NIST Technical Note
NIST TN 2359

Citizen's Broadband Radio Service (CBRS) Sharing Ecosystem Assessment (SEA) Sensor Data Manual

A guide to using and understanding sensor data produced by the NASCTN CBRS SEA program

Aric W. Sanders
Anthony W. Romaniello
Daniel G. Kuester
Todd R. Schumann
Adam Wunderlich

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2359

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Technical Note
NIST TN 2359

Citizen's Broadband Radio Service (CBRS) Sharing Ecosystem Assessment (SEA) Sensor Data Manual

A guide to using and understanding sensor data produced by the NASCTN CBRS SEA program.

Aric W. Sanders
Daniel G. Kuester
Adam Wunderlich
NIST, Boulder, CO, USA

Anthony W. Romaniello
Todd R. Schumann
Institute for Telecommunication Sciences, National Telecommunications and Information Administration, Boulder, CO, USA

This publication is available free of charge from:
https://doi.org/10.6028/NIST.TN.2359

January 2026

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

Publication History
Approved by the NIST Editorial Review Board on 2025-12-29

How to cite this NIST Technical Series Publication:
Sanders AW, Romaniello AW, Kuester DG, Schumann TR, Wunderlich AJ (2026) Citizen's Broadband Radio Service (CBRS) Sharing Ecosystem Assessment (SEA) Sensor Data Manual. (National Institute of Standards and Technology, Gaithersburg, MD), NIST TN 2359. https://doi.org/10.6028/NIST.TN.2359

Author ORCID iDs
Aric W. Sanders: 0000-0002-2305-543X
Anthony W. Romaniello: 0000-0001-8437-6504
Daniel G. Kuester: 0000-0003-0711-5700
Todd R. Schumann: 0009-0000-6127-9357
Adam Wunderlich: 0000-0002-8463-9156

Contact Information
NASCTN Program Manager: Duncan A. McGillivray
NASCTN Project Manager: Keith A. Hartley
nasctn@nist.gov

National Advanced Spectrum and Communications Test Network (NASCTN)

Mission Statement: *The National Advanced Spectrum and Communications Test Network (NASCTN) provides, through its members, a network for robust test processes and validated measurement data necessary to develop, evaluate and deploy spectrum sharing technologies that can improve access to the spectrum by both federal agencies and non-federal spectrum users.*

NASCTN originated in 2015 as a multi-agency chartered partnership and as of 2022, is a congressionally mandated function of the National Institute of Standards and Technology (NIST). NASCTN is coordinated with the National Telecommunications and Information Administration (NTIA) and other Federal agencies, as appropriate. NIST hosts the NASCTN capability at the Department of Commerce Boulder Laboratories in Boulder, Colorado.

The NASCTN program office facilitates and coordinates the use of intellectual capacity, modeling and simulation, and laboratory and test facilities to meet national spectrum interest and challenges.

NASCTN test projects follow a transparent framework including public dissemination of test plans, adjudicated comments to the test plans, formal reports containing test methods, data, and software to the maximum extent permissible.

The NASCTN provides an impartial, scientifically rigorous forum for addressing spectrum-sharing challenges in an effort to accelerate the deployment of wireless technologies among commercial and federal users and to measure the impacts of spectrum dependent systems deployments. NASCTN provides robust test processes, validated measurements data, and statistical analysis necessary to develop, evaluate, and deploy spectrum sharing technologies that can improve access to the spectrum by both federal agencies and non-federal spectrum users.

NASCTN
National Advanced Spectrum and Communications Test Network

NASCTN CBRS SEA

This work is a result of the NASCTN Citizens Broadband Radio Service (CBRS) Sharing Ecosystem Assessment (SEA) program. The NASCTN CBRS SEA program was requested and sponsored by Defense Information Systems Agency (DISA) to measure emissions in the CBRS band. These emissions would be analyzed to ascertain the effectiveness of the sharing ecosystem between CBRS systems as managed by Spectrum Access Systems (SASs), and Department of Defense (DoD) systems as monitored by Environmental Sensing Capabilities (ESCs). The collected data will be used by DoD to provide insight into the sharing ecosystem's effectiveness, and track changes in the spectrum environment over time. This document explains access to that data. Test members are drawn from government, academia, and/or industry and are the foundation to NASCTN's coordinated, national testing and evaluation capability to enable the improvement and expeditious addressing of national spectrum test interest and challenges.

- Provide expertise and capabilities to cooperatively enable NASCTN studies;
- Work as a partnership to address the interests and equities of all spectrum stakeholders in a fair, equitable, and non-preferential manner; and
- Through sharing of technical resources, with consideration for cost, provide for, liaison, or coordinate access to testing and evaluation capabilities.

Test Members to the CBRS SEA Program are (in alphabetical order):

- Department of Defense (DoD), Chief Information Officer (CIO)
- MITRE Corporation
- National Aeronautics and Space Administration (NASA), Langley Research Center
- National Institute of Standards and Technology (NIST), Communications Technology Laboratory
- National Telecommunications and Information Administration (NTIA), Institute for Telecommunication Sciences (ITS)

Other collaborative partners to CBRS SEA are:

- Hampton University, Norfolk VA.
- Naval Surface Warfare Center, Dahlgren Division
- University of Colorado, Boulder CO.
- United States Coast Guard, C5ISR Program, Rescue 21
- United States Marine Corps, IC4
- United States Navy, Naval Base Pt Loma
- Virginia Department of Transportation

Abstract

The purpose of the NASCTN CBRS SEA project is to provide data-driven insight into the CBRS sharing ecosystem's effectiveness between commercial and DoD incumbent systems, and to track changes in the spectrum environment over time. CBRS is the "first of a kind" nationwide shared spectrum ecosystem in the 3550-3700 MHz band. The implementation of this project included the installation of sensor systems at selected sites on the east and the west coast of the United States, along with a control and prototyping system in Boulder, Colorado. This data manual provides:

1. **Introduction:** A brief description of the NASCTN CBRS SEA program, installed sensor sites and scope.
2. **Data Repository:** A description of the the active data sharing program.
3. **Data Structures and Manipulation Code:** An in-depth description of the data structure and code used to interact with the data.
4. **Sensor Data and Signal Processing:** A discussion of the signal processing of collected in-phase and quadrature data that results in the published data product.

In addition, an appendix describing amplitude probability distribution (APD) has been included. For more information on the program or other test documents visit: https://www.nist.gov/programs-projects/cbrs-sharing-ecosystem-assessment.

Keywords

Advanced Communications;CBRS;CBSD;DPA;ESC;GAA;NASCTN;PALS;RADAR;SAS;Sharing Ecosystem Assessment

Acronyms

ADC	analog-to-digital converter
AGL	above ground level
AMSL	above mean sea level
APD	amplitude probability distribution
CBRS	Citizens Broadband Radio Service
CBSD	Citizens Broadband Radio Service device
CCDF	complementary cumulative distribution function
CSV	comma separated value
DFT	discrete Fourier transform
DISA	Defense Information Systems Agency
DNS	domain name system
DoD	Department of Defense
DPA	Dynamic Protection Area
DTFT	discrete-time Fourier transform
ENBW	equivalent noise bandwidth
ESC	Environmental Sensing Capability
FCC	Federal Communications Commission
FFT	fast Fourier transform
GAA	General Authorized Access
GB-DPA	ground-based Dynamic Protection Area (DPA)
GMM	Green Mountain Mesa
IIR	infinite impulse-response
IQ	in-phase and quadrature
ITS	Institute for Telecommunication Sciences
JSON	JavaScript object notation
LTE	long-term evolution
LZMA	Lempel-Ziv-Markov chain algorithm
MITRE	The MITRE Corporation
NASCTN	National Advanced Spectrum and Communications Test Network
NIT	Norfolk International Terminal
NIST	National Institute of Standards and Technology
NIST BOX	NIST Box Cloud Storage
NTIA	National Telecommunications and Information Administration
PAL	Priority Access License
PFP	periodic frame power
PNG	portable network graphics
PSD	power spectral density
RF	radio frequency
RMS	root mean square
SAS	Spectrum Access System
SCOS	spectrum-characterization and occupancy sensing
SEA	Sharing Ecosystem Assessment
SigMF	signal metadata format
SPU	signal processing unit
TDD	time division duplex
UTC	Coordinated Universal Time

Table of Contents

List of Tables

List of Figures

Technical Contributors

Contributor	Organization	Primary Contribution Areas
Daniel G. Kuester	NIST	Data analysis, software and writing. Developer of nasctn-sea-ingest.
Duncan A. McGillivray	NIST	Data analysis and data product refinement.
Thao T. Nguyen	NIST	Data analysis.
Aric W. Sanders	NIST	Data analysis, software, data pipeline and writing. Developer of nasctn-sea-visualization.
Adam Wunderlich	NIST	Writing of APD appendix.
Douglas Boulware	NTIA ITS	SCOS software, data pipeline.
Anthony W. Romaniello*	NTIA ITS	Data analysis, software and writing. Developer of nasctn-sea-ingest.
Todd Schumann	NTIA ITS	Data analysis, software and writing.
Joseph R. Mruk	MITRE	Data product consultation.
William F. Young	MITRE	Data product consultation.

*Currently at a different institution

Executive Summary

The purpose of the National Advanced Spectrum and Communications Test Network (NASCTN) Citizens Broadband Radio Service (CBRS) Sharing Ecosystem Assessment (SEA) project is to provide data-driven insight into the CBRS sharing ecosystem's effectiveness between commercial and Department of Defense (DoD) incumbent systems, and to track changes in the spectrum environment over time. CBRS is the "first of a kind" nationwide shared spectrum ecosystem in the 3550-3700 MHz band. The implementation of this project included the installation of sensor systems at selected sites on the east and the west coast of the United States, along with a control and prototyping system in Boulder, Colorado. This data manual provides:

1. **Introduction:** A brief description of the NASCTN CBRS SEA program, installed sensor sites and scope.
2. **Data Repository:** A description of the the active data sharing program.
3. **Data Structures and Manipulation Code:** An in-depth description of the data structure and code used to interact with the data.
4. **Sensor Data and Signal Processing:** A discussion of the signal processing of collected in-phase - quadrature data that results in the published data product.

This data manual is not designed to be a comprehensive document about the NASCTN CBRS SEA project. Detailed information about the goals of the project, sensor construction, modeling, and ecosystem analysis can be found at https://www.nist.gov/programs-projects/cbrs-sharing-ecosystem-assessment.

This data manual is targeted at the commercial, military and civilian stakeholders of CBRS that would benefit from in-depth real time and longitudinal analysis of a shared spectrum environment. To access the data produced by this program, email nasctn@nist.gov and request permissions to the repository that is hosted on NIST Box Cloud Storage (NIST BOX). Interoperable data for analysis can be found in comma separated value (CSV) format, along with preprocessed images for inspection. For more advanced analysis in the python programming language, users can access two purpose built repositories hosted on github at https://github.com/usnistgov/nasctn-sea-ingest and https://github.com/usnistgov/nasctn-sea-visualization.

1. Introduction

The Defense Information Systems Agency (DISA) proposed a measurement campaign to the National Advanced Spectrum and Communications Test Network (NASCTN) to measure emissions in the Citizens Broadband Radio Service (CBRS) band (i.e., 3550-3700 MHz). This measurement campaign is designed to ascertain the effectiveness of the sharing ecosystem between CBRS systems (Citizens Broadband Radio Service devices (CBSDs)) as managed by Spectrum Access Systems (SASs), and Department of Defense (DoD) systems as monitored by Environmental Sensing Capabilities (ESCs). The collected data provides DISA insight into the sharing ecosystem's effectiveness and tracks changes in the spectrum environment over time.

This document outlines the data collected in the production period during the execution phase of the test plan, including its location, how to access it, and the methods by which it is derived. The data was collected by sensors deployed on the east coast, west coast and in Boulder, Colorado. The east coast sensors are in proximity to the Dynamic Protection Area (DPA) known as East-1 that covers the maritime region near Norfolk, Virginia, as well as the harbor DPA. The west coast sensors are proximal to the DPAs west-13 and west-14 that cover maritime regions near San Diego and Long Beach, California, as well as the harbor DPA for San Diego. In Figure 1, the sensor's locations are visualized. The sensor in Boulder, Colorado, provides a non-Environmental Sensing Capability (ESC) controlled reference point, monitoring an area not covered by a DPA.

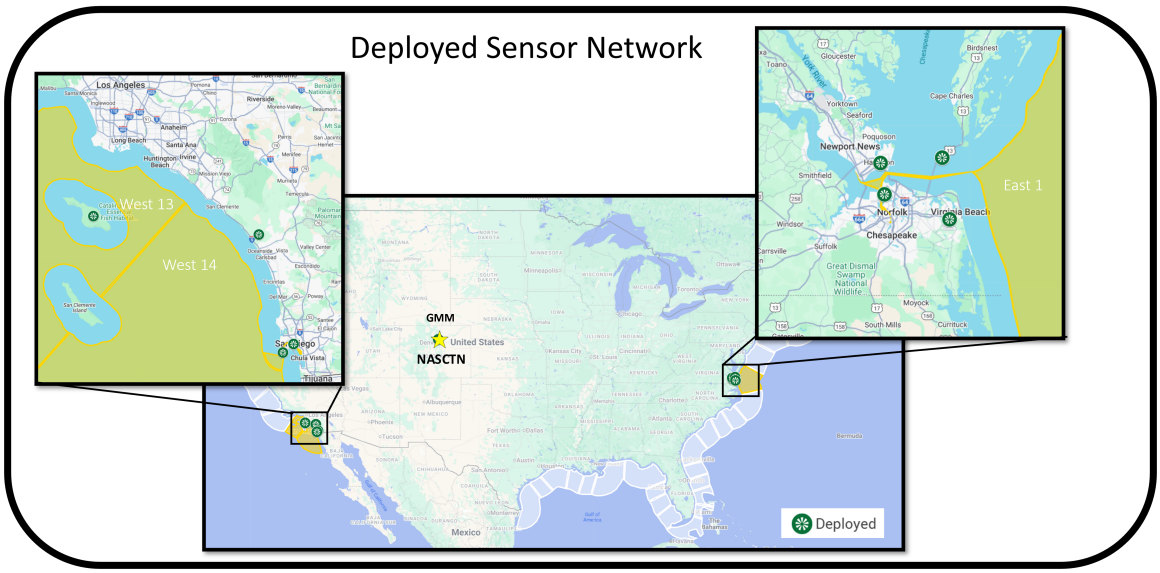

Fig. 1. Map of Deployed Sensors

Each one of the sensors acquires data on a continuous automated basis. This data is intended for use by data scientists, policy makers and other CBRS stakeholders. For example, in Figure 2, a maximum power spectral density for March 30th, 2025 from Hampton University, illustrates both CBSD and probable incumbent activity. Hampton university tower resides on the east coast (more information about its location can be found in Table 1. At 12:00 UTC, the frequency range 3640-3650 MHz is dominated by normal operations of the CBSD, which is illustrated in the lower right of Figure 2. At 13:00 UTC however, a strong incumbent signature and the lack of CBSD activity is observed (upper right of Figure 2). The incumbent signature is characterized by power in a narrow frequency band that switches on every 1 ms in time.

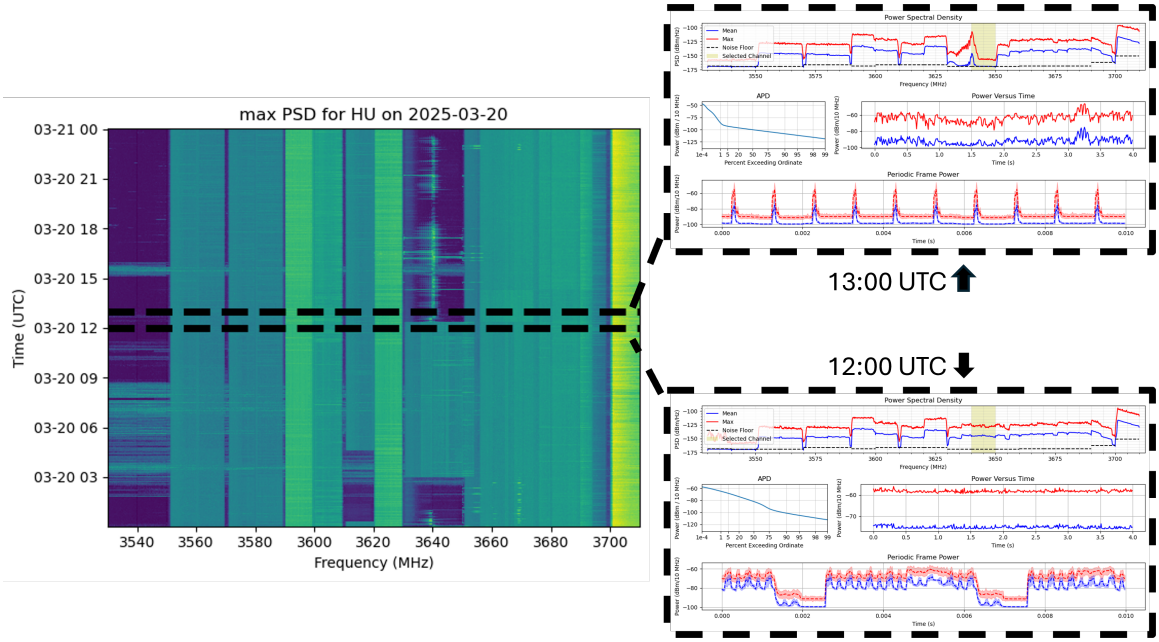

Fig. 2. Example Event Analysis

In addition to the event analysis outlined above, the data also provides insights over long period of times for the deployed sensor regions. In Figure 3, an extended time period from Jan 1st, 2025 to August 12th, 2025 of three sensors' observations, located on the east coast, is shown(the sensors name and location appears in Table 1). In Figure 3, the maximum channel power is displayed at the starting time of each observation of the frequency range of 3530-3710 MHz and the color bar to the left indicates the magnitude of maximum power in dBm for each 10 MHz channel.

In the image, two key features are apparent. First, there are high-power, broad-frequency features that appear in all three sensors. These features are most likely incumbent activity, but further analysis would be required to determine their exact nature. Second, the CBSD activity in the observed frequency range is similar but not identical. This is important

because events that affect all local CBSDs such as DPA activations can be corroborated by multiple sensors, while the differences indicate multiple operating systems are simultaneously monitored. To understand the impact of these observations it is important to briefly describe the CBRS ecosystem and the specific test request made by DISA.

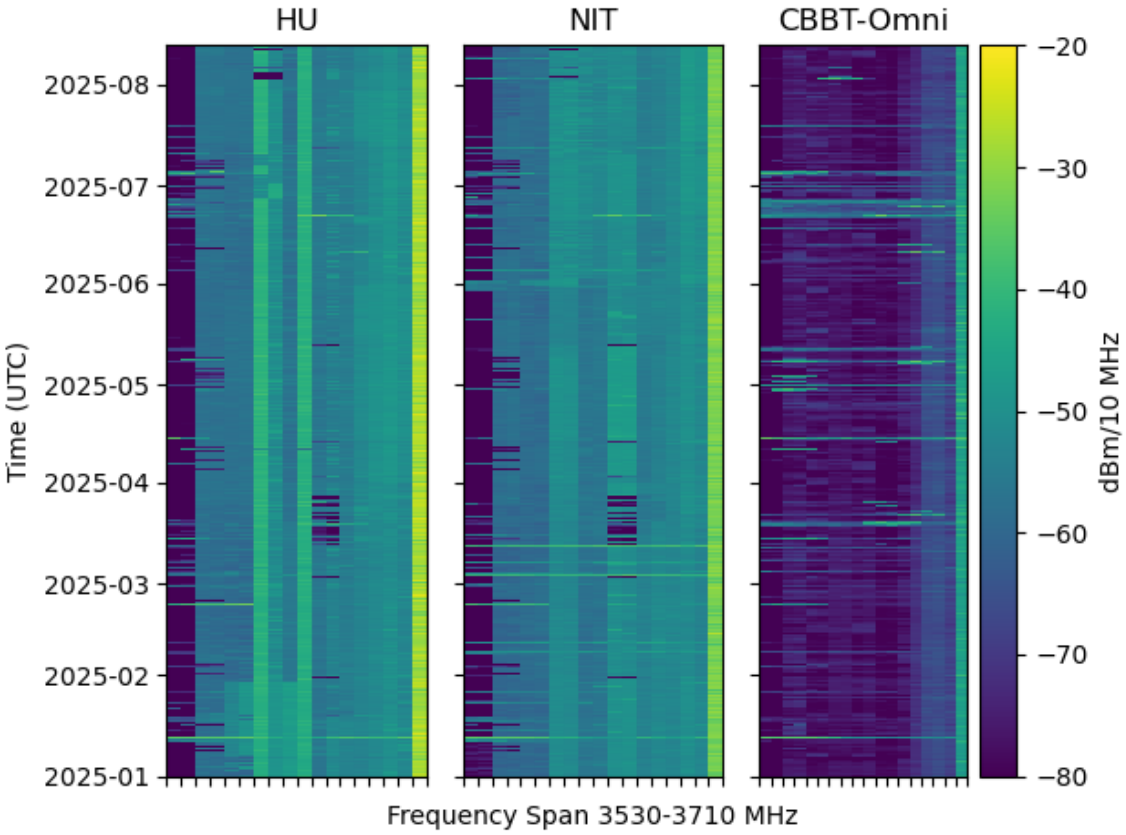

Fig. 3. Example Extended Time Analysis

1.1. CBRS Ecosystem

The Federal Communications Commission (FCC) established the CBRS in the 3550-3700 MHz band, and created a first of its kind 3-tiered access and authorization framework to accommodate Federal incumbent (tier 1) and non-federal users which are tiered as Priority Access License (PAL) (tier 2) and General Authorized Access (GAA) users (tier 3) as shown in Figure 4. The FCC auctioned the band (Auction #105) for commercial use in July 2020. A system has been established to enable this sharing, which is composed of two main parts: The SAS performs automated spectrum use coordination for the PAL and GAA CBSDs, and the ESC detects incumbent use of the band and transmits that information back to the SASs.

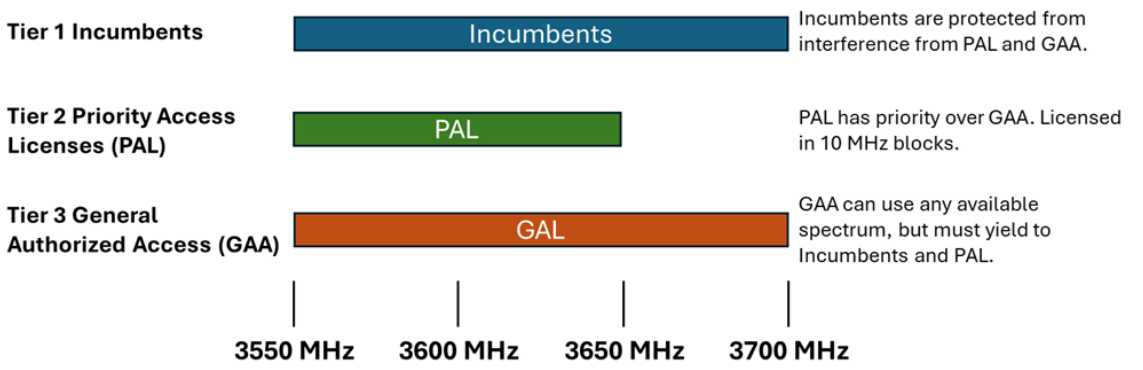

Fig. 4. FCC's three-tiered access for CBRS.

1.2. Test Request

DoD 3.5 GHz Transition Plans focus on ecosystem validations, environmental assessments, and continued engagement on refining the CBRS infrastructure. DISA identified one component for submission to NASCTN: to evaluate the effectiveness of the CBRS sharing ecosystem that manages the co-existence between Federal incumbent systems and commercial users, via an independent trusted agent.

The DISA submitted a CBRS Sharing Ecosystem Assessment (SEA) test request to NASCTN to achieve four objectives regarding the CBRS band:

1. Provide data to determine efficacy of permanent sharing between CBRS systems as managed by SAS systems and ESC systems
    - Attempt to obtain corroborating data from CBRS community stakeholders
2. Collect power levels in the CBRS band through continuous automated observations
3. Collect emissions in the CBRS band in the vicinity of San Diego, CA and Norfolk, VA
4. Collect emissions in the CBRS band within at least one Always-On DPA

This DISA test request was divided into three major components:

- **Passive Observation Coastal DPAs**: Characterize aggregate emissions within the CBRS band 3550-3700 MHz in the vicinity of at least two Coastal DPAs, with and without DPA activations. Assess ecosystem performance to timely respond, and measure increase in background emissions due to wireless system deployments over time.
- **Passive Observation Always On DPA**: Characterize aggregate emissions within the CBRS band 3550-3700 MHz in at least one Always-On DPA or ground-based DPA (GB-DPA). Assess ecosystem performance to limit CBRS emissions in the Always-On DPA.

- **Long Term Support**: Collection and analysis of Passive Observations over 4 years, and support transfer of data to a DoD data repository.

These test request components led directly to a data pipeline for collecting and disseminating observations made by sensors[1, 2]. The text that follows is an explanation of that data pipeline, how to access the data, and how in-phase and quadrature (IQ) is processed into specific types of observations.

1.3. Program Publication

As previously noted, this document is not designed to be a comprehensive document about the NASCTN CBRS SEA project. Several documents are either available in draft form or are in preparation. In Figure 5 the theme, publication status at the time of this publication, and responsible agency are shown. This document supplements the series of previous and pending publications.

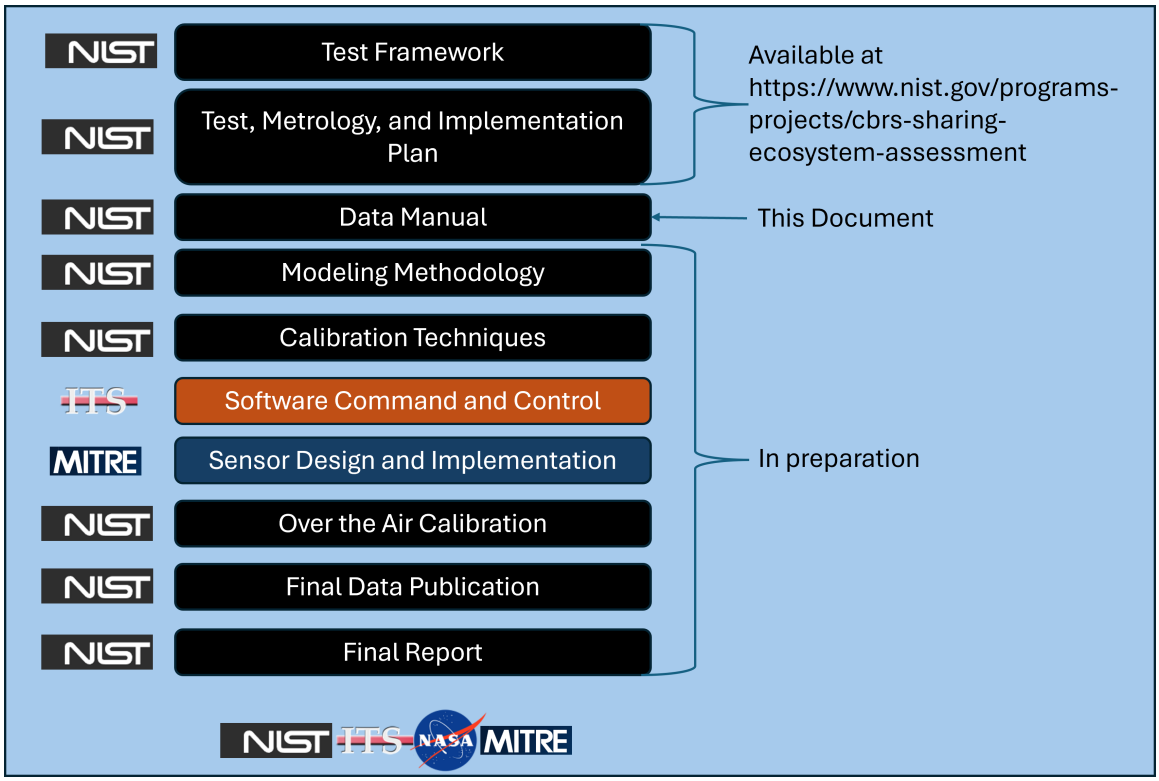

Fig. 5. NASCTN CBRS SEA Publications

2. Data Repository

The NASCTN CBRS SEA project instantiated a data repository to manage and catalog data transmissions from individual radio frequency (RF) sensors, their unique configuration, and the contents of the sensors measurands. The repository is a core element of the data pipeline defined as the mechanism that describes the data instantiation, transport, shaping, and cataloging.

2.1. Data Transport and Storage

Data are retrieved from the sensors using a private network operated by National Telecommunications and Information Administration (NTIA) Institute for Telecommunication Sciences (ITS) on behalf of NASCTN and then securely transported to NIST Box Cloud Storage (NIST BOX). The initial upload to NIST BOX is a golden copy, or definitive source, of the data. This directory remains unmodified except to add new data in the form of zip archives, or to update the archival format of the data. From this directory, content with collections of one or more sensors are derived, see Figure 6.

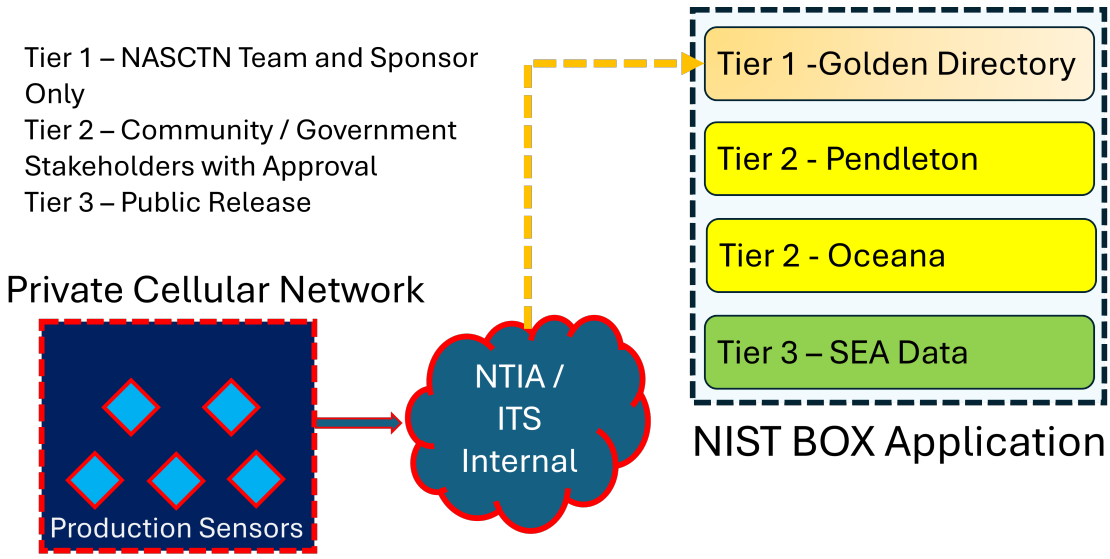

Fig. 6. Sensor to Data Store Transport

This content is then organized into multiple directories (Raw Data, Day Plots, power spectral density (PSD), periodic frame power (PFP) Aligned, Summaries and Extra Information), see Figure 7. Below is a detailed discussion of these directories and their content.

```
SEA-DATA
  Raw Data
  Day Plots
  PFP Aligned
  PSD
  Summaries
  Extra Information
```

Fig. 7. File Hierarchy

2.2. SEA-DATA

The primary publicly accessible data directory is located on NIST BOX, and once a request is made, access will be granted. In addition to the SEA-DATA directory, other directories with data that require a confirmed use case and more stringent identity verification are available to certain stakeholders. The specific content and sensitivity of the data in the repositories may change in the future. In Figure 6, these directories are displayed in yellow with the identifier Tier 2. The directories denoted, correspond with sensors deployed at privileged locations. If other potentially sensitive data is identified, it will also be designated as Tier 2. This data manual focuses on the Tier 3, or publicly released data.

2.2.1. Raw Data

The "Raw Data" directory contains data for each sensor on a per-day basis zipped into individual archives. The date and short name for the sensor is used, for example 2024-09-08_GMM.zip is the data acquired by the Green Mountain Mesa (GMM) sensor in Boulder, CO on the 8th of September 2024. The sensor identification is done by a designated common name which is currently an acronym or short name describing the location of the sensor, see table 1, and the date is always in YYYY-MM-DD format with leading zeros referenced to Coordinated Universal Time (UTC). Each zip file contains the sensors' data captures for the day, in signal metadata format (SigMF) format [3], which began on the date indicated by the file name as of the start time of the first channel capture. These SigMF files contain all data and diagnostics collected from the sensors at any given time spanning that day (see Section 3.1 for a complete description). In addition, for sensors that utilize dual antennas, a further distinction is made between the antenna types: either omnidirectional or directional. Specifics on the sensor configurations and antenna patterns are beyond the scope of this document and are outlined in references [1, 2].

Table 1. The names used in the data repository to identify a particular sensor along with the location of that sensor and an extended name. If a sensor has two antennas the name on the repository with be the sensor short name along with a hyphen and the word omni for an omni-directional antenna, or directional for a 15° beam-width panel antenna (for example, 2025-07-02_Catalina-Directional.zip). The exception to this convention is the prototyping sensor located in Boulder, CO, it is operated with a sector antenna. *Sensors marked with an asterisk, *, are operated in a dual antenna configuration.*

The column sensor short name corresponds to the naming convention in the data repository and the columns above ground level (AGL) and above mean sea level (AMSL) respectively.

Sensor Location and Naming						
Sensor Short Name	Sensor Long Name	Location	Latitude	Longitude	AGL (m)	AMSL (m)
NIT	Norfolk International Terminal	Norfolk, VA	36.91552	-76.32245	45.9	48.5
HU	Hampton University Tower	Hampton, VA	37.02578	-76.34120	68.6	70.4
CBBT*	Island 4, Chesapeake Bay Bridge Tunnel	Norfolk, VA	36.82511	-76.02953	40.4	40.4
Oceana	Naval Air Station Oceana	Virginia Beach, VA	36.825115	-76.029533	8.5	14.6
Catalina*	US Coast Guard Tower Site (Mt Orizaba)	Catalina, CA	33.37956	-118.41641	4.5	604.5
Pendleton	USMC Camp Pendleton	USMC Camp Pendleton, CA	33.28231	-117.38838	6.4	180.4
Midway	USS Midway Museum	San Diego, CA	32.71398	-117.17525	50.0	50.0
PtLoma*	Point Loma	San Diego, CA	32.67026	-117.24047	3.0	129.3
GMM	Green Mountain Mesa	Boulder, CO	39.99180	-105.27450	5.4	1783.4

2.2.2. Day Plots

The directory “Day Plots” contains portable network graphics (PNG) images of multiple data streams visualized as a daily synopsis. The visualization is over the frequency range of 3540-3660 MHz. This reduced frequency range was chosen to emphasize the interplay between incumbents and the CBSDs. An example is shown in Figure 8. In the figure, the left plot is an overload mask where red denotes overload values of true (whenever a single signal value exceeds the maximum voltage of the analog-to-digital converter (ADC) in the 4 s capture). The center plots are time down-sampled versions of the root mean square (RMS)-mean and Peak-Max periodic frame power. For the specific data component see Table 4 and for the calculation see Section 4.4.3. The right most plot is the frequency down-sampled maximum power spectral density. For this specific data component, see Table 4 and for the calculation method, see Section 4.4.2.

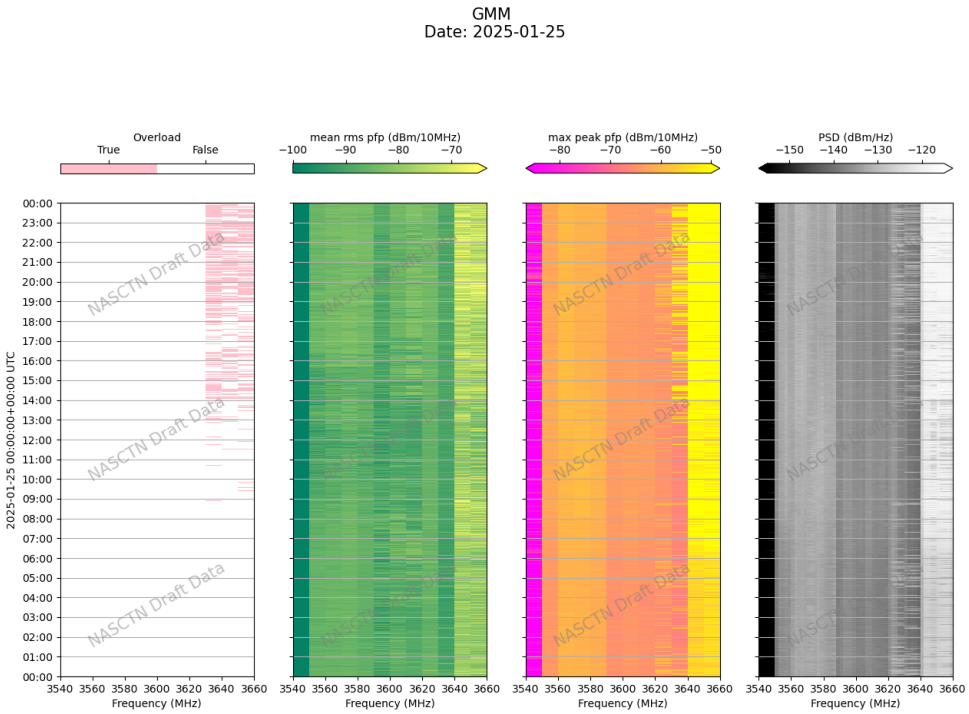

Fig. 8. Typical Plot Found in the Day Plots Directory

2.2.3. PSD

The directory “PSD” holds comma separated value (CSV) files of the PSD traces acquired for each day using different capture statistics (mean, 99th percentile, etc). Each file is in plain text with the first column being the capture timestamp and following columns representing the frequency at which that percentile is calculated. The first row is a header with the

column values being the RF frequency of the PSD. Computer code for Python based loading and plotting of the power spectral density is shown in Figures 9 and 10.

```
import matplotlib.pyplot as plt
import pandas as pd
test_path = r".\PSD\2024-06-14_GMM_max.csv"
psd_df = pd.read_csv(test_path,index_col=0)
psd_df.index = pd.to_datetime(psd_df.index)
colormesh = plt.pcolormesh(1e-6*psd_df.columns.astype(float),psd_df.index,psd_df)
ax= plt.gca()
figure = plt.gcf()
figure.colorbar(colormesh,ax=ax,label="dBm/Hz")
plt.xlabel("Frequency (MHz)")
plt.ylabel("Time (UTC)")
plt.show()
```

Fig. 9. Python Code for Loading the PSDs and Plotting

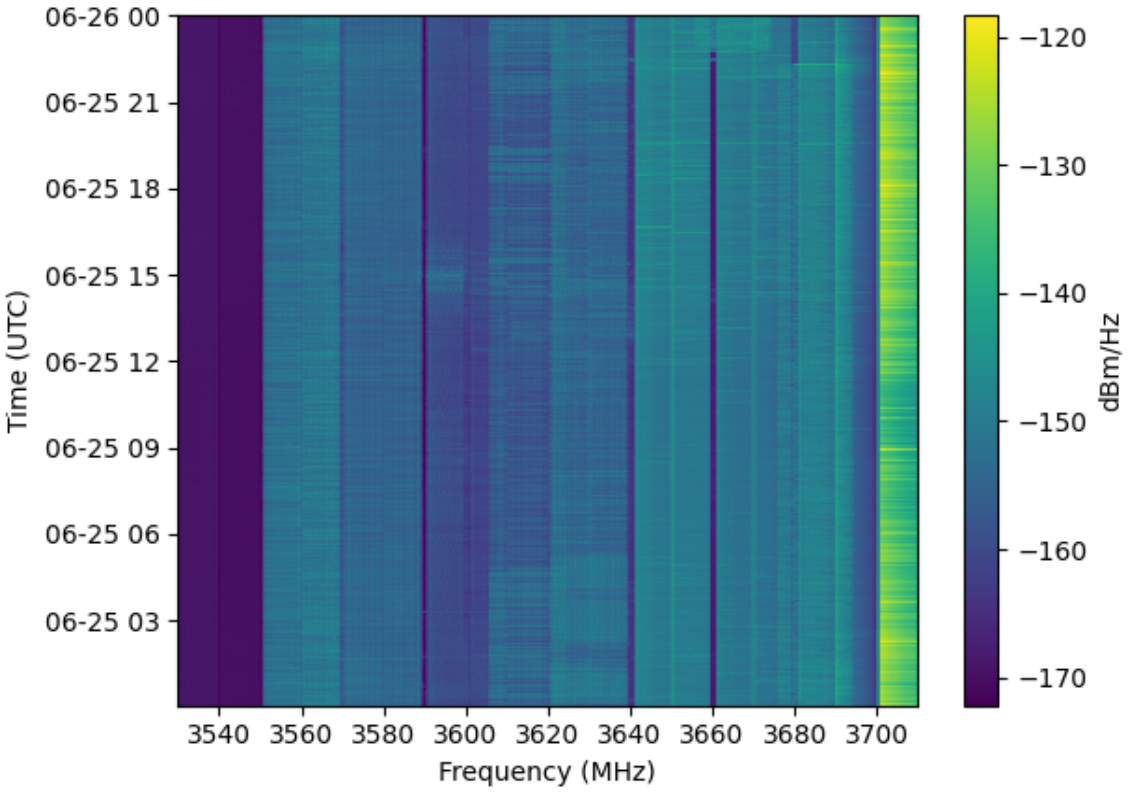

Fig. 10. Example PSD Plot

2.2.4. PFP Aligned

"PFP Aligned" are aligned PFP traces, statistics based on 10 ms power versus time sections for each 10 MHz channel. The alignment procedure calculates the highest correlation between a template waveform and a smoothed version of the PFP, then shifts the time basis to give the appearance of a periodic signal from CBSDs that begins at the same time point. The first two columns are time and frequency of the measurement and the following columns are numbered positions, representing time from 0 to 10 ms. The first row is a header with the index of the trace point within the 10 ms frame. Computer code for Python based loading and plotting of the periodic frame power are shown in Figures 11 and 12.

```python
#imports
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
#data ingest
test_path = r".\PFP Aligned\PFP_2025-07-09_NIT_mean_rms.csv"
frequency = 3605e6
pfp_df= pd.read_csv(test_path,index_col=0)
pfp_df.index = pd.to_datetime(pfp_df.index,format='mixed')
#data selection
x_data = np.linspace(0,10,560)
selected_data = pfp_df[pfp_df["frequency"]==frequency]
selected_data = selected_data.drop("frequency",axis=1)
#plotting
fig, ax = plt.subplots()
colormesh = ax.pcolormesh(x_data,selected_data.index,selected_data,cmap="viridis")
ax.set_ylabel("Time (UTC)")
ax.set_xlabel("Periodic Frame Time (ms) ")
fig.colorbar(colormesh,ax=ax,label="dBm/10 MHz)
plt.show()
```

Fig. 11. Python Code for Loading PFP and Plotting

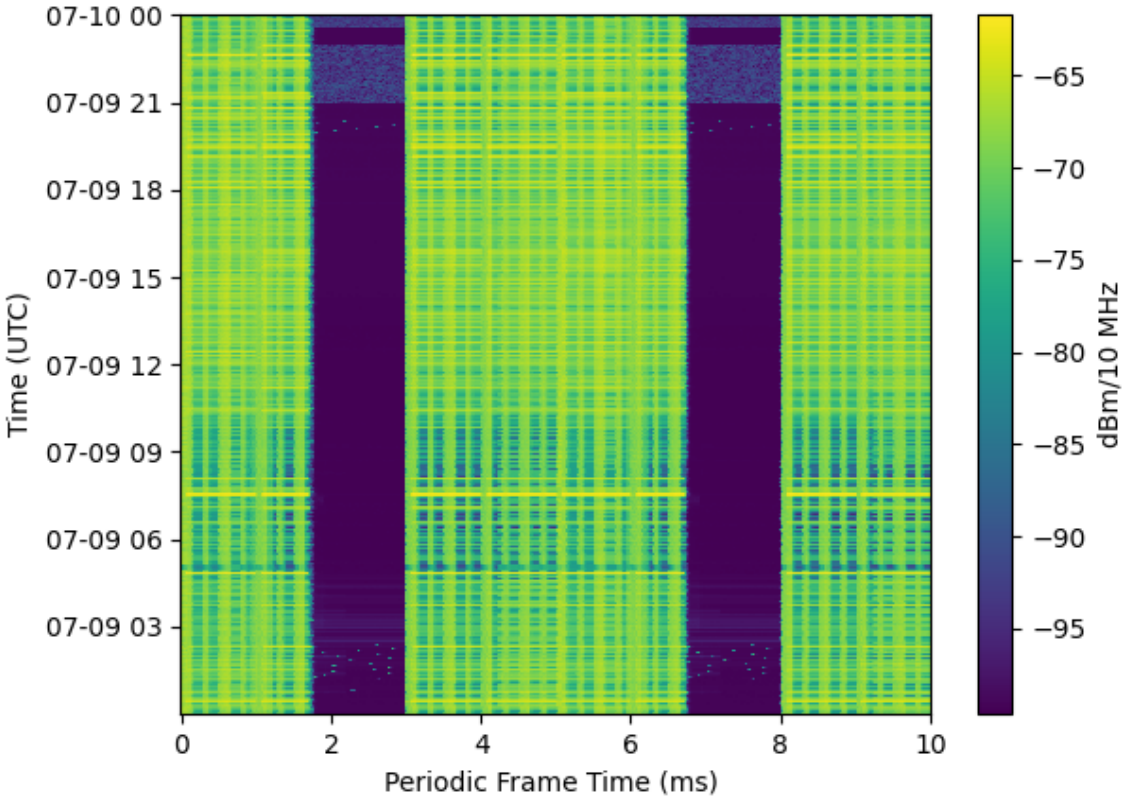

Fig. 12. Example PFP Plot

2.2.5. Summaries

The directory "Summaries" contains CSV tables of overview information or summary statistics (mean, median, and max power) for each channel for all sensors over a month. This provides a convenient way of exploring longitudinal trends of one or more sensors, the fields included are:

- **sensor_hostname**: Sensor host name for domain name system (DNS) look up on NTIA ITS private network.
- **date**: Date of the row in YYYY-MM-DD format.
- **sensor_name**: Short form of sensor name, see Table 1.
- **sensor_id**: Integer for internal use.
- **file**: Raw Data File used to extract data.
- **channel**: Integer corresponding to order in which the 10 MHz channel was acquired.

- **channel_frequency_mhz**: Center frequency in MHz of the 10 MHz channel acquisition.
- **timestamp**: UTC Timestamp of 10 MHz acquisition in YYYY-MM-DD HH:MM:SS.uuuuuu format.
- **acquisition_timestamp**: UTC Timestamp of the beginning of all channel acquisitions (the same for all channels in a single .sigmf file) in YYYY-MM-DD HH:MM:SS.uuuuuu format.
- **max**: The absolute maximum in units of dBm of the 4 s capture for the 10 MHz capture after the infinite impulse-response (IIR) filter.
- **median**: The linear power median in units of dBm of the 4 s capture for the 10 MHz capture after the IIR filter.
- **mean**: The linear power mean in units of dBm of the 4 s capture for the 10 MHz capture after the IIR filter.
- **overload**: The overload flag from the instrument for the 4 s capture for the 10 MHz channel. This value will be True if any single point caused an ADC overload in 4 s IQ capture.

Computer code for Python based loading and plotting of the maximum power in the summary is shown in Figure 13 and Figure 14.

```python
import os
import pandas as pd
import matplotlib.pyplot as plt
repository = r"./Summaries"
file_name = "2025_07.csv"
sensor = "HU"
stream = "max" #could be median or mean
summary = pd.read_csv(os.path.join(repository,file_name))
summary["acquisition_timestamp"] = pd.to_datetime(summary["acquisition_timestamp"],
                                                  format="mixed")
selected_data=summary[summary["sensor_name"]==sensor]
pivot = selected_data.pivot(index="acquisition_timestamp",
                            columns="channel_frequency_mhz", values=stream)
colormesh=plt.pcolormesh(pivot.columns,pivot.index,pivot.values,
                         cmap="viridis",rasterized=True)
plt.ylabel("Time (UTC)")
plt.xlabel("Frequency (MHz)")
figure = plt.gcf()
figure.colorbar(colormesh,label="dBm/10 MHz")
plt.title(f"{sensor} for {file_name.replace('.csv','')}")
plt.tight_layout()
plt.show()
```

Fig. 13. Python Code for Loading Summary and Plotting Maximum Channel Power

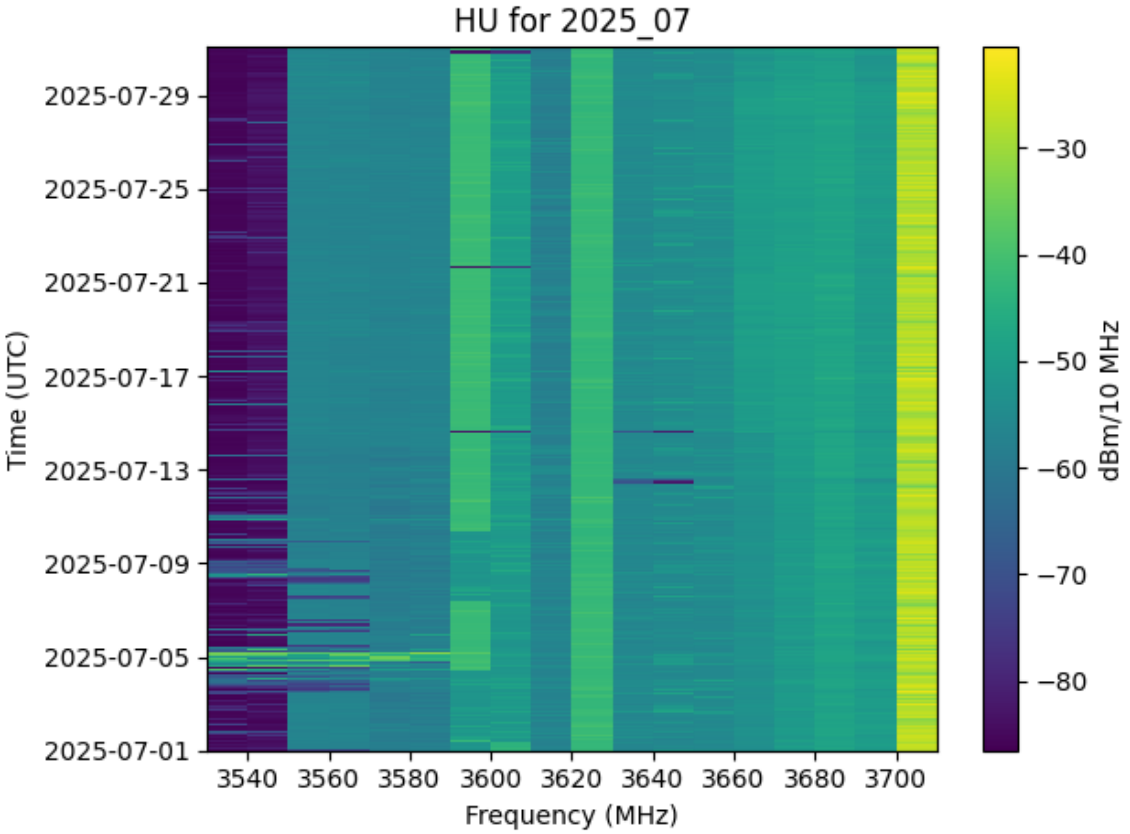

Fig. 14. Example Summary Plot

2.2.6. Extra Information

The extra information directory is a collection of ancillary information. This ancillary information is currently a presentation in power point format that describes the sharing process.

3. Data Structures and Manipulation Code

Every sensing action of the sensor produces a serialized data product that consists of a metadata file and a numeric array of half precision floating point numbers. These files are saved as JavaScript object notation (JSON) and compressed binary respectively. To prevent confusion, both the metadata and numeric list are placed into a single tar repository. This tar repository is then designated with the .sigmf extension. The SigMF format is a standard data format for saving radio frequency datasets. This project has utilized this standard data format with a set of extensions to describe the data product produced by each sensing action. In order to facilitate transport and storage, these SigMF files have been further aggregated into zip archives where each archive includes all the daily sensing actions for a single sensor location and antenna port on a UTC time basis. If an adjective describing the antenna port is **not** present in the file name, then the antenna is an omni-directional antenna[1]. To simplify interactions with this data, several python based packages have been prepared. These packages allow users with a properly configured python environment, see Table 2, to import the data from the zip archives or single sigmf files and transform them to structured python pandas dataframes [4].

Table 2. Table of Python Requirements

Component	Version
Python	3.9 - 3.10
pyarrow	—
methodtools	>=0.4
frozendict	>=2.3
msgpack	>=1.0
eliot	>=1.14.0
distributed	>=2023.2.1
dask[complete]	>=2023.3.0”
lz4	>=4.3.2
matplotlib	>=>=3.6
timezonefinder	>=6.0.2
numpy	>=1.19,<2
msgspec	>=0.15
line-profiler	>=4.03

[1]With the exception of the Green Mountain Mesa (GMM) sensor which utilized a 120° sector antenna as the only capture antenna.

3.1. Sensor Dayblocks

The primary data sources are the zip archives of sensor sensing actions. These are known as dayblocks and all other data included in the repository structure are processed elements of these. As mentioned in section 2.2.1, these are collections of sensing actions for a particular sensor-antenna combination over a period of a day. In Figure 15, the structure of the dayblock is visualized. The directory contains many zip archives of sensor-antenna-day unique combinations, with most sensors only having a single antenna. These zip archives are collections of tar archives (SigMF), see section 3.2, of metadata files in the JSON format and a numeric array of half precision floats compressed using the Lempel-Ziv-Markov chain algorithm (LZMA).

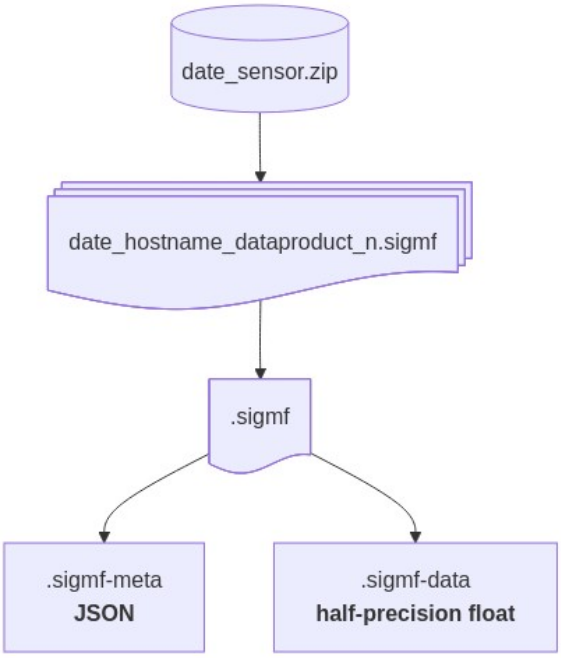

Fig. 15. Zip Archive of Sensing Actions

3.2. Realized Signal Metadata Format

SigMF is a data format designed for the interoperability and longevity of recorded signal data. During the course of the NASCTN CBRS SEA project, SigMF was chosen as the storage mechanism of the results of sensing actions. As pilot studies and sensor deployments evolved, the information resulting from these sensing actions also evolved. For example, the earliest versions of the serialized data product included almost all of the metadata for describing the numeric array in the subfield Annotations, while the current version stores none of it there. These previous data structures are documented through the use of code

and metadata specifications in the package [5], the metadata extensions used are included in Table 3.

3.3. Metadata Structure

Metadata are recorded alongside each edge data sweep in a JSON-formatted text file based on SigMF [3]. The metadata provide the context needed to parse and interpret the raw data, including summary statistics and diagnostic sensor information. The top-level metadata follows the SigMF metadata specification (see core namespace in table 3), i.e. the top most level is divided into three keys, global, captures and annotations. In the realized metadata structure, the element "annotations" is empty, the element global follows the structure in figure 16, and the captures field follows the structure in figure 17. Additionally, any string identifier that precedes a key is a namespace that can be found in table 3.

Table 3. JSON Metadata key Namespaces. Each composite key has a prefix which identifies its namespace. This namespace identifies the meaning and type of the field.

Namespace	Link
core	https://sigmf.org/index.html#sigmf-metadata-format
ntia-core	https://github.com/NTIA/sigmf-ns-ntia/blob/master/ntia-core.sigmf-ext.md
ntia-algorithm	https://github.com/NTIA/sigmf-ns-ntia/blob/master/ntia-algorithm.sigmf-ext.md
ntia-diagnostics	https://github.com/NTIA/sigmf-ns-ntia/blob/master/ntia-diagnostics.sigmf-ext.md
ntia-nasctn-sea	https://github.com/NTIA/sigmf-ns-ntia/blob/master/ntia-nasctn-sea.sigmf-ext.md
ntia-scos	https://github.com/NTIA/sigmf-ns-ntia/blob/master/ntia-scos.sigmf-ext.md
ntia-sensor	https://github.com/NTIA/sigmf-ns-ntia/blob/master/ntia-sensor.sigmf-ext.md

```
global
  core:datatype
  core:extensions
  core:geolocation
      coordinates
      type
  core:num_channels
  core:recorder
  core:sample_rate
  core:sha512
  core:version
  ntia-algorithm:data_products
  ntia-algorithm:processing
  ntia-algorithm:processing_info
  ntia-core:classification
  ntia-diagnostics:diagnostics
      computer
          action_cpu_usage
          action_runtime
          cpu_max_clock
          cpu_mean_clock
          cpu_min_clock
          cpu_overheating
          cpu_temp
          cpu_uptime
          disk_usage
          memory_usage
          software_start
          software_uptime
          ssd_smart_data
              available_spare
              available_spare_threshold
              critical_warning
              integrity_errors
              percentage_used
              temp
              test_passed
              unsafe_shutdowns
          system_load_5m
      datetime
      preselector
```

```
          antenna_path_enabled
          door_closed
          humidity
          lna_powered
          lna_temp
          noise_diode_path_enabled
          noise_diode_powered
          noise_diode_temp
          temp
     software
          preselector_api_version
          python_version
          scos_actions_version
          scos_sensor_version
          scos_sigan_plugin
               name
               version
          sigan_api_version
          sigan_firmware_version
          system_platform
     spu
          battery_backup
          door_closed
          humidity_sensors
          low_battery
          power_sensors
          preselector_powered
          replace_battery
          sigan_powered
          temperature_control_powered
          temperature_sensors
          ups_healthy
ntia-nasctn-sea:max_of_max_channel_powers
ntia-nasctn-sea:mean_channel_powers
ntia-nasctn-sea:median_channel_powers
ntia-nasctn-sea:median_of_mean_channel_powers
ntia-scos:action
     name
     summary
ntia-scos:schedule
     id
     interval
```

```
     name
     priority
     start
ntia-scos:task
ntia-sensor:sensor
     sensor_sha512
     sensor_spec
          id
```

Fig. 16. Structure of the JSON Global Metadata

```
captures
   capture 0
        core:frequency
        core:datetime
        ntia-sensor:duration
        ntia-sensor:overload
        ntia-sensor:sensor_calibration
             datetime
             gain
             noise_figure
             reference
             temperature
        ntia-sensor:sigan_settings
             reference_level
             attenuation
             preamp_enable
        core:sample_start
   capture 1
   capture 2
   capture 3
   capture 4
   capture 5
   capture 6
   capture 7
   capture 8
   capture 9
   capture 10
   capture 11
   capture 12
   capture 13
   capture 14
   capture 15
   capture 16
   capture 17
```

Fig. 17. Structure of the JSON Captures Metadata

```
import json
test_file = r"2025-07-09_GMM/2025-7-9-0-0-57563/2025-7-9-0-0-57563.sigmf-meta"
with open(test_file,"r") as infile:
    metadata = json.load(infile)
```

Fig. 18. Python Code for Loading Metadata into Memory

3.4. Numeric Array Structure

The numeric array is an LZMA compressed binary file of half precision floating point numbers. It contains all of the edge data products in a prescribed order, see table 4 for particulars. To load the numeric data, it must first be extracted from the raw zip archive, then extracted from the .sigmf tar archive, then decompressed and then read. A python based process for doing this after extraction is shown in figure 19.

```
import numpy as np
import lzma
test_file = r"2025-07-09_GMM/2025-7-9-0-0-57563/2025-7-9-0-0-57563.sigmf-data"
lzma_data = open(test_file,"rb").read()
byte_data = lzma.decompress(lzma_data,format=lzma.FORMAT_XZ)
numeric_data = np.frombuffer(byte_data, dtype="half")
```

Fig. 19. Python Code for Loading Numeric Array into Memory

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the nth channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
Maximum Power Spectral Density	dBm/Hz	0	124	125	Results of statistical detector max applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.
Mean Power Spectral Density	dBm/Hz	125	249	125	Results of statistical detector mean applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the n^{th} channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
Median Power Spectral Density	dBm/Hz	250	374	125	Results of statistical detector median applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.
25th Percentile Power Spectral Density	dBm/Hz	375	499	125	Results of statistical detector 25th percentile applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.
75th Percentile Power Spectral Density	dBm/Hz	500	624	125	Results of statistical detector 75th percentile applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.
90th Percentile Power Spectral Density	dBm/Hz	625	749	125	Results of statistical detector 90th percentile applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the n^{th} channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
95th Percentile Power Spectral Density	dBm/Hz	750	874	125	Results of statistical detector 95th percentile applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.
99.9th Percentile Power Spectral Density	dBm/Hz	1000	1124	125	Results of statistical detector 99.9th percentile applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.
99.99th Percentile Power Spectral Density	dBm/Hz	1125	1249	125	Results of statistical detector 99.99th percentile applied to power spectral density samples, trimmed to 10 MHz analytic bandwidth. FFTs computed on IIR-filtered data.

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the n^{th} channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
Maximum Power vs. Time	dBm	1250	1649	400	Maximum channel power vs. time, with an integration time of 10 ms. Each data point represents the result of a statistical detector applied over the previous 10 ms. In total, 56000000 IQ samples were used as the input.
Mean Power vs. Time	dBm	1650	2049	400	Mean channel power vs. time, with an integration time of 10 ms. Each data point represents the result of a statistical detector applied over the previous 10 ms. In total, 56000000 IQ samples were used as the input.
Peak-Minimum Periodic Frame Power	dBm	2050	2609	560	Channelized periodic frame power statistics reported over a 10 ms frame period, with frame resolution of 17.86 μs. The max detector is first applied over the frame resolution, then minimum statistics are computed on samples sharing the same index within the frame period.

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the n^{th} channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
Peak-Maximum Periodic Frame Power	dBm	2610	3169	560	Channelized periodic frame power statistics reported over a 10 ms frame period, with frame resolution of 17.86 μs. The max detector is first applied over the frame resolution, then maximum statistics are computed on samples sharing the same index within the frame period.
Peak-Mean Periodic Frame Power	dBm	3170	3729	560	Channelized periodic frame power statistics reported over a 10 ms frame period, with frame resolution of 17.86 μs. The max detector is first applied over the frame resolution, then mean statistics are computed on samples sharing the same index within the frame period.

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the n^{th} channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
RMS-Minimum Periodic Frame Power	dBm	3730	4289	560	Channelized periodic frame power statistics reported over a 10 ms frame period, with frame resolution of 17.86 μs. The mean (RMS) detector is first applied over the frame resolution, then minimum statistics are computed on samples sharing the same index within the frame period.
RMS-Maximum Periodic Frame Power	dBm	4290	4849	560	Channelized periodic frame power statistics reported over a 10 ms frame period, with frame resolution of 17.86 μs. The mean (RMS) detector is first applied over the frame resolution, then maximum statistics are computed on samples sharing the same index within the frame period.

Table 4. The numeric array structure. The numeric array is saved as half precision floats. Each data product is stored as a set of numbers from start index to stop index, and each array structure is repeated for each 10 MHz channel, with a total of 5560 points per channel. This gives a total number of half precision floats as 18 channels * 5560 = 100080. To retrieve the n^{th} channel use the index $start\_index + n * 5560$ where all indices are zero referenced. The channels start at a center frequency of 3535 MHz and end at 3705 MHz. In the tar archive with the extension .sigmf, the numeric array is saved as a separate file designated as .sigmf-data.

Data Product	Units	Start Index	Stop Index	Size	Description
RMS-mean Periodic Frame Power	dBm	4850	5409	560	Channelized periodic frame power statistics reported over a 10 ms frame period, with frame resolution of 17.86 μs. The mean (RMS) detector is first applied over the frame resolution, then mean statistics are computed on samples sharing the same index within the frame period.
Amplitude Probability Distribution	Percentage	5410	5560	151	Estimate of the APD, using a 1.0 dB bin size for amplitude values. The data payload includes probability values, as percentages, indicating the probability that a given IQ sample exceeds the corresponding amplitudes.

3.5. Manipulation Code

3.5.1. nasctn-sea-ingest

nasctn-sea-ingest is a python package concerned with the transformation of SigMF and archives of many SigMFs to pandas [4] tables. These tables are then manipulated to create derived data or plots of interesting features. The pandas tables are arranged in a list of dictionaries indexed by their common name. A simple sample of python code for importing

a single data product into memory is shown in figure 20. More information and examples can be found at https://github.com/usnistgov/nasctn-sea-ingest.

```python
from sea_ingest import read_seamf

# Path to your .sigmf file
file_path = "path/to/your/file.sigmf"

# Read the data
data = read_seamf(file_path)

# Access, for example, the PSD data. In a Jupyter notebook, skip the "print()"!
print(data["psd"])
```

Fig. 20. Python Code for Loading Single File into Memory

3.5.2. nasctn-sea-visualization

nasctn-sea-visualization is a python package concerned with reshaping data, providing access to common plots, and building content in the main data sharing repositories. For an example of importing data and plotting it see Figure 21. More information and examples can be found at https://github.com/usnistgov/nasctn-sea-visualization.

```python
import sys
import os
import numpy as np
import matplotlib.pyplot as plt
from data_models import *
from plotters import *
hu_path = r".\Raw Data\2024-08-10_HU.zip"
hu_day_block = DayBlock(file_path = hu_path)
hu_plotter = DayBlockPlotter(hu_day_block)
# This plots mean psd for the full day
hu_plotter.plot_day_psd(capture_statistic="mean")
#this plots the day plot
hu_plotter.plot_day()
# This plots the aligned PFP for the 11 channel,
# you can use frequency as the selector also
hu_plotter.plot_aligned_pfp(channel=11)
# This plots a DL /UL separation attempt
hu_plotter.plot_day_link_separation(channel=11)
# This selects a single sigmf, slice can be in a pd.datetime object too
hu_slice = DataProduct(data=hu_day_block.get_single(100))
hu_slice_plotter = DataProductPlotter(hu_slice)
# plots a channel pfp and histogram
hu_slice_plotter.plot_channel_pfp_hist(11)
# plots a channel summary
hu_slice_plotter.plot_channel_summary(hu_day_block.frequencies[10])
#plots all psd's for a single slice
hu_slice_plotter.plot_all_psds()
#plots a single channel pfp
hu_slice_plotter.plot_channel_pfp(hu_day_block.frequencies[0])
```

Fig. 21. Python Code for Plotting Day Values of Interest

Table 5. Table of Public SCOS components. In addition to the public components scos-manager is used to coordinate sensor network.

Component	Link
scos-sensor	https://github.com/NTIA/scos-sensor
scos-actions	https://github.com/NTIA/scos-actions
scos-tekrsa	https://github.com/NTIA/scos-tekrsa
preselector	https://github.com/NTIA/Preselector

4. Sensor Data and Signal Processing

This chapter addresses the real-time acquisition and analysis of radio channel measurements performed by each sensor in the frequency range 3530-3710 MHz.

4.1. Edge Data Payloads

Edge data payloads are the measurements produced by each SEA sensor. These data comprise a stream of power statistics and their distributions across time and frequency, together with associated metadata. A subset of the sensor data are used in post-analysis to produce derived measurements that address the sponsor requirements stated in section 1.2. Edge data payloads are thus intermediate between IQ waveforms and other aggregate analysis. Each sensor operates as a field instrument tailored to assess the CBRS sharing environment. This operation required the development of automation software that is responsible for operating the signal analyzer, digital signal filtering, statistical analysis, and integration with the spectrum-characterization and occupancy sensing (SCOS) software framework [6] (see table 5 for public components). The sensors acquire waveforms from CBRS channels, apply calibration corrections to the data, and record the end results. The measurements are packaged with metadata by the SCOS software for upload to centralized data storage hosted by NASCTN as discussed in section 2.

4.2. Sensing Architecture

The sensor performs a stepped frequency sweep across the band for each CBRS channel. In each channel, the sensor computes statistics on RF power based on IQ waveforms streamed from its signal analyzer. Within practical data and computation constraints, the goal is to generate a rich data payload which allows for flexible and in-depth analysis in aggregate.

Overall design of the edge-computed signal processing algorithm is driven by project goals subsection 1.2 and operating parameters of CBRS but constrained by practical implementation limitations. For example, a goal of the project is to understand aggregate emissions of CBSDs in the presence and absence of a protected incumbent. There is also a maximum data throughput rate supported by the long-term evolution (LTE) backhaul, which limits the

amount of data which can be sent from sensors at the edge back to the central NASCTN data repository. Finally, there are inherent constraints of the signal processing unit (SPU) computer, such as processing and memory capacity.

Figure 22 shows a high-level overview of the edge compute algorithm's processing flow and parallelization. The sensors run two primary processes in parallel: acquisition of IQ waveform captures (the blue block), and IQ sample post-processing (the green block). After the sensor records IQ samples in one channel, the post-processing of those samples begins concurrently with the next channel IQ recording. Finally, after all IQ recording and processing has been completed, capture metadata and sensor diagnostics are recorded and the results are saved (the red block). On average, the combination of these processes takes approximately 5 s in total to complete for a single 10 MHz channel using a channel dwell time of 4 s, with a measurement of the full frequency span of 3530-3710 MHZ being performed in approximately 90 s. The total recording and processing is ultimately a function of the IQ recording dwell time in each channel. The dwell time itself sets a lower bound on total process time. Longer channel dwell times also result in more samples being processed into power statistics, and the computation time of these processes increases with the number of input samples.

Figure 23 shows the IQ post-processing which occurs for each channel after recording has completed. The IQ post-processing for each channel is implemented using a combination of serial and parallel tasks. First, the IIR filter is applied to the IQ waveform. Next, parallel processes compute each of power spectral density (PSD), time series power, amplitude probability distribution (APD), and periodic frame power (PFP) results. Each of these steps is described in detail below.

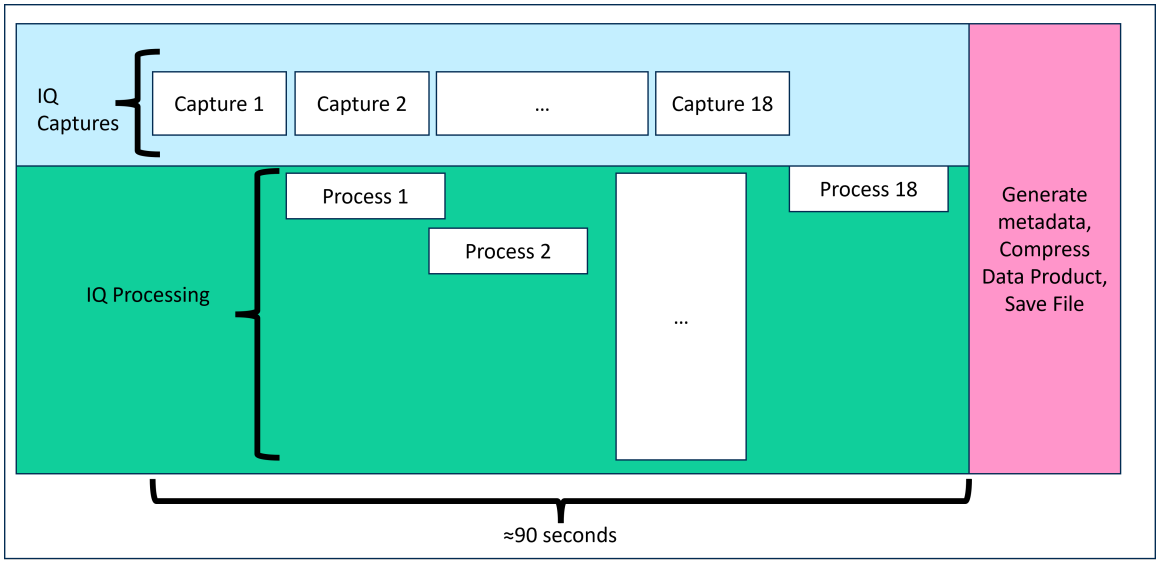

Fig. 22. A block diagram indicating the high-level processing flow of IQ captures and data processing for a full-band edge computed data capture.

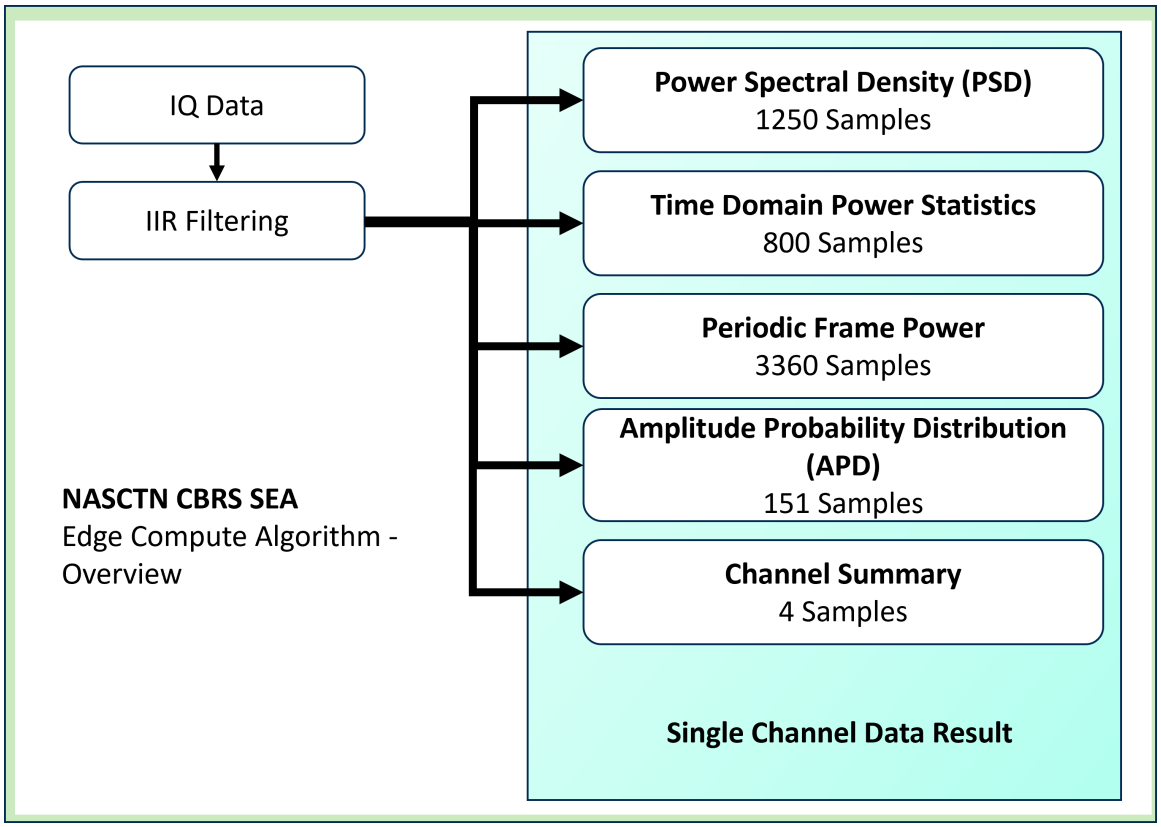

Fig. 23. A block diagram indicating the digital signal processing flow for a single channel IQ capture within the edge compute algorithm. The IQ samples are filtered and then processed in parallel to generate the various components of the edge data payload.

4.3. Real-Time Waveform Streaming

Working within these constraints, a signal processing algorithm is computed at the edge on the SPU. This algorithm produces the following power statistics for the IQ waveform acquired in each channel and sweep: time series, power spectral density (PSD), periodic frame power (PFP), and amplitude probability distribution (APD). IIR filtering is used to constrain all of these power statistics to each channel. Each component of the sensor data stream, and the processing flow of the algorithm overall, are described below.

4.3.1. Acquisition

First, the spectrum analyzer is used to capture IQ samples. The analyzer is set to use a 14 MHz sampling rate, which results in 10 MHz of valid instantaneous measurement bandwidth due to the anti-aliasing filters built in to the spectrum analyzer itself. The center frequency is set to 3535 MHz, and a 4 s IQ capture is performed which instantaneously measures the entire 10 MHz channel from 3530 to 3540 MHz. The attenuation setting

Table 6. An overview of edge compute parameters used for prototype IQ capture and data processing.

Category	Parameter	Value
IQ Capture and Spectrum Analyzer	Record time	4 s
	Center Frequency	3535, 3545, ..., 3705 MHz
	Attenuation	Variable
	Preamp Enable	True
	Reference Level	-25 dBm
	Sampling Rate	14 MHz
IIR Filter	Max. Passband Ripple	0.1 dB
	Min. Stopband Attenuation	40.0 dB
	Passband Edge Frequency	5 MHz
	Stopband Edge Frequency	5.008 MHz
Time Domain Power Series	Detector Window	10 ms
PSD	discrete Fourier transform (DFT) Length	175
	Trimmed DFT Length Number of DFTs for Detection	125 320,000
	Window Type	Flat top-Energy Corrected
	Frequency Resolution Δf	80 kHz
PFP	Reference Frame Period	10 ms
	Resolution	$1/56$ ms (≈ 17.9 µs)
APD	Power Bin Size	1 dBm

on the signal analyzer is variable, dependent on the sensor location and the strength of signals observed, with the goal being to use as little attenuation as possible while avoiding spectrum analyzer overload on any channel. A single channel IQ capture results in a discrete-time sequence of complex-valued samples of a voltage waveform:

$$\text{IQ}[n] = I[n] + jQ[n]. \tag{1}$$

Here, $I[n]$ is the in-phase component and $Q[n]$ is the quadrature component of the quadrature baseband signal. The sample index n ranges from 0 to 55,999,999 for a 4 s IQ capture using a 14 MHz sampling rate.

4.3.2. Calibration Corrections

Before any other stage of processing, the channelized IQ data is scaled using a gain factor obtained from calibrations, effectively changing the reference point of the data from the input of the spectrum analyzer to the input port of the antenna. Y-Factor calibrations are

automatically performed periodically in the field after sensor deployment. Using the calibrated noise source built into the preselector, this routine yields a calibrated power gain value for each channel. This process is covered in detail in [1, 2] and longitudinal trends are expected to be covered in future publications. The actual signal gain is frequency-dependent, but the calibration routine generates an average calibrated gain value representing the entire 10 MHz channel. If the calibrated channel power gain, in linear units, is G_{eff}, then the scaling is applied to recorded IQ samples.

$$\text{IQ}[n] = \frac{I[n]}{\sqrt{G_{\text{eff}}}} + j\frac{Q[n]}{\sqrt{G_{\text{eff}}}} \tag{2}$$

4.3.3. Digital Filtering

Background: Investigations of the spectrum analyzer response revealed that the integrated hardware anti-aliasing filters provide slow roll-off in frequency response outside of the measured channel. This presents a need for additional digital filtering of IQ samples, so that power statistics and other data results can be accurately and independently reported for each 10 MHz channel. Digital filtering drastically improves the out-of-band rejection performance for each channel power measurement. Because the filtering is applied after the samples are digitized by the spectrum analyzer it does not provide protection from ADC overrange events occurring due to high power out-of-channel signals. These effects are mitigated by the combination of a physical filter and a programmable attenuator inside of the spectrum analyzer.

Description: An IIR filter was designed to optimize for filter stability, high passband flatness, high stopband rejection, and small transition band from passband to stopband. Because the observations of interest are power based, linear phase response is not a primary consideration. Therefore IIR filtering was selected over finite impulse response filtering. This selection results in faster and more efficient digital filtering of large streams of IQ samples at the edge. The selected IIR filter is an elliptic (Cauer) filter of order 12. The elliptic filter selection is motivated by its minimization of the transition interval between the passband and stopband for a given amount of ripple. The implemented filter has a maximum passband ripple of 0.1 dB below unity gain, minimum stopband attenuation of 40.0 dB, and a transition width of 8 kHz between the passband and stopband. The implemented filter has a minimal passband ripple that approximates the response of a Type II, or inverse, Chebyshev filter. The implemented low-pass filter behaves like a band-pass filter for a complex input sequence such as the stream of input IQ samples. In this case, the low-pass filter's 5 MHz cutoff frequency functions as a 10 MHz bandpass filter on the complex baseband signal. The filter's poles lie within the unit circle in the Z-plane, indicating stability. To avoid numerical errors and potential reduced filter stability due to finite floating point precision, the filter is implemented using second-order sections. The filter's transfer function is defined to be:

Table 7. The coefficients of the IIR filter used in IQ data processing

i	a_i	b_i
0	0.22001756	1
1	1.89508588	5.98460684
2	8.08369813	19.1994547
3	22.28438409	40.7912472
4	43.9358511	63.2429677
5	65.02462875	74.3311099
6	73.93117717	67.6982677
7	65.02462875	47.8732528
8	43.9358511	26.1496244
9	22.28438409	10.7528549
10	8.08369813	3.21640614
11	1.89508588	0.636398683
12	0.22001756	0.0740808688

$$H(z) = \frac{\sum_{i=0}^{12} b_i z^{-i}}{\sum_{j=0}^{12} a_j z^{-j}} \tag{3}$$

The filter coefficients are given in Table 7. The filter's amplitude and phase responses are shown up to the Nyquist frequency in Figure 24. Additionally, the amplitude and phase responses are shown over smaller frequency spans near the transition region in Figures 25 and 26. The transfer function is represented in the Z-plane using a pole-zero plot in Figure 27. Finally, the filter's step response is shown in Figure 28.

Uses: The filtering stage is performed after IQ samples are scaled for calibrated sensor gain, and before all other subsequent, parallelized processing stages. Therefore, the benefits of limiting the measurement bandwidth to the 10 MHz channel are carried on to each of the edge data payload components.

Calculation: The recorded IQ samples are filtered using the described IIR filter, implemented as a series of second-order filters with direct-form II transposed structure. This structure uses the minimum number of delay elements and multiplications, and implements the zeros before the poles. These qualities minimize numerical precision errors during filtering.

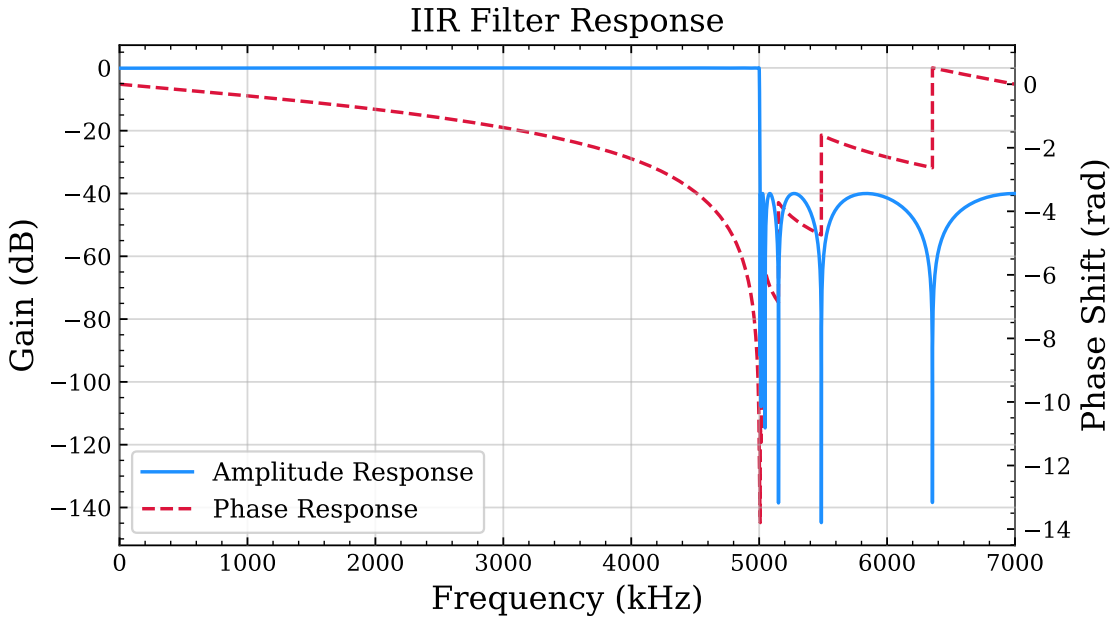

Fig. 24. The amplitude and phase response of the IIR filter, shown up to the Nyquist frequency. The amplitude response displays a sharp cutoff at 5 MHz.

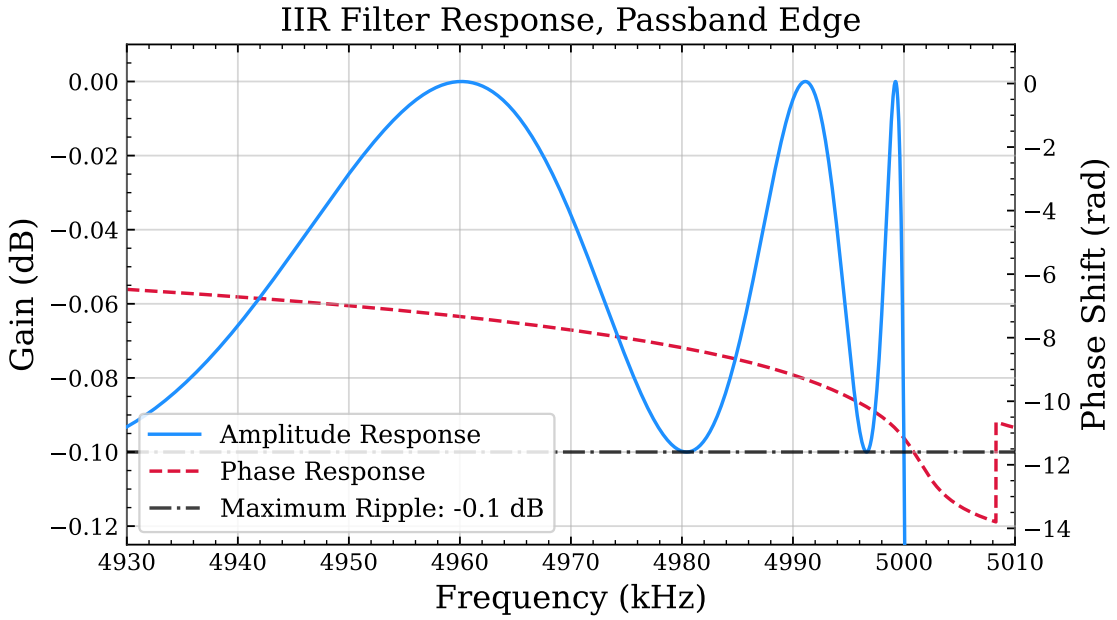

Fig. 25. The amplitude and phase response of the IIR filter, zoomed into the region from 4930 kHz to 5010 kHz to show the passband ripple.

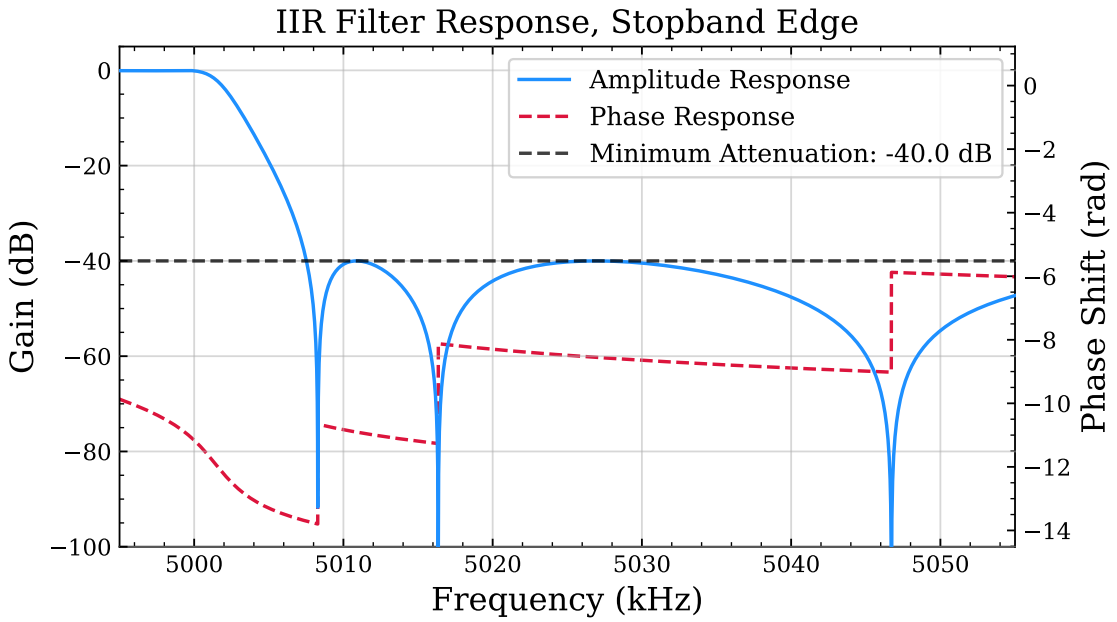

Fig. 26. The amplitude and phase response of the IIR filter, zoomed into the region from 4995 to 5055 kHz to show the transition and part of the stopband response.

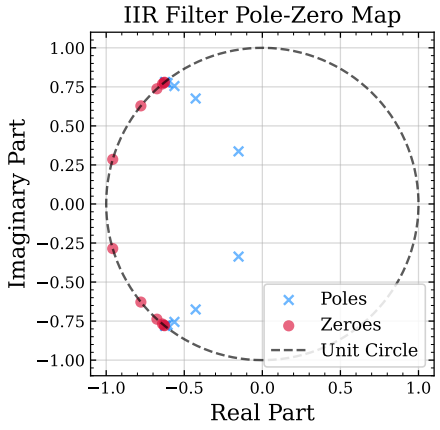

Fig. 27. A pole-zero map, displaying the poles and zeroes of the IIR filter in the complex Z-plane. All of the poles lie within the unit circle, indicating filter stability.

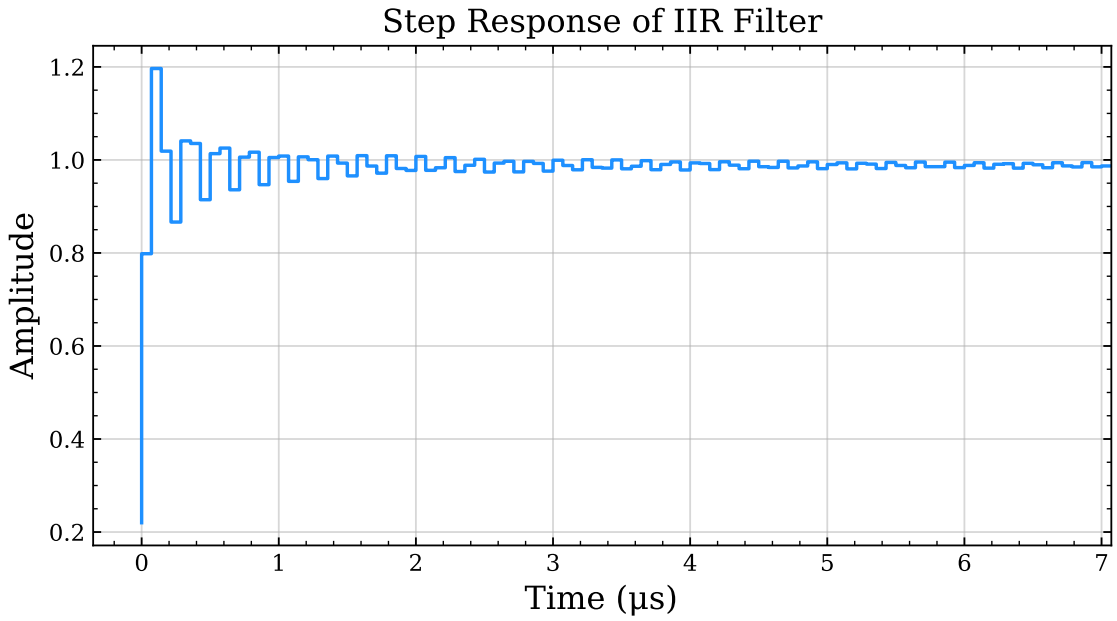

Fig. 28. Plot showing step response of the IIR filter, demonstrating the initial overshoot and subsequent ringing effects.

4.4. Power Statistics

4.4.1. Time-Series

Description: The channel power time-series quantifies the time-progression of mean and maximum channel power values across the duration of a waveform capture. In the realized implementation, peak and RMS power detectors are applied on short time-scales to give complementary insights about short bursts and average power, respectively. The time-resolution of the detectors, at 10 ms, is set to a single radio frame in the physical layer of cellular communication standards; insight on shorter time-scales is provided by the PFP, detailed in Section 4.4.3.

Background: This is a common type of measurement trace available on commercial test instruments, including spectrum analyzers and oscilloscopes. These instruments sometimes refer to this measurement as *channel power*, or *power vs time*.

Uses: The time-series can be used to identify the incident time and duration for exceptionally powerful signals present in the recorded waveform. Time-series traces of power can also be used to to compute summary statistics of the time-domain characteristics of each waveform capture. The linear power average, equivalent to the RMS power detection with bin size equal to the full duration of the waveform, is the mean of the RMS power trace. The instantaneous peak power of the capture is the maximum value of the peak power trace. These are useful metrics for quantifying average and instantaneous peak power recorded in the channel.

Calculation: A sequence of instantaneous power samples are calculated from IQ waveform samples by

$$P[n] = \frac{|\mathrm{IQ}[n]|^2}{2Z_0}, \tag{4}$$

where $Z_0 = 50\,\Omega$ is the reference impedance of the sensor input. The time-series power statistics are computed for a single-channel IQ capture by the following procedure.

1. Scale IQ samples by the calibrated sensor gain per Equation 2.
2. Filter IQ samples using the IIR filter described by Equation 3 and Table 7.
3. Convert IQ samples (Volts) to instantaneous power samples (Watts) using Equation 4.
4. Reshape instantaneous power data into 400 contiguous blocks of 140,000 samples each, with each block of samples representing power samples in contiguous 10 ms blocks of time.
5. *For the RMS power trace:* Take the mean of each block of samples, yielding 400 samples representing time-sequential RMS signal power on 10 ms timeframes.
6. *For the peak power trace:* Take the maximum of each block of samples, yielding 400 samples representing time-sequential peak signal power on 10 ms timeframes.

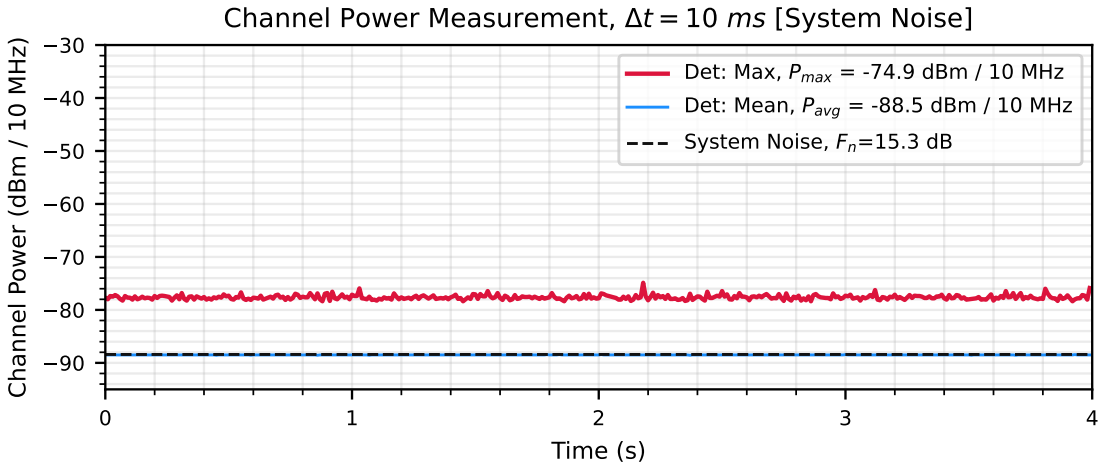

Fig. 29. Mean and max. detected power statistics over a 4 s interval for a capture of system noise in a 10 MHz channel, with a calibrated noise figure $F_n = 15.3$ dB. The predicted noise floor is shown as a black dashed line.

7. Convert both results from Watts to dBm.

Examples:

Examples of these power statistics are shown in Figures 29, 30, and 31. Figure 29 shows the results for a capture of just system noise. In this figure, the RMS power detector results accurately reflect the system's expected noise floor. Figure 30 shows the results for a suspected bin 1 radar signal. In this figure, the suspected radar is nearby the sensor, but the power over the 4 s period varies as the radar's boresight is rotated; the RMS power trace shows a increase above noise since the radar pulse repetition rate is smaller than the detector window of 10 ms. Figure 31 shows the results for a capture likely containing a commercial CBSD, where there is a more constant offset from the thermal noise floor.

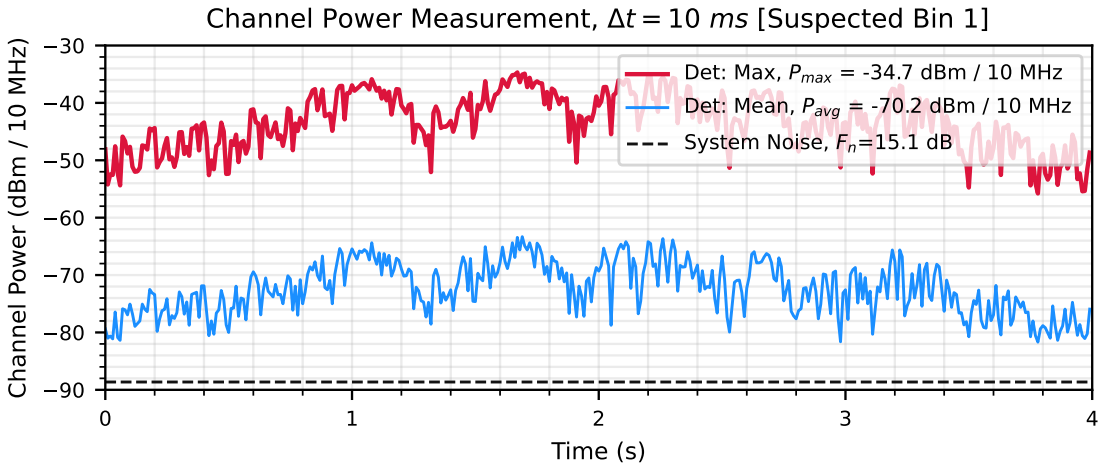

Fig. 30. Mean and max. detected power statistics over a 4 s interval for a suspected bin 1 radar in a 10 MHz channel, with a calibrated noise figure $F_n = 15.1$ dB. The predicted noise floor is shown as a black dashed line.

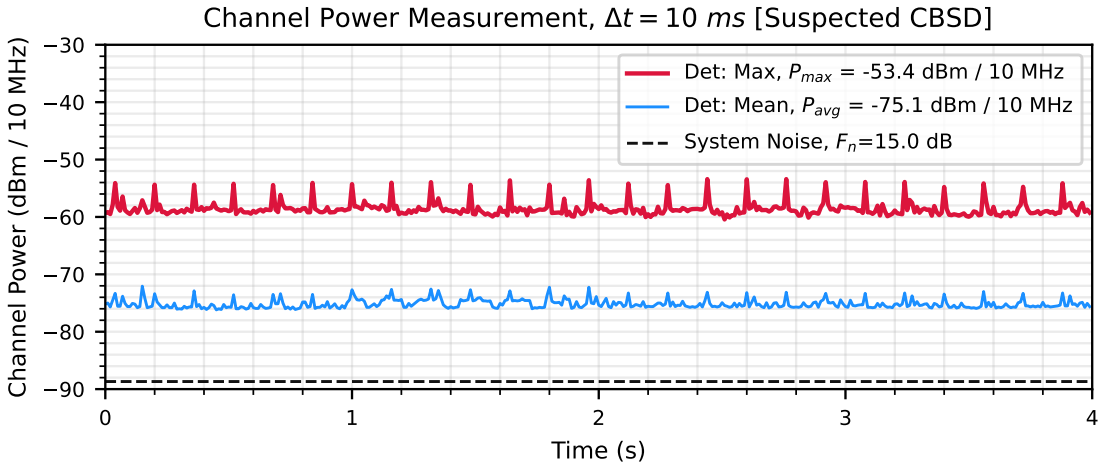

Fig. 31. Mean and max. detected power statistics of a capture likely containing a CBSD, with a calibrated noise figure $F_n = 15.0$ dB. The system's calibrated noise floor in the channel is shown as a black dashed line.

4.4.2. Power Spectral Density

Description: The power spectral density quantifies the distribution of signal power with respect to frequency within a waveform capture. A set of power spectra are computed from the IQ waveform, each with frequency bin size of $\Delta f = 80$ kHz. Peak and RMS statistical detectors are then applied to give complementary insights about short bursts and average power, respectively. The frequency resolution of the computed spectra is chosen to provide insight into the frequency distribution of the power of cellular communication signals within their resource blocks, and to allow for seamless aggregation of adjacent channel results. Windowing is used to reduce the undesirable effects of spectral leakage and scalloping loss.

Background: Representing RF signals as functions of frequency rather than time is a common practice in radio science and engineering. Viewed this way, it is possible to easily detect important signal features and characteristics, such as dominant signal frequencies or the presence of frequency-dependent signal structures. The discrete Fourier transform (DFT) provides a method for sampling the discrete-time Fourier transform (DTFT), which is equivalent within the Nyquist bandwidth to the Fourier transform. The DFT calculates the sequence $X[m]$ from an input $\mathrm{IQ}[n]$ by:

$$X[m] = \sum_{n=0}^{N-1} \mathrm{IQ}[n] \cdot \exp\left(-\frac{j2\pi nm}{N}\right) \tag{5}$$

The corresponding sequence of frequency values $F[m]$ is determined using the sampling rate F_s and center frequency f_c pertaining to the input sequence $\mathrm{IQ}[n]$ by:

$$F[m] = \frac{mF_s}{N} + f_0, \quad 0 \leq m \leq N-1 \tag{6}$$

The DFT is commonly implemented in practice using the fast Fourier transform (FFT) algorithm, which allows for the DFT to be computed much more efficiently than by using Equation 5 directly ($\mathcal{O}(N \log N)$ vs. $\mathcal{O}(N^2)$) [7].

One characteristic of the DFT is spectral leakage, which occurs due to the implicit assumption in the DFT that the input signal is exactly periodic over the length of the input. For any frequency component of the input which is not an exact multiple of Δf, this assumption fails. Since all recorded signals cannot be expected to be constrained in this way for any choice of Δf, leakage is practically unavoidable in DFT-based processing for this measurement application. Windowing is therefore used to reduce the effects of leakage. Windows are periodic functions which go to zero at the first and last sample of the DFT input, satisfying the periodicity assumption of the DFT. Many types of windows exist, each with various advantages and disadvantages. The flat top window was chosen due to its good performance in reducing spectral leakage as well as its optimal performance in reducing

Table 8. Coefficients for the flat top window used in PSD statistics processing.

A_0	0.21557895
A_1	-0.41663158
A_2	0.277263158
A_3	-0.083578947
A_4	0.006947368

scalloping error. The flat top window has worst-case scalloping error of 0.01 dB, while demonstrating maximal flatness in its main lobe [8]. The flat top window sequence is a 5th-order cosine function, defined by Equation 7 and Table 8.

$$w[n] = \sum_{l=0}^{L-1} A_l \cdot \cos\left(\frac{2\pi l n}{2N+1}\right) \tag{7}$$

The application of the window inherently reduces the amplitude and energy of the signal to which it is applied. In order to accurately measure the power spectral density at the calibration terminal, the window energy correction factor is calculated from the window by Equation 8.

$$\mathrm{ECF} = \sqrt{\frac{N}{\sum_{n=0}^{N-1} w[n]^2}} \tag{8}$$

Calculation: The power spectral density statistics are computed for a single-channel IQ capture by the following procedure. Note that the net effect of the various scaling factors applied is to remove the effects of windowing, normalize the FFT, and account for the equivalent noise bandwidth (ENBW) of the DFT bins. The results are the mean and maximum power spectral densities, with sample amplitudes in dBm/Hz, referenced to the calibration terminal.

1. Scale IQ samples by the calibrated sensor gain per Equation 2.
2. Filter IQ samples using the IIR filter described by Equation 3 and Table 7.
3. Scale IQ samples for the amplitude loss due to windowing, by multiplying each sample by the window energy correction factor defined in Equation 8.
4. Reshape IQ samples into 320,000 contiguous blocks of 175 samples each.
5. Apply the window by multiplying each block element-wise by the flat top window sequence defined by Equation 7 and Table 8.
6. Compute the DFT of each block (Equation 5 with $N = 175$) using the FFT algorithm.

7. Convert the DFT result amplitudes (Volts) to power (Watts) using Equation 4. This step include scaling to account for RF-to-baseband downconversion.
8. Scale power values to a power spectral density (W/Hz) by first dividing by the sampling rate F_s, then dividing by the DFT length N.
9. Sort the PSD samples by their corresponding frequencies.
10. Apply statistical detectors bin-wise; that is, the i-th sample in a detector result is determined from the i-th samples of all 320,000 PSD results.
11. *For each trace* Take the statistic of interest (mean, median, 99th percentile,etc,) of all PSD results, bin-wise.
12. Convert both PSD results from Watts/Hz to dBm/Hz.
13. Discard the first and last 25 samples from both PSD results, which represent frequency ranges outside the measured channel.

Examples:

Examples of these power statistics are shown in Figures 32, 33, and 34. The values P_{avg} are computed by summation of the mean PSD results with the appropriate bandwidth offsets applied for the bin width and window ENBW. Figure 32 shows the results for a measurement of system noise. In this figure, the mean PSD detector results accurately reflect the calibrated system's expected noise floor. Figure 33 shows the results for a suspected bin 1 radar signal centered between 2 10 MHz channels. In this figure, the main lobe and some sidelobes of the suspected radar signal appear as wide peaks in the maximum PSD result. The mean PSD shows a greatly reduced structure and the lower percentile traces do not show any structure due to the very brief radar pulse time. Figure 34 shows the PSD across all measured channels for 3530-3710 MHz for the same capture. This figure shows an example of multi-channel data aggregation: the data shown was computed from 18 separate, channelized IQ recordings. The choice of $N = 125$ allows for this aggregation with constant sample spacing, even across channel boundaries. In addition, this figure demonstrates that the frequencies 3530-3550 MHz were unused at this instant in time, most of the CBRS band (3550-3700 MHz) was allocated as 20 MHz grants, and that 3700-3710 MHz was occupied.

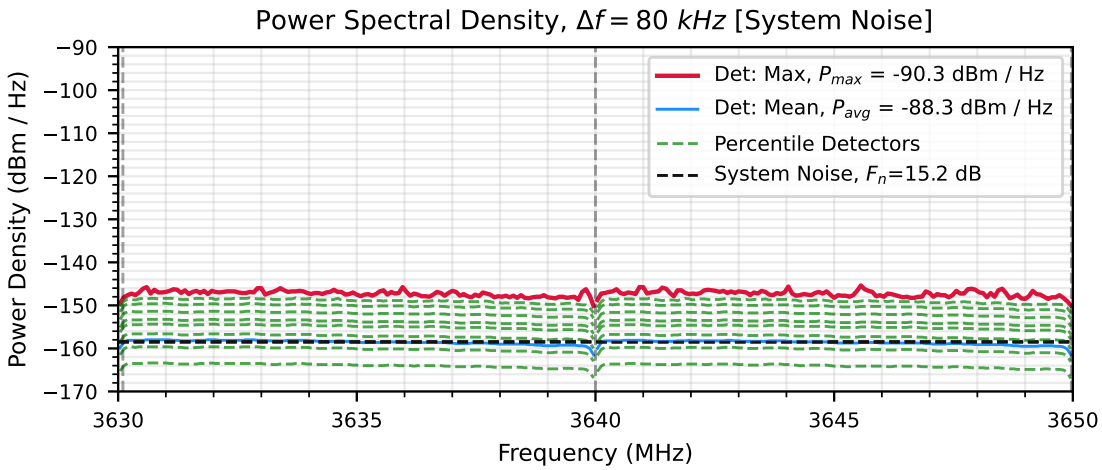

Fig. 32. Mean and max. detected power spectral density for simulated system noise in a 10 MHz channel, with simulated channel noise figure $F_n = 4.1$ dB. The predicted noise floor is shown as a black dashed line.

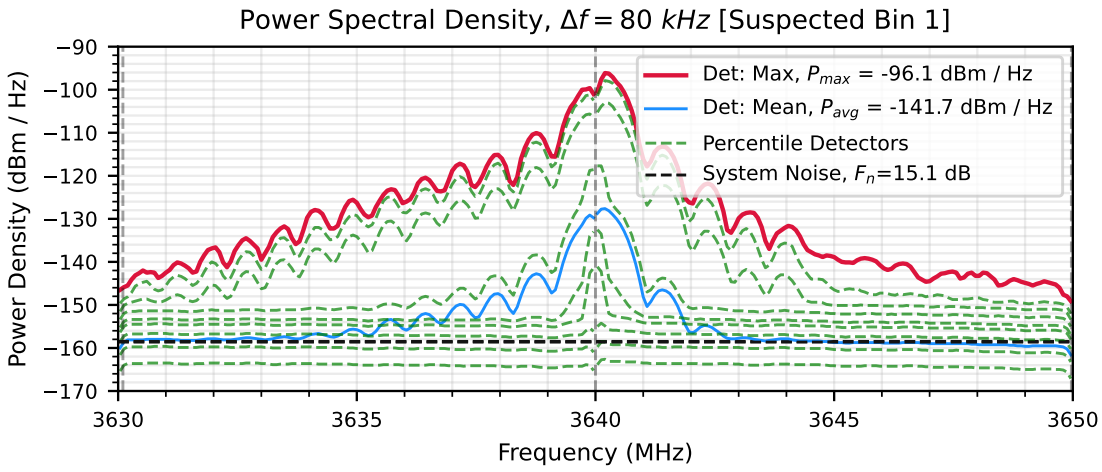

Fig. 33. Mean and max. detected power spectral density for a suspected bin 1 radar in adjacent 10 MHz channels. The predicted noise floor is shown as a black dashed line.

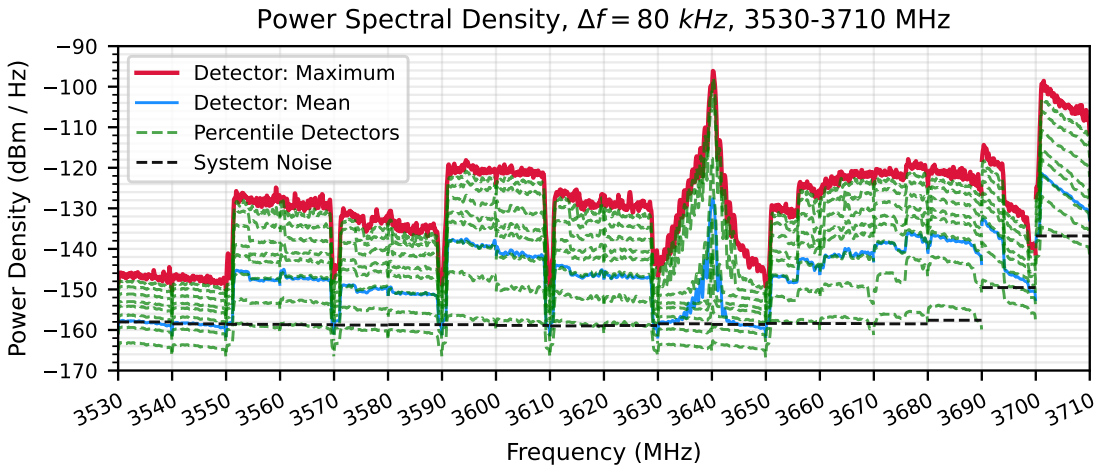

Fig. 34. Mean, max, median and selected percentiles of power spectral density for real-world signals in the CBRS band, with calibration-predicted channel noise floors shaded regions. Example PSD data product produced by a sensor in Norfolk, Virginia.

4.4.3. Periodic Frame Power

Description: We developed the periodic frame power measurement for this project to help differentiate between transmissions from CBSDs and others, including pulsed radar, and for forensic application pf SAS activation events. The underlying approach is to capture periodic channel occupancy characteristics that are fixed parameters of these different types of users [9], with special emphasis on the time division duplex (TDD) radio frame.

Background: Past monitoring campaigns support the idea that channel occupancy and power can be used to help classify between downlink and uplink. Downlink transmissions tend to exhibit high channel occupancy, and typically broadcast in almost every frame [10]. In contrast, uplink signaling, which often consumes battery power in mobile devices, is optimized to *minimize* transmission time; as a result, measurements of uplinks showed dramatically lower occupancy rates, sometimes by factors as large as 10^6 [10]. In addition to much lower occupancy, the source of DL transmissions is more intense and couples more directly with most practical antenna configurations for monitoring. These factors, when considered together, give a strong impetus to create fast time monitoring observations. Specifically, resolving statistical measures of channel power as a function of time on a relevant scale to the spectrum entrants and incumbents allows monitoring efforts the ability to differentiate relative use.

Each of the most widely-discussed types of incumbent and entrant users in CBRS have fixed and clearly defined periodic occupancy characteristics.

- **5G cellular** CBSDs operate TDD networks with frame duration 10 ms. A base station subdivides the frame into a schedule of interleaved downlink (base station) and uplink (user equipment) time allocations. The schedule of allocations has time resolution equal to 1 symbol, which in the CBRS band may have duration 1⁄56 ms, 1⁄28 ms, or 1⁄14 ms. Cellular standards define a large number of different scheduling patterns that may be selected by the base station. These may remain fixed during normal operation of the base station.
- **4G cellular** CBSDs operate TDD networks with frame duration 10 ms. A base station subdivides the schedule into a schedule of interleaved downlink and uplink time allocations. These allocations have time resolution of 1 ms. Only 7 different duplex scheduling patterns are supported, and remain fixed during normal operation of a base station.
- **IEEE 802.16 (WiMaX)** CBSDs operate TDD networks. A network operates in a fixed frame duration, which must be one of {2, 2.5, 4, 5, 8, 10, 12.5, 20} ms [11]. Of these, equipment is only required to support 5 ms for WiMaX Forum certification, making this a common assumption for the actual frame duration in these networks. Recent standards also define a *superframe* as a sequence of 5 ms frames, reinforcing 5 ms as the base periodicity constant.

- **Bin 1 Radars** are pulsed radars operating with pulse repetition rate of very close to 1 ms.

In each of these cases, a fixed division period is defined that is further subdivided to determine transmission timing. When transmissions repeat occupancy patterns within these periods, we have an opportunity to simplify the short-time features of the channel power by characterizing their typical behavior.

The key to performing power measurements of these characteristics is to define a *reference frame* that is a common multiple of all division periods under study. For our realized implementation, we selected 10 ms (Table 6), since it is the least common multiple of 1 ms, 5 ms, and 10 ms.

Calculation: The calculation of periodic frame power begins with the time-series of instantaneous power samples P_n calculated from equation (4), and proceeds as follows:

1. Partition the time-series into M contiguous reference frames, following the reference frame duration (Table 6). For a reference waveform capture length of 4 s, this produces $M = 400$ frames per waveform.
2. Partition each frame into N power bins. The time-duration of these bins should be no longer than the shortest possible transmission occupancy event. The reference duration of the power bins corresponds with the shortest possible duration of a single symbol defined for CBRS in 5G cellular standards: including the cyclic prefix, this duration is approximately 17.9 µs. This results in $N = 560$ power bins per frame.
3. Apply a power detector to evaluate channel power in each power bin. The realized implementation evaluates both peak and RMS power detectors, which are reported as separate measurement traces.
4. Compute statistics across n^{th} power bin of each frames. The resulting frame power measurement trace has M samples, each of which has been computed across the N frames. The realized implementation records $\{\text{min}, \text{mean}, \text{max}\}$ for each of the peak and RMS power detectors.

The reference frame duration should also be an integer multiple of the waveform sampling period.

Examples: The interpretation and usage of PFP is most easily conveyed by concrete example. To this end, the same capture of the three signal types is used.

When little to no external signal is present as in Figure 35, the PFP is composed entirely of noise power samples. With a measured noise figure of 15.3 dB, most of this power produced inside the sensor receiver. The solid lines represent the mean of the RMS power detector (blue) and peak power detector (red) readings; the average at each point is evaluated across the $M = 400$ data points evaluated at the same "cyclic time," which is the time elapsed into each 10-ms block. The shaded envelopes around each trace bound the

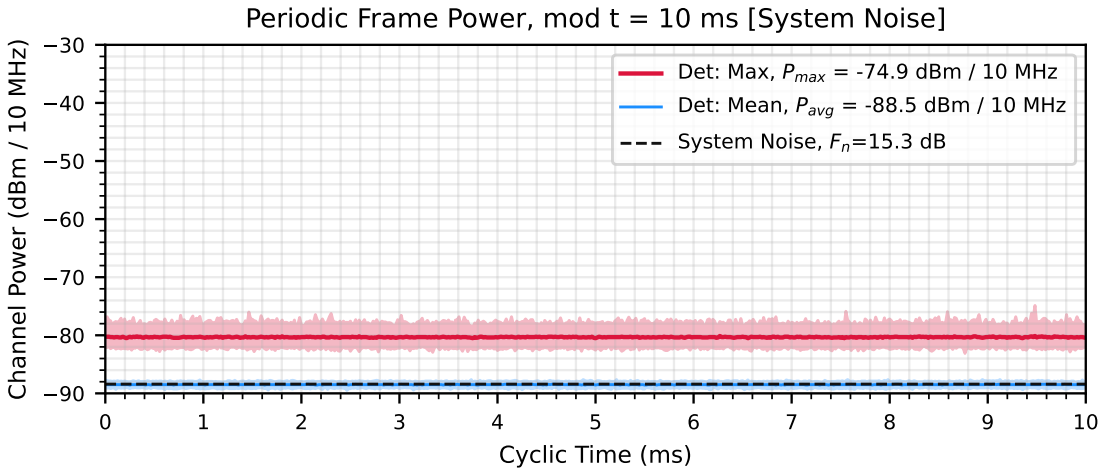

Fig. 35. Peak and RMS-detected PFP results for system noise in a 10 MHz channel, with a measured noise figure $F_n = 15.3$ dB. The predicted noise floor is shown as a black dashed line.

extrema across the same M data points. Because the expected value of the thermal noise power in the receiver is constant, there is no time structure here except for random noise variability. Notice that the extrema for the RMS power detector are particularly close together, because the averaging detector has reduced the standard error of the noise power measurement.

Continuing to the suspected bin 1 radar activity produces the PFP illustrated by Figure 36. The pulse repetition interval of the waveform, which is very close to 1 ms, repeats almost exactly 10 times per frame. As a result, peaks occur close to the same cyclic time elapsed in each frame, at least relative detector resolution of 17.9 µs. The periodic characteristic is then very clear by inspection of the plot. Meanwhile, the RMS detector also shows a similar cyclic time structure, but the magnitude difference between the peak- and RMS-detected traces increases beyond that of the thermal noise due to the different nature of the signal and the fluctuation of that signal throughout the 4 s capture.

Figure 37 shows the PFP applied to a capture of a suspected CBSD signal. The periodic response can be divided into two regions: a TDD downlink section which can be further subdivided into segments of the TDD resource allocation and a TDD uplink portion section consisting of thermal noise in the case of the showcased capture. It should be noted that the 17.9 µs cyclic time resolution is not an exact multiple of the LTE symbol duration of 71.4 µs (nearly a factor of 4, but not exactly due to the sample rate) nor is the PFP directly aligned with the LTE frame, so it should not be assumed that the PFP bins align with the LTE symbols.

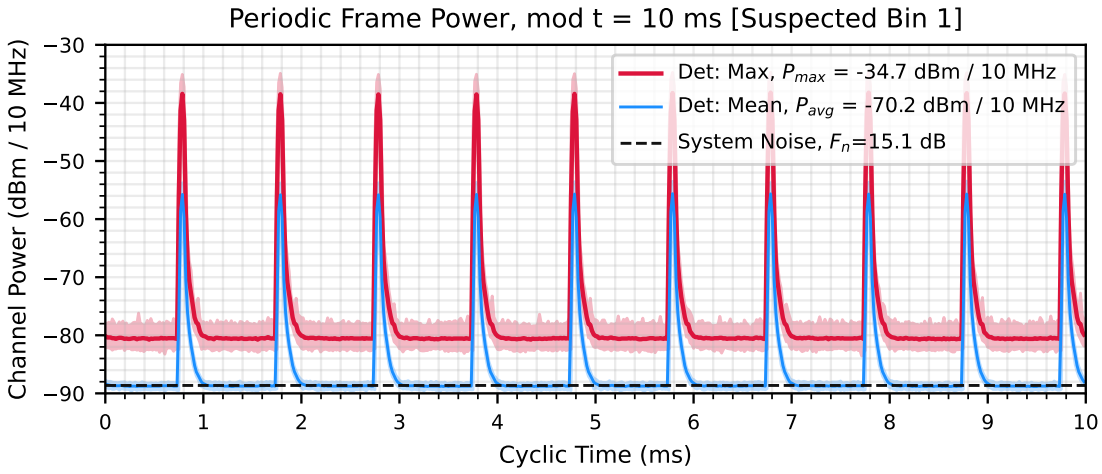

Fig. 36. Peak- and RMS-detected PFP measurements for a suspected bin 1 radar and system noise in a 10 MHz channel, with measured channel noise figure $F_n = 15.1$ dB. The predicted noise floor is shown as a black dashed line.

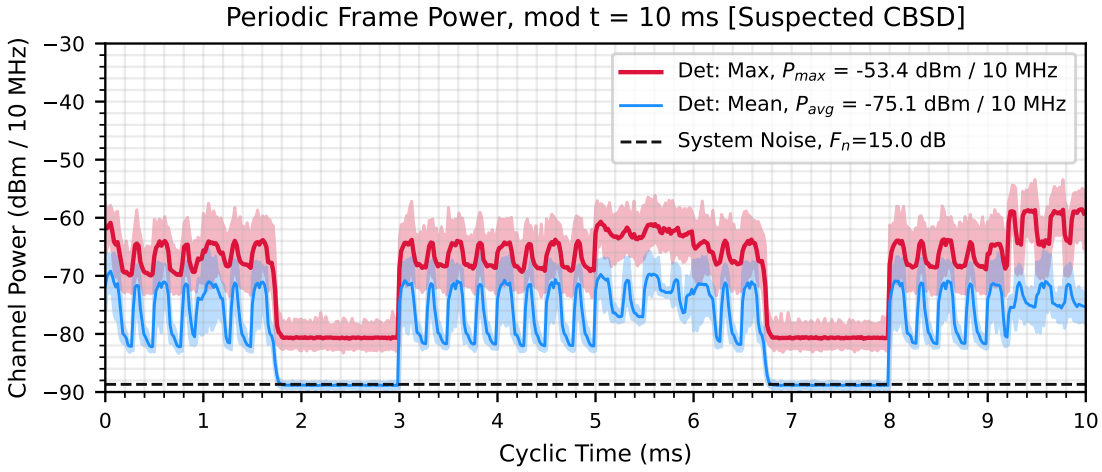

Fig. 37. Peak and RMS-detected PFP results for a suspected CBSD signal, with the calibration-predicted channel noise floor shown as a black dashed line.

Uses: A primary application of PFP is as a forensic tool to investigate CBRS DPA potential activation events. A real-world example is illustrated by Figure 38, for an event observed by the prototype sensor deployed at the Norfolk International Terminal (NIT) site in Virginia.

In order to start with a view across a longer time period, simple statistics were on the PFP trace in each capture for 16 minutes in Figure 38(a): a “capture median” (the median of average RMS detector readings), and “capture max” (maximum of the peak detector

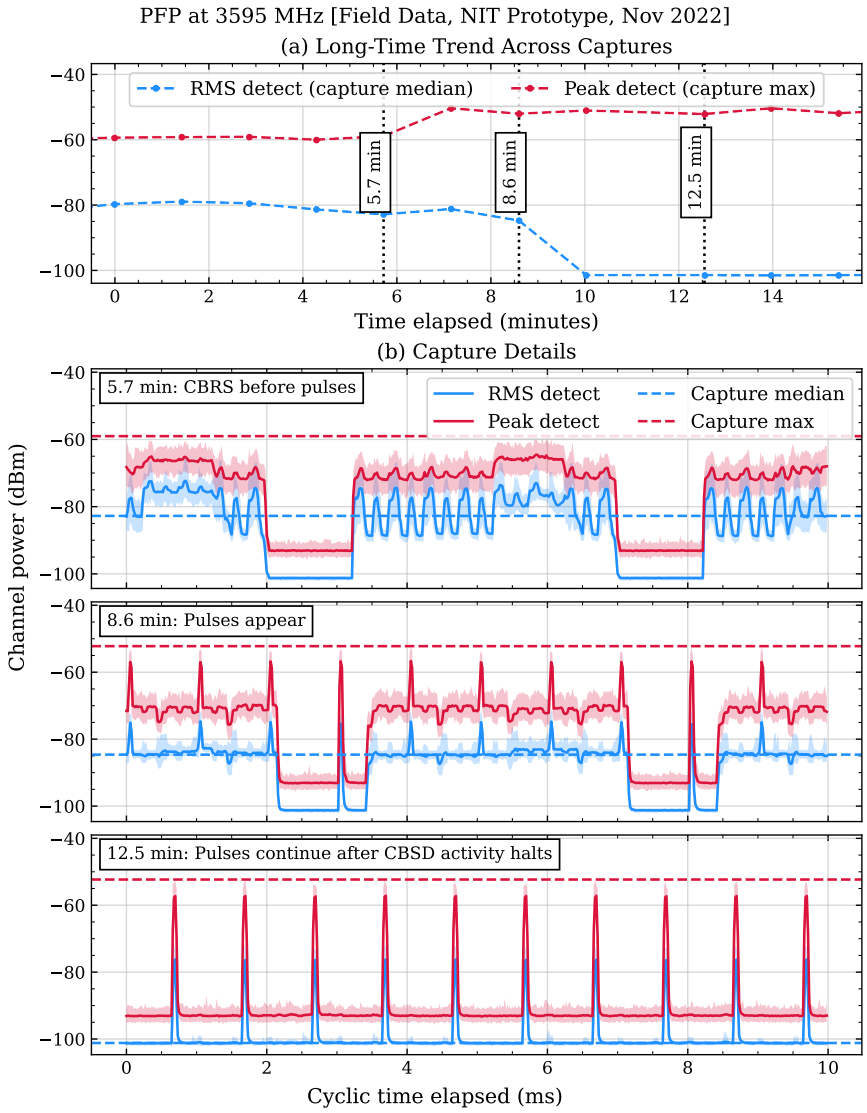

Fig. 38. Measurements of PFP during a possible DPA activation event. First, time series of capture-by-capture statistics (a) exhibits an abrupt drop in median RMS power. On further investigation, PFP traces in (b) detail CBSD activity before the event ("Capture 1"), followed by the appearance of pulses with 1 ms pulse repetition interval ("Capture 2"), and later the same pulse pattern but without CBSDs. The first capture with pulse activity occurred approximately 3 min before the first capture without CBRS.

readings). Just after 7 min time elapsed, a capture maximum power reading has increased by about 10 dB, without significant change in the capture median. This suggests that a pre-existing signal has continued while short bursts have become present. Then, near 10 min elapsed, the capture median falls to the receive system noise level: for most samples in these captures, there is no detectable activity in the channel.

Figure 38(b) details PFP captures at the 3 key points marked along sub-figure (a). First, at 5.7 min., we see the envelope of ongoing TDD CBSD activity in the channel. By 8.6 min., a pulse train has emerged that is superimposed on the pre-existing channel activity; here, the variability in the RMS trace has decreased significantly, indicating a change in CBSD occupancy behavior, though the duration of its "off" time remains unchanged. By 12.5 min., the CBSD activity has stopped, and the pulse behavior has changed. The pulse repetition interval, at approximately 1 ms, is consistent with that of a bin 1 radar, though this is not by itself a conclusive indication of that this specific type of incumbent user is present in the channel.

When combined with other data streams, a PFP analysis like the above represents a powerful tool to add confidence to conclusions about the CBRS environment. For example, comparisons against logs of SAS channel grants lead CBSDs to respond as expected in terms of physical channel radio channel occupancy. In test cases where "ground truth" data is available that can confirm the presence of incumbents in the sensor's DPA, the PFP can be used to help determine whether power detected by the SEA sensor originates from incumbent or CBSD transmissions.

4.4.4. Amplitude Probability Distribution

Description: An APD is the complementary cumulative distribution function (CCDF) of the instantaneous channel power, evaluated empirically. It is evaluated across the full duration of each channel waveform capture. Physically, it is equivalent to the percentage of time in which the capture exceeds a specified power threshold. Plots of the APD provide qualitative insights to the duty cycle of a measured signal. Rare occurrences of high amplitude samples across a full capture may indicate the presence of a strong, briefly pulsed signal, such as pulsed radar. Quantitative analysis may also leverage the APD to estimate occupancy characteristics in the channel. For example, assuming bursted noise-like transmissions in CBRS wireless data networks, the transmission time-occupancy rate may be estimated based on the deviation between the APD and an assumed underlying distribution, such as idealized Gaussian noise. For a more complete discussion of the APD see Appendix A.

Background: The original application for the APD was for the characterization of noise originating in the atmosphere at low frequencies [12]. Because the APD makes no assumption about time structure of the electromagnetic environment, it can be computed independent of wireless networking parameters, unlike periodic frame power (PFP). In conventional graphical representation, the vertical axis is power transformed to decibel units; the horizontal axis is the "percent exceeding ordinate" with a transformation to $Q-Q$ probability space such that idealized thermal noise can be represented as a straight line.

Following Achatz, Cotton, and Dalke [13], we calculate a discrete estimate of the continuous APD. The continuous distribution is expressed in terms of the random variable $\mathscr{X}$, a function which assigns a real number $\mathscr{X}(v)$ to each element v of a sample space. In these terms, the APD is expressed in terms of the probability operator $\mathscr{P}$ and signal amplitude a by Equation 9.

$$\mathscr{F}(a) = \mathscr{P}(\mathscr{X}(v) > a) \tag{9}$$

The discrete APD estimate $\mathscr{F}[n]$ is obtained from the input of N complex baseband samples, IQ$[n]$. First the amplitudes are defined as the ordered amplitudes, from smallest to largest, of the IQ samples, as defined in Equation 10.

$$a[n] = \mathrm{sort}\big(\big|\mathrm{IQ}[n]\big|\big) \tag{10}$$

And the probabilities are defined based on these sorted amplitudes by Equation 11.

$$\mathscr{P}[n] = \mathscr{P}\big(\mathscr{X} > a[n]\big) = 1 - \frac{n}{N} \tag{11}$$

This method of obtaining a discretized APD is simple and accurate, yet not practical for continuous real-time sensing. The operation to sort all samples as per Equation 10 is computationally costly, which slows down the sweep time across the band. Additionally, the APD output vector has, for both axes, the same dimension as the input vector. This situation is (1) far too large to stream back over the available backhaul solution, at 56 megasamples per 10 MHz channel, and (2) far more detailed along the probability axis than is useful. In a successful effort to optimize IQ post-processing time and output data size, an optimized method was implemented which is based on the definitions of the APD estimate above. In the optimized routine, amplitudes are replaced with amplitude bins, defined based on a parameterized power bin size. The bin edges are defined to span from the minimum amplitude value up to (potentially including but not exceeding) the maximum amplitude value, with uniform spacing by the determined bin size. The output size of the optimized APD therefore depends both on the choice of bin size parameter as well as the dynamic range of the recorded signal, and varies from one capture to the next. Then, the indices are found in the sequence of bin edges at which the unsorted amplitude samples could be inserted while maintaining the sorted ordering. The numbers of occurrences of each index are then counted to give the distribution of amplitude values within in each bin. Finally, these occurrences are cumulatively summed and subtracted from the input length N, before being normalized. This method removes the requirement for a large-input sort operation, and solves both previously mentioned issues. The optimized method trades reduced resolution in the result for faster computation time and lower output data rates, while still providing enough information to be useful for analysis.

Calculation: The APD is computed for each single-channel waveform capture as follows:

1. Scale IQ samples by the calibrated sensor gain per Equation 2.
2. Filter IQ samples using the IIR filter described by Equation 3 and Table 7.
3. Calculate IQ sample amplitudes by taking the absolute value of each sample, $\left|\mathrm{IQ}[n]\right|$.
4. If any amplitudes are zero-valued, replace them with NaN to avoid floating point arithmetic errors.
5. Generate the sequence of amplitude bin edge values, beginning with the smallest amplitude sample and increasing in uniform steps of half the parameterized power bin size, up to but not exceeding the maximum amplitude value.
6. Using a binary search, locate the indices in the sequence of amplitude bin edges at which the unsorted amplitudes could be inserted while maintaining the sorted bin ordering.
7. Count the number of occurrences of each located index value, representative of the number of amplitude samples contained in each bin.
8. Cumulatively sum the sequence of these counts, and subtract each value from the number of input IQ samples N.

9. Normalize the resulting sequence by dividing each value by the number of input IQ samples N. The results are CCDF probabilities, within the interval $[0, 1]$.
10. Scale the resulting amplitude bin values from dBV amplitudes to power samples in dBm, accounting for system impedance and the power-scaling factor of 2 associated with the down-conversion of the RF signal to baseband.

Examples:

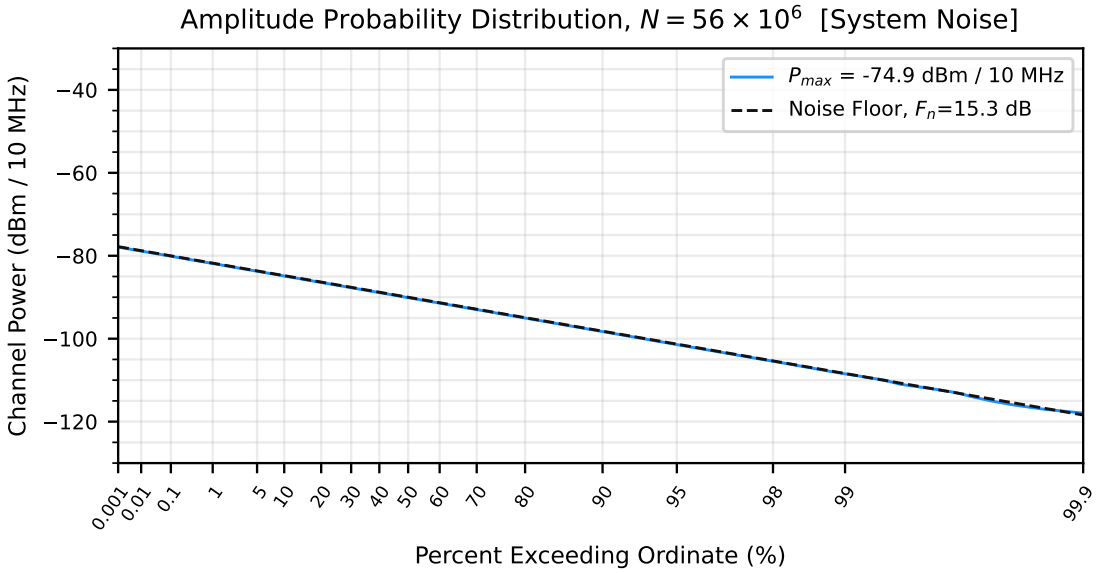

Fig. 39. Downsampled APD result for system noise in a 10 MHz channel, with measured noise figure $F_n = 15.3$ dB. The predicted APD of the system noise is shown as a black dashed line.

The APD is described in greater detail in Appendix A including the effects of mixture distributions, but it is worth noting here the difference between the predicted noise floor and the noise samples in the capture APD in Figure 41. The deviation from the predicted noise is a function of the number of noise samples in the mixture distribution. Since the uplink portion consists of thermal noise, but only consists of 20 % of the total capture, the sampling of the thermal noise is reduced by 80 %, resulting in a shift away from the predicted thermal noise line. Intuitively, the mean of the noise remains constant whether it is taken across the entire 4 s capture as in Figure 39 or across just the thermal noise portion of the capture, as in Figure 41, and would have the same quantiles. However, since the thermal noise samples are only a subset of the overall mixture distribution in Figure 41, this results in an effective "lateral shift" of the thermal noise portion of the distribution from the ideal noise line. The same effect occurs in Figure 40, however since the duty cycle of the bin 1 radar is very low, the subset of samples associated with thermal noise is quite high and the resulting shift from the ideal thermal noise line is nearly negligible.

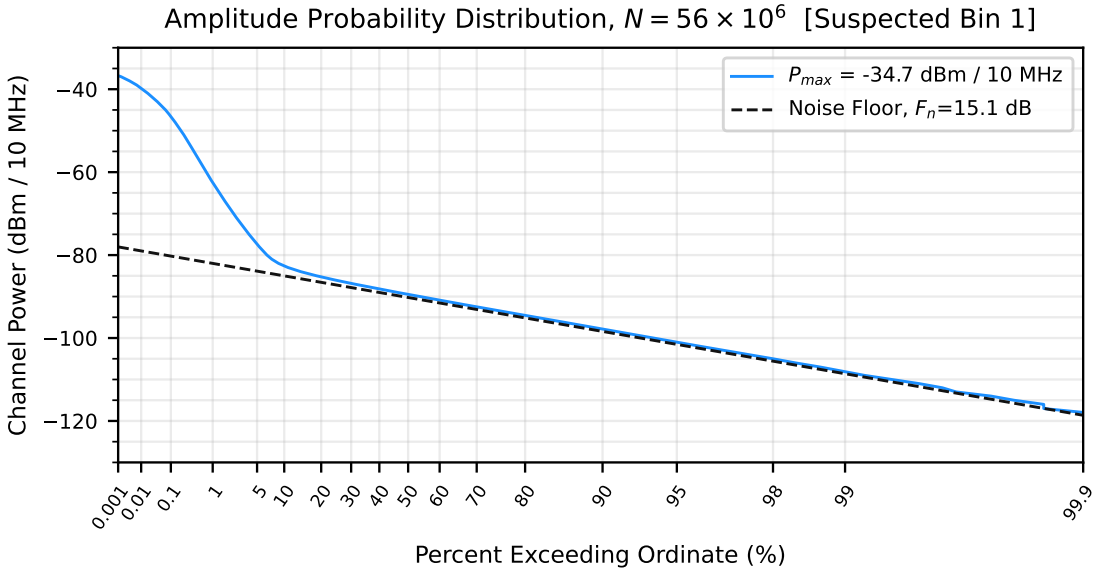

Fig. 40. Downsampled APD result for system noise and a suspected bin 1 radar in a 10 MHz channel, with measured noise figure $F_n = 4.1$ dB. The predicted APD of the system noise is shown as a black dashed line.

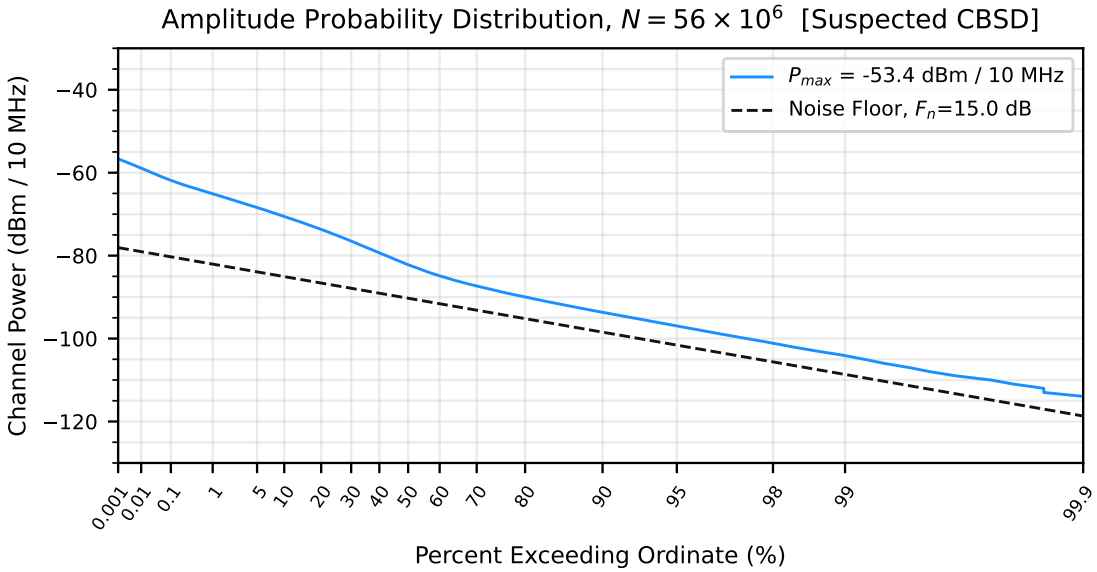

Fig. 41. Downsampled APD result for a capture of a suspected commercial CBSD, with the predicted APD of the system noise shown as a black dashed line.

5. Conclusion

The purpose of the NASCTN CBRS SEA project is to provide data-driven insight into the CBRS sharing ecosystem and to track changes in the spectrum environment over time. The implementation of this project included the installation of sensor systems at selected sites on the east and the west coast of the United States, along with a control and prototyping system in Boulder, Colorado. In order to facilitate stakeholder interactions with the data produced from those sensors, several key documentary goals must be met. Among these goals are the methods for deriving, accessing, and manipulating the data products, each of which are met by this document. For further information on NASCTN please visit https://www.nist.gov/ctl/nasctn or contact the program at nasctn@nist.gov

References

[1] (2024) Draft cbrs ecosystem sharing assessment (sea) test and metrology test plan. Available at https://www.nist.gov/system/files/documents/2024/09/04/NASCTN_CBRS_SEA_Draft_Task1_Testplan%20-%20September%202024.pdf.

[2] (2023) Draft cbrs ecosystem sharing assessment (sea) test and metrology test plan: Characterize aggregate emissions within the cbrs band 3550-3700 mhz in at least one always-on dpa. Available at https://www.nist.gov/system/files/documents/2023/12/19/NASCTN%20CBRS%20SEA%20Task%202%20Draft%20Test%20Plan%20-%20December%202023.pdf.

[3] Signal metadata format (sigmf). Available at https://github.com/gnuradio/SigMF.

[4] The pandas development team pandas-dev/pandas: Pandas. https://doi.org/10.5281/zenodo.3509134. Available at https://github.com/pandas-dev/pandas

[5] Kuester D, Romaniello A (2025) nasctn-sea-ingest. Available at https://github.com/usnistgov/nasctn-sea-ingest.

[6] (2020) Ieee standard for spectrum characterization and occupancy sensing. *IEEE Std 80215223-2020* :1–58https://doi.org/10.1109/IEEESTD.2020.9253019

[7] Cooley JW, Tukey JW (1965) An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation* 19(90):267–301. https://doi.org/10.2307/2003354

[8] Dalke R (2010) Radio Spectrum Estimates Using Windowed Data and the Discrete Fourier Transform (National Telecommunications and Information Administration), 10-470.

[9] Kuester D, Romaniello A, MATHYS P (2024) Cyclostationary channel power measurements for cbrs coexistence assessment (National Radio Science Meeting (NRSM), Boulder, CO, US). Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=956490.

[10] Kuester D, Lu X, et al. (2022) Radio spectrum occupancy measurements amid covid-19 telework and telehealth. https://doi.org/https://doi.org/10.6028/NIST.TN.2240. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=934697

[11] (2018) Ieee standard for air interface for broadband wireless access systems. *IEEE Std 80216-2017 (Revision of IEEE Std 80216-2012)* :1–2726https://doi.org/10.1109/IEEESTD.2018.8303870

[12] Hoff RS, Johnson RC (1952) A statistical approach to the measurement of atmospheric noise. *Proceedings of the IRE* 40:185–187. https://doi.org/10.1109/JRPROC.1952.274022

[13] Achatz R (2004) Estimating and graphing the amplitude probability distribution function of complex-baseband signals (ieee p802.15-04-0428-00). Available at https://grouper.ieee.org/groups/802/15/pub/04/15-04-0428-00-003a-estimating-and-graphing-amplitude-probability-distribution-function.pdf.

[14] Horner F, Harwood J (1956) An investigation of atmospheric radio noise at very low frequencies. *Proceedings of the IEE-Part B: Radio and Electronic Engineering* 103(12):743–751.

[15] Watt A, Maxwell E (1957) Measured statistical characteristics of VLF atmospheric radio noise. *Proceedings of the IRE* 45(1):55–62.

[16] Watt A, Coon R, Maxwell E, Plush R (1958) Performance of some radio systems in the presence of thermal and atmospheric noise. *Proceedings of the IRE* 46(12):1914–1923.

[17] Furutsu K, Ishida T (1961) On the theory of amplitude distribution of impulsive random noise. *Journal of Applied Physics* 32(7):1206–1221.

[18] Crichlow W, Roubique C, Spaulding A, Beery W (1960) Determination of the amplitude-probability distribution of atmospheric radio noise from statistical moments. *Journal of Research of the National Bureau of Standards* 64D(1):49–56.

[19] Tobias P (2003) Sec. 8.2.2.1 Probability Plotting. *NIST/SEMATECH e-Handbook of Statistical Methods* (National Institute of Standards and Technology, Gaithersburg, MD). Available at https://www.itl.nist.gov/div898/handbook/apr/section2/apr221.htm.

[20] Crichlow W, Spaulding A, Roubique C, Disney R (1960) *Amplitude-Probability Distributions for Atmospheric Radio Noise* (National Bureau of Standards). Monograph 23 Available at https://nvlpubs.nist.gov/nistpubs/Legacy/MONO/nbsmonograph23.pdf.

[21] Spaulding A, Roubique C, Crichlow W (1962) Conversion of the amplitude-probability distribution function for atmospheric radio noise from one bandwidth to another. *Journal of Research of the National Bureau of Standards* 66D(6):713–720.

[22] Shepelavey B (1963) Non-Gaussian atmospheric noise in binary-data, phase-coherent communication systems. *IEEE Transactions on Communications Systems* 11(3):280–284.

[23] Beckmann P (1964) Amplitude probability distribution of atmospheric radio noise. *Journal of Research of the National Bureau of Standards* 68D(6):723–736.

[24] Spaulding A, JS W (1985) Atmospheric Radio Noise: Worldwide Levels and Other Characteristics (National Telecommunications and Information Administration), TR-85-173. Available at https://its.ntia.gov/publications/download/85-173.pdf.

[25] Achatz R, Lo Y, Papazian P, Dalke R, Hufford G (1998) Man-made noise in the 136 to 138-MHz VHF meteorological satellite band (National Telecommunications and Information Administration), TR 98-355. Available at https://www.its.ntia.gov/publications/download/TR-98-355.pdf.

[26] Parsons JD (2000) *The Mobile Radio Propagation Channel* (Wiley, New York), 2nd Ed.

[27] Hoffman R, Cotton M, Achatz R, Statz R, Dalke R (2001) Measurements to determine potential interference to gps receivers from ultrawideband transmission systems (National Telecommunications and Information Administration), TR-01-384. Available at https://its.ntia.gov/publications/details?pub=2415.

[28] Wagstaff A, Merricks N (2005) Man-made noise measurement programme. *IEE Proceedings - Communications* 152(3):371–377. https://doi.org/https://doi.org/10.1049/ip-com:20045025
[29] Cotton M, Achatz R, Wepman J, Bedford B (2005) Interference potential of ultrawideband signals (National Telecommunications and Information Administration), TR-05-419. Available at https://its.ntia.gov/publications/download/TR-05-419.pdf.
[30] Papazian P, Lemmon J (2011) Radio channel impulse response measurement and analysis (National Telecommunications and Information Administration), TR-11-476. Available at https://its.ntia.gov/publications/download/11-476.pdf.
[31] Cotton M, Vu L, Eales B, Hicks A (2020) 3.45–3.65 GHz Spectrum Occupancy from Long-Term Measurements in 2018 and 2019 at Four Coastal Sites (National Telecommunications and Information Administration), TR-20-548. Available at https://its.ntia.gov/publications/download/TR-20-548.pdf.
[32] International Telecommunication Union (2024) Recommendation ITU-R P.372-17 Radio Noise. Available at https://www.itu.int/rec/R-REC-P.372-17-202408-I/en.
[33] Hufford G Chapter 15. Statisical Distributions. Unpublished notes.
[34] Achatz R, Cotton M, Dalke R (2004) Estimating and graphing the amplitude probability distribution function of complex-baseband signals, IEEE 802.15 Working Group for Wireless Personal Area Networks, Document Control Number 428. Available at https://mentor.ieee.org/802.15/dcn/04/15-04-0428-00-003a-estimating-and-graphing-amplitude-probability-distribution-function.pdf.
[35] Siddiqui MM (1962) Some problems connected with Rayleigh distributions. *Journal of Research of the National Bureau of Standards* 66D(2):167–174.
[36] Casella G, Berger RL (2002) *Statistical Inference* (Duxbury, Pacific Grove, CA), 2nd Ed.
[37] Molisch AF (2023) *Wireless Communications* (John Wiley & Sons).

Appendix A. Understanding the Amplitude Probability Distribution (APD) Plot

Statistical attributes for the envelope of received radio frequency signals and noise can be summarized with a special graph called the amplitude probability distribution (APD) plot. Although it has been used in its current form since the 1960s, many radio engineers are not familiar with the graph and its utility. The purpose of this appendix is to clarify the theoretical foundations of APD plots and to collect references illustrating their historical development and applications.

A.1. Introduction

Measurements of the APD to characterize ambient levels of radio noise have been performed since at least the 1950s, e.g., [14–17], using plots of the probability that a received amplitude exceeds a threshold[2]. In 1960, Crichlow et al. [18] introduced the form of the APD plot that has been primarily used since that time. Namely, the amplitude exceedance threshold, in decibels, is plotted versus the exceedance probability on a special scale, sometimes called "Rayleigh probability paper," defined so that a Rayleigh distribution is a straight line.[3]

Many publications have employed APD plots to characterize radio frequency emissions, e.g., [20–32]. The majority of these publications originated from the the National Bureau of Standards (NBS) Central Radio Propagation Laboratory (CRPL) and its successor, the National Telecommunications and Information Administration (NTIA) Institute for Telecommunication Sciences (ITS) in Boulder Colorado.

The following sections cover the theory underlying the definition of the APD plot and discuss interpretations. Further details on the derivation and implementation of APD plots can be found in [22, 23, 33, 34] [27, Appendix E] [29, Appendix D].

A.2. Rayleigh Distribution Theory

First, we review statistical distribution theory for white Gaussian noise (WGN) in I/Q data, which arises both via thermal noise in radio receivers and in communication channels exhibiting Rayleigh fading. Below, we assume basic familiarity with random variables and probability distributions.

Let $X = X_I + jX_Q$ with $j^2 = -1$ be a complex baseband signal with real in-phase (I) and quadrature (Q) components X_I and X_Q, respectively. WGN in the I and Q components is typically well-modeled with independent, identically distributed normal distributions with zero mean and variance σ^2, denoted $X_I \sim \mathcal{N}(0, \sigma^2)$ and $X_Q \sim \mathcal{N}(0, \sigma^2)$. In this case, it

[2] In statistics, this function is called the complementary cumulative distribution function (CCDF); see Sec. A.2

[3] Before the availability of modern computers and fast iterative fitting algorithms, probability graph paper, which is scaled so that a specified distribution graphs as a straight line, was a common data analysis tool; see [19] for examples.

is straightforward to show that the envelope amplitude $R = \sqrt{X_I^2 + X_Q^2}$ follows a Rayleigh distribution [35]. Specifically, for $r \geq 0$, the probability density function (PDF) of R is

$$f_R(r) = \frac{r}{\sigma^2} \exp\left[-r^2/(2\sigma^2)\right] \tag{12}$$

and the cumulative probability density function (CDF) is

$$F_R(r) \equiv P(R \leq r) = 1 - \exp\left[-r^2/(2\sigma^2)\right]. \tag{13}$$

Above, $P(\cdot)$ denotes the probability of an event. When R follows a Rayleigh distribution with scale parameter, σ, we denote this fact by writing $R \sim \text{Rayleigh}(\sigma)$. The inverse CDF, called the quantile function, can be used to calculate specific quantiles (or percentiles). Namely, the quantile function for $R \sim \text{Rayleigh}(\sigma)$ is

$$Q_R(p) \equiv F_R^{-1}(p) = \sqrt{-2\sigma^2 \ln(1-p)} \qquad 0 \leq p \leq 1. \tag{14}$$

In many radio engineering contexts, it is desirable to assess the probability that the received amplitude, R, exceeds a given threshold, r. When $R \sim \text{Rayleigh}(\sigma)$, the exceedance probability is

$$\bar{F}_R(r) \equiv P(R > r) = 1 - F_R(r) = \exp\left[-r^2/(2\sigma^2)\right]. \tag{15}$$

The function $\bar{F}_R(r)$ is known as the complementary CDF (CCDF). Alternatively, the CCDF is called the exceedance probability, the tail distribution, or the survival function. The inverse CCDF, called the complementary quantile function, maps a given exceedance probability, q, to an amplitude threshold. We denote the complementary quantile function as $\bar{Q}_R(q) \equiv \bar{F}_R^{-1}(q)$. Now, $q = \bar{F}_R(r) = 1 - F_R(r)$ implies that $F_R(r) = 1 - q$, and hence, $\bar{F}_R^{-1}(q) = r = F_R^{-1}(1-q)$. In other words, the complementary quantile function and quantile function are related as follows

$$\bar{Q}_R(q) = Q_R(1-q) \qquad 0 \leq q \leq 1. \tag{16}$$

When $R \sim \text{Rayleigh}(\sigma)$, (14) and (16) yield

$$\bar{Q}_R(q) = \sqrt{-2\sigma^2 \ln q} \qquad 0 \leq q \leq 1. \tag{17}$$

A.3. Derivation of the APD Graph

A measured APD can be presented with a CCDF graph, i.e., exceedance probability plotted versus threshold. When plotted on a logarithmic scale, CCDF plots are a useful tool for visualizing the upper tail of the distribution and can be used to read percentiles directly; for examples see [14–17]. A drawback of conventional CCDF plots is that it is not straightforward to assess whether the observed CCDF matches a standard type of distribution, such as Rayleigh. The APD graph introduced in [18] addresses this drawback by transforming the plot axes so that a Rayleigh distribution is a straight line. Below, we derive the APD graph in a general manner to encompass different conventions present in the literature.

Let $Y = A\log_a(R)$ be the amplitude expressed on an arbitrary logarithmic scale, which is decibels when $A = 20$ and $a = 10$, and Nepers when $A = 1$ and $a = e \approx 2.71828$. Defining $g(x) = A\log_a(x)$, the CDF transformation theorem [36, Thm. 2.1.3, p. 51] implies that the CDF of $Y = g(R)$ is

$$F_Y(y) = F_R(g^{-1}(y)) \tag{18}$$
$$= F_R(a^{y/A}) \tag{19}$$

and the corresponding quantile function for $0 \le p \le 1$ is

$$Q_Y(p) = g(Q_R(p)) \tag{20}$$
$$= A\log_a(Q_R(p)). \tag{21}$$

Applying (16), which holds for any continuous random variable, and (20), we see that

$$\bar{Q}_Y(q) = Q_Y(1-q) \tag{22}$$
$$= g(Q_R(1-q)) \tag{23}$$
$$= g(\bar{Q}_R(q)) \tag{24}$$
$$= A\log_a(\bar{Q}_R(q)). \tag{25}$$

When $R \sim \text{Rayleigh}(\sigma)$, (17) and (25) give

$$\bar{Q}_Y(q) = A\log_a[(-2\sigma^2\ln q)^{1/2}] \tag{26}$$
$$= \frac{A}{2}\left[\log_a(-\ln q) + \log_a(2\sigma^2)\right]. \tag{27}$$

For particular choices of A, a, and m, the APD graph defines the abscissa (x or horizontal coordinate) and ordinate (y or vertical coordinate) as

$$\boxed{x = \frac{A}{2m}\log_a(-\ln q) \qquad y = \bar{Q}_Y(q).} \tag{28}$$

Hence, when the amplitude, R, follows a Rayleigh distribution, the APD graph is a straight line of the form $y = mx + c$ with slope m and y-intercept $c = (A/2)\log_a(2\sigma^2)$.

Two conventions are common for the choice of the Rayleigh slope, m. Namely, many authors take $m = -1/2$, e.g., [18, 20–25, 32], while others take $m = -1$, e.g., [28–30, 33, 34]. Thus, when presenting APD graphs, it is important to specify which convention is used for the slope, m, as well as the logarithmic scale for the amplitude specified by A and a.

Scaling the amplitude in decibels, i.e., $A = 20$ and $a = 10$, and taking the Rayleigh slope to be $m = -1$ we obtain the following coordinates for the APD graph

$$\boxed{x = -10\log_{10}(-\ln q) \qquad y = \bar{Q}_Y(q).} \tag{29}$$

In this case, $c = 10\log_{10}(2\sigma^2)$ for a Rayleigh distribution. We use the coordinates in (29) for the APD plots below. Example Python code for making an APD graph from amplitude samples is provided below in Sec. A.6.

A.4. Examples

The APD plot is essentially a graph of the complementary quantile function for the envelope amplitude, scaled in decibels, plotted versus a nonlinear transformation of the exceedance probability. Fig. 42 shows an example APD plot for a mixture of two Rayleigh distributions, where the upper component occurs with probability 0.3 and the lower component occurs with probability 0.7. The abscissa (x-axis) is labeled with the exceedance probability, q, and the ordinate (y-axis) is the envelope amplitude in decibels.

Quantiles (percentiles) can be read off an APD plot by recalling from (22) that the complementary quantile function evaluated at $0 \leq q \leq 1$ is equal to the quantile function evaluated at $p = 1 - q$. For example, exceedance probabilities of 50% and 90% correspond to the median and 10th percentile, respectively.

As expected, the APD plot in Fig. 42 is linear in regions dominated by each Rayleigh component. Specifically, the APD is a straight line beyond the 30% exceedance probability, which corresponds to the mixture weight of the upper component.

A more involved example with mixtures of three components is given in Fig. 43. The lower mode is a Rayleigh distribution and the upper modes are log-normal distributions, which is a common model for large-scale fading attributed to shadowing effects in radio channels [37, p. 95]. Three sets of mixture probabilities were used: $[0.5, 0.4, 0.1]$, $[0.5, 0.3, 0.2]$, and $[0.5, 0.1, 0.4]$, denoted as Mixture 1, Mixture 2, and Mixture 3, respectively. As expected, the APD is a straight line for exceedance probabilities above 50%, which corresponds to the Rayleigh component. By contrast the log-normal components, which occur for exceedance probabilities below 50%, are not straight lines. Note that the mixture weight for the upper component (probabilities of 0.1, 0.2 and 0.4 for Mixtures 1, 2, and 3) corresponds to the steep drop visible at an exceedance probability of 0.1, 0.2 and 0.4, respectively.

It is common to normalize the envelope amplitude by a reference value, such as the average, median, or thermal noise floor on the ordinate of the APD plot. Two different amplitude normalizations are illustrated in Fig. 44, which shows the APDs for various Rice distributions. Here, the Rice distribution, a common model for small-scale fading with a line-of-sight (LOS) component, is graphed with different K factors, where K is the ratio of power in the LOS component to the power in the diffuse component [37, p. 86]. When $K = 0$, the Rice distribution reduces to a Rayleigh distribution, and as expected, the APD plot is a straight line. For this reason, APDs have been used to assess the presence of a LOS component in radio frequency measurements, e.g., see [30].

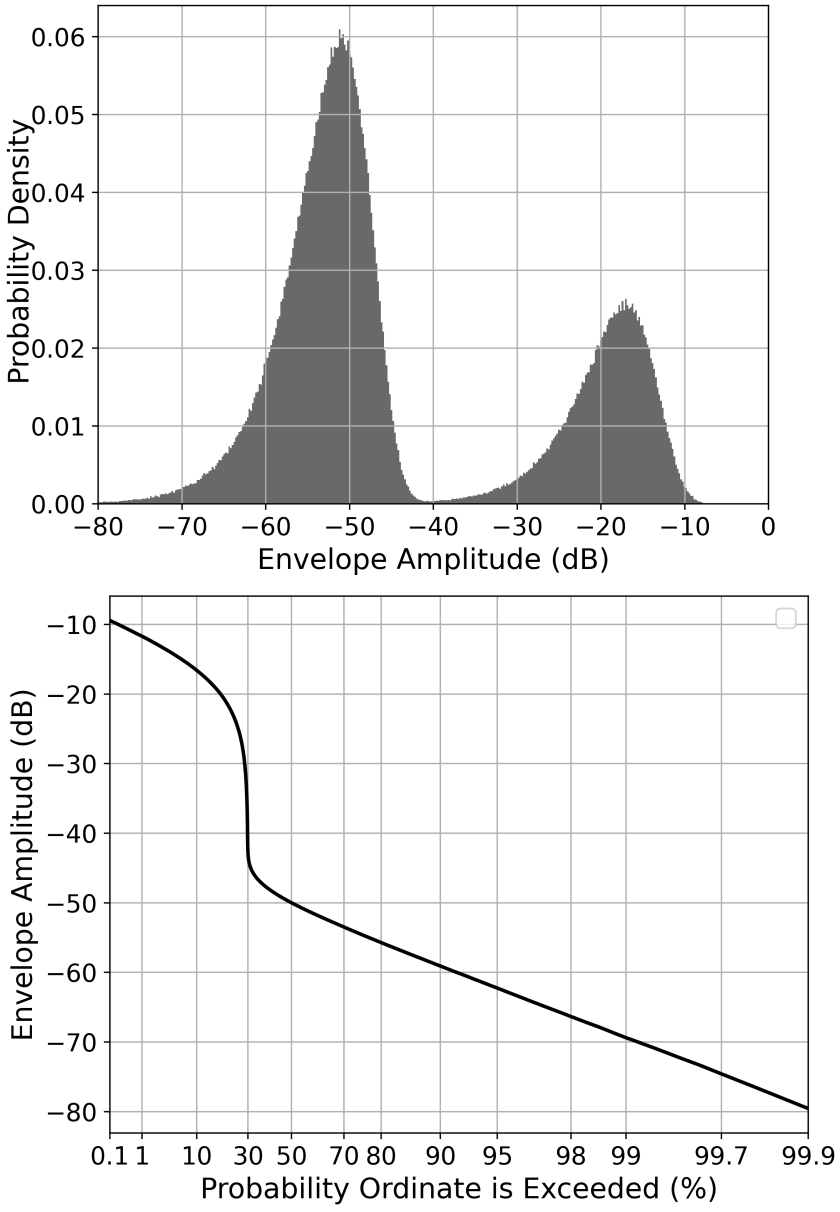

Fig. 42. Example with a mixture of two Rayleigh distributions, where the probability weight of the upper component is 0.3. Top: Histogram. Bottom: APD plot.

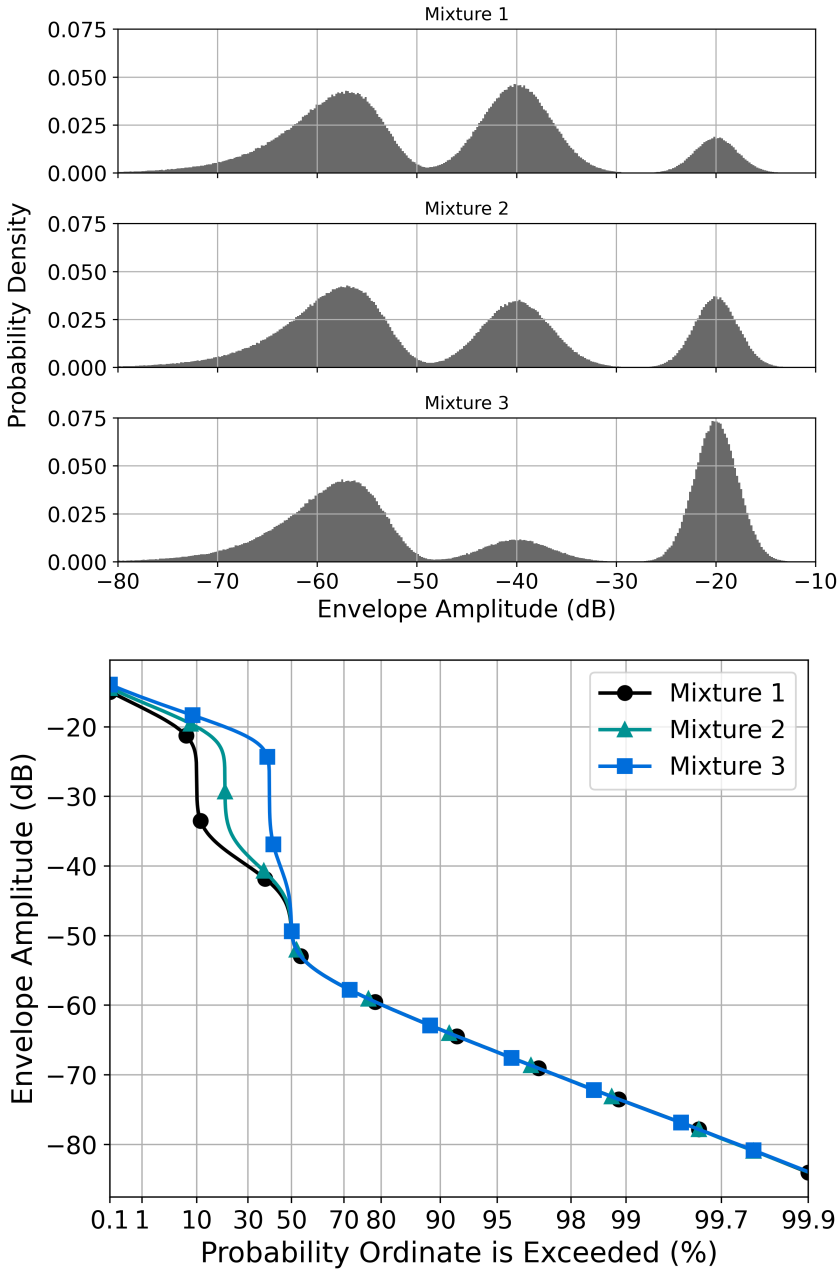

Fig. 43. Examples with mixture distributions consisting of three components, where the lower component follows a Rayleigh distribution and the upper components are log-normal distributions. Top: Histograms. Bottom: APD plots.

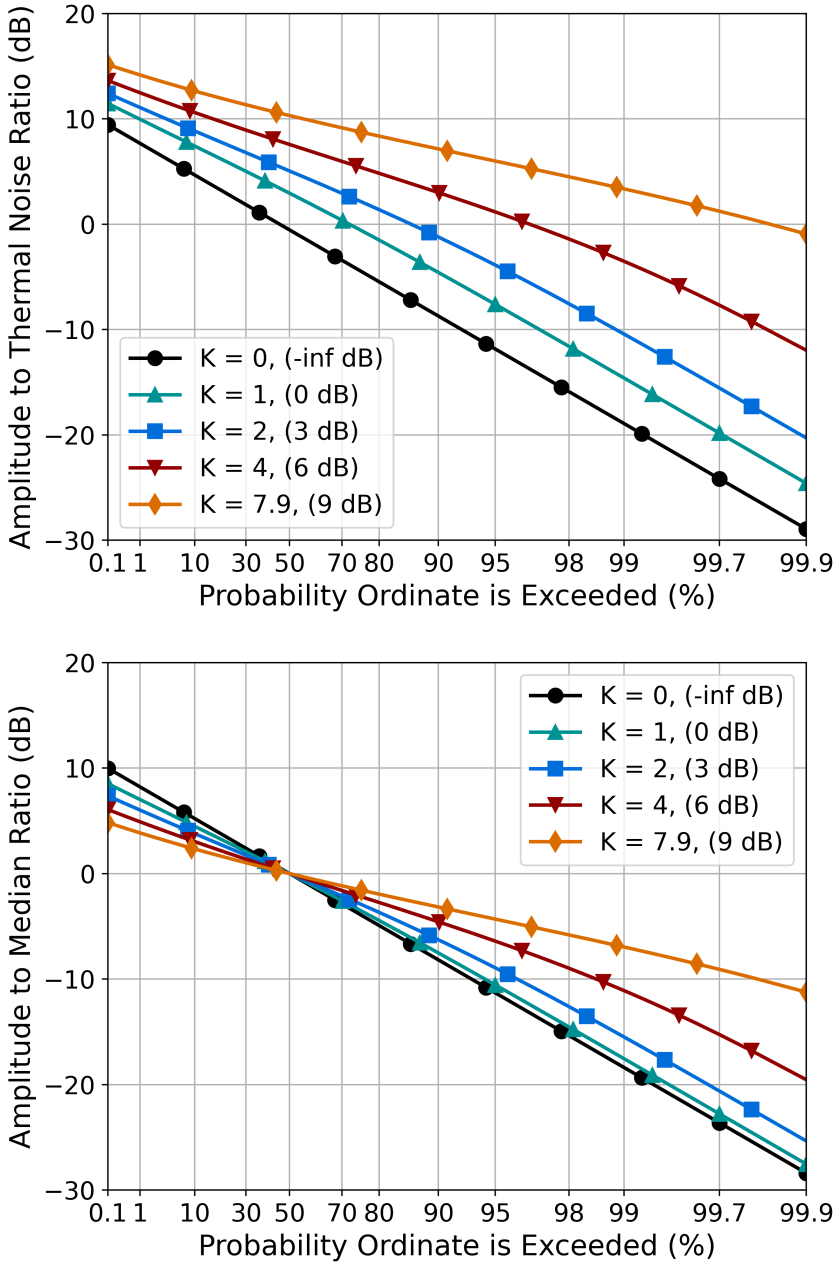

Fig. 44. Example APD plots for Rice distributed amplitudes with various K factors. Top: Thermal noise floor normalization. Bottom: Median normalization.

A.5. Discussion

The APD graph is valuable for identifying a Rayleigh-distributed component of the envelope amplitude, since such a component appears as a straight line. However, APD plots also have some drawbacks. First, because the abscissa on an APD plot is a complicated function of the exceedance probability, APD plots can be challenging to interpret for non-Rayleigh distributed amplitudes. Second, because most of the APD graph space is devoted to the lower quantiles of the amplitude distribution, i.e., the lower tail, the APD graph is not necessarily the best visualization to assess the upper tail of the distribution. Finally, the limited use of APD graphs means that few practitioners are familiar with their interpretation.

A.6. Example Python Code for an APD Graph

```
import numpy as np
import matplotlib.pyplot as plt

def apd_plot_from_samples(R, fig=None, ax=None, label=None):
    '''
    Parameters
    ----------
    R : 1-D array
        ampltiude samples in linear units (e.g., voltage)
    fig : matplotlib figure object (optional)
    ax : matplotlib axis object (optional)
    label : string
            plot label

    Returns
    -------
    fig :
        figure object
    ax :
        axis object
    '''
    Y = 20*np.log10(R) # amplitude in decibels
    q = np.linspace(0.1, 99.9, 1000)/100 # complementary quantiles to plot
    x = -10*np.log10(-np.log(q))
    y = np.quantile(Y, 1-q)

    if fig == None:
        fig, ax = plt.subplots(1,1,figsize = (8,6))
```

```
    ax.plot(x, y, linewidth=2, label=label)
    ax.set_xlabel('Probability Ordinate is Exceeded (%)', fontsize = 16)
    ax.set_ylabel('Envelope Amplitude (dB)', fontsize = 16)
    qticks = np.array([0.1, 1, 10, 30, 50, 70, 80, 90, 95, 98, 99, 99.7, 99.9])/100
    xticks = -10*np.log10(-np.log(qticks))
    xtick_labels = ['0.1', '1', '10', '30', '50', '70',
                    '80', '90', '95', '98', '99', '99.7', '99.9']
    ax.set_xticks(xticks, xtick_labels)
    ax.set_xlim(x[0],x[-1])
    ax.grid(True)
    ax.tick_params(axis='both', which='major', labelsize=14)
    ax.legend(fontsize = 14)
    return fig, ax
```