconds, even for multi-hour recordings. Wideband VLBI (100 MHz, 2-bit) processes a full scan in under 8 minutes on a standard workstation. A Cython back-end provides up to 13.5x speedup for 2-bit unpacking; a pure-NumPy fallback requires no compilation.

Scenario	Sample Rate	Duration	File Size	Reading	Tone Extraction	Cython Speedup
Navigation	1 kHz 16-bit	6 h	90 MB	0.30 s	0.77 s	1.1x 1.4x
Radio-science	100 kHz 16-bit	2 h	2.9 GB	2.3 s	46.7 s	1.3x 1.3x
VLBI wideband	100 MHz 2-bit	60 s	3.0 GB	6.6 s	489.5 s	13.5x 1.1x

Table 1. I/O and carrier estimation benchmarks across three operational scenarios (Cython back-end).

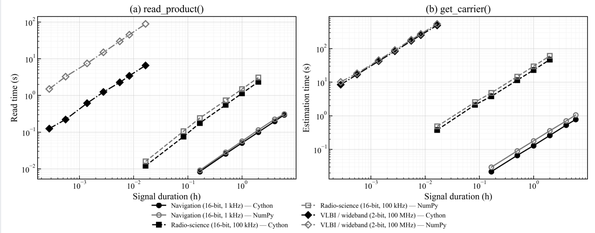

Figure 4. Read and carrier estimation time vs. signal duration. Filled/open markers distinguish Cython and NumPy back-ends; marker shapes denote the three scenarios from Table 1.

6 AVAILABILITY

- RDEFpy v1.0 is available on GitHub upon request (e-mail authors), compatible with Python 3.12+.
- Community feedback and testing are welcomed.
- Companion journal paper in preparation; submission expected June 2026.

Contact: antogni@moreheadstate.edu

References
1.CCSDS 506.1-B-1 – CCSDS, "Delta-DOR Raw Data Exchange Format," Recommended Standard, Issue 1, CCSDS 506.1-B-1, June 2013.
2.NASA OIG, 2023 – NASA Office of Inspector General, "Audit of NASA's Deep Space Network," Report No. IG-23-016, July 12, 2023.
3.Molera Calvés et al., 2021 – Molera Calvés, G., Pogrebenko, S.V., et al., "High spectral resolution multi-tone Spacecraft Doppler tracking software: Algorithms and implementations," Publications of the Astronomical Society of Australia, 38, e065, 2021.