Columbia Integrated Systems Laboratory

ASPEN
CONSULTING GROUP INC

Peraton LABS

High-Speed Sensing of the Electromagnetic Environment for Cognitive Radio Receivers

Presenters: Matt Bajor, Ron Li

Outline

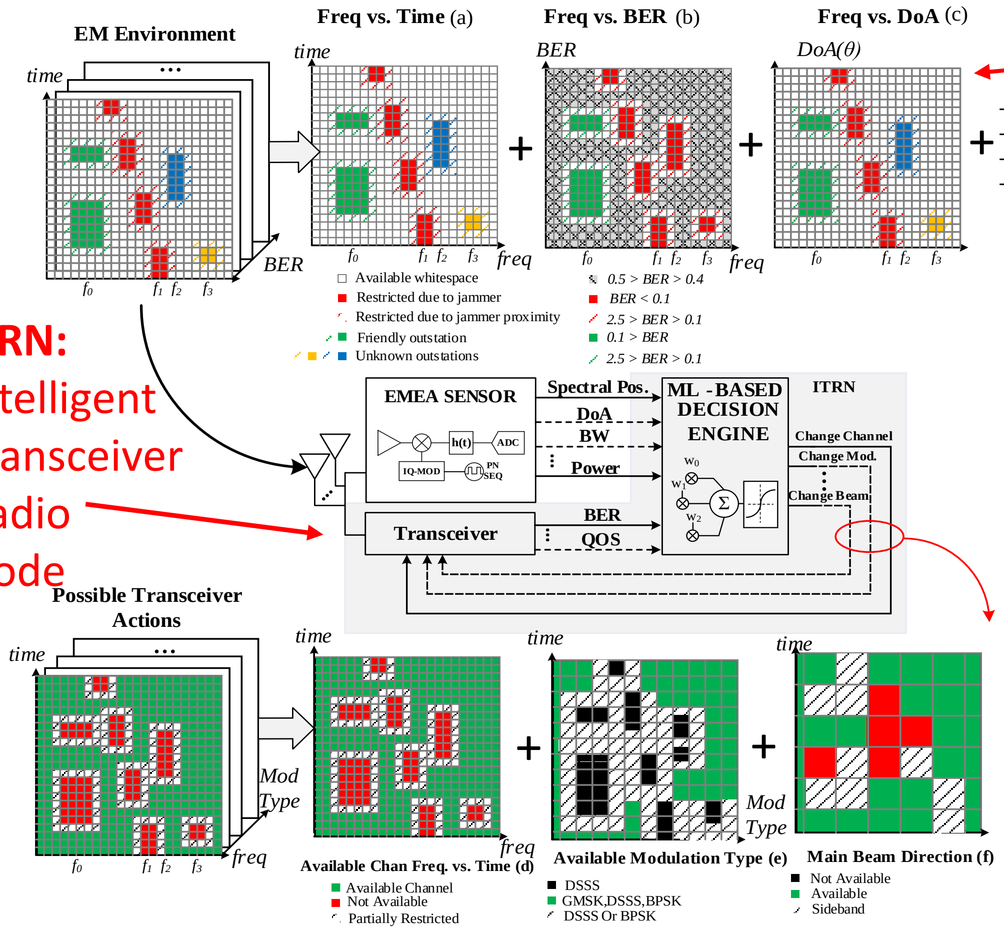
- Introduction
- Background and Motivation
  - High-Level System Diagram
- Problem Solution
- Electromagnetic Environment Aware (EMEA) Sensor
  - Standalone Performance
- Hardware Diagram
  - Modifications to the FPGA
  - Compressed Sensing Implementation in GNU Radio
  - MLBDE Input BER Cost Function Circuit
- Machine Learning-Based Decision Engine (MLBDE) Overview
- Multi-Armed Bandit (MAB) ML Problem Formulation
- MLBDE Proof-of-Concept Validation Results
- References

ASPEN CONSULTING GROUP INC

Introduction

- The EM environment is cluttered in multiple domains (frequency, time, angle, etc.) making reception with an outstation increasingly difficult.
- Current detection methods for finding available receiver whitespace do not scale well in terms of speed and energy consumption.
- We present a receiver architecture that can be used for sensing an emitter's spectral location in a fraction of the time and energy as the current state of the art.

Background and Motivation



- Modulation
- Internal Characteristics
- Training Sequences
- etc ...

ITRN: Intelligent Transceiver Radio Node

Figure 1: N-dimensional "Resource Cube"

"Resource cubes" contain information of the EM environment

Radio Inputs

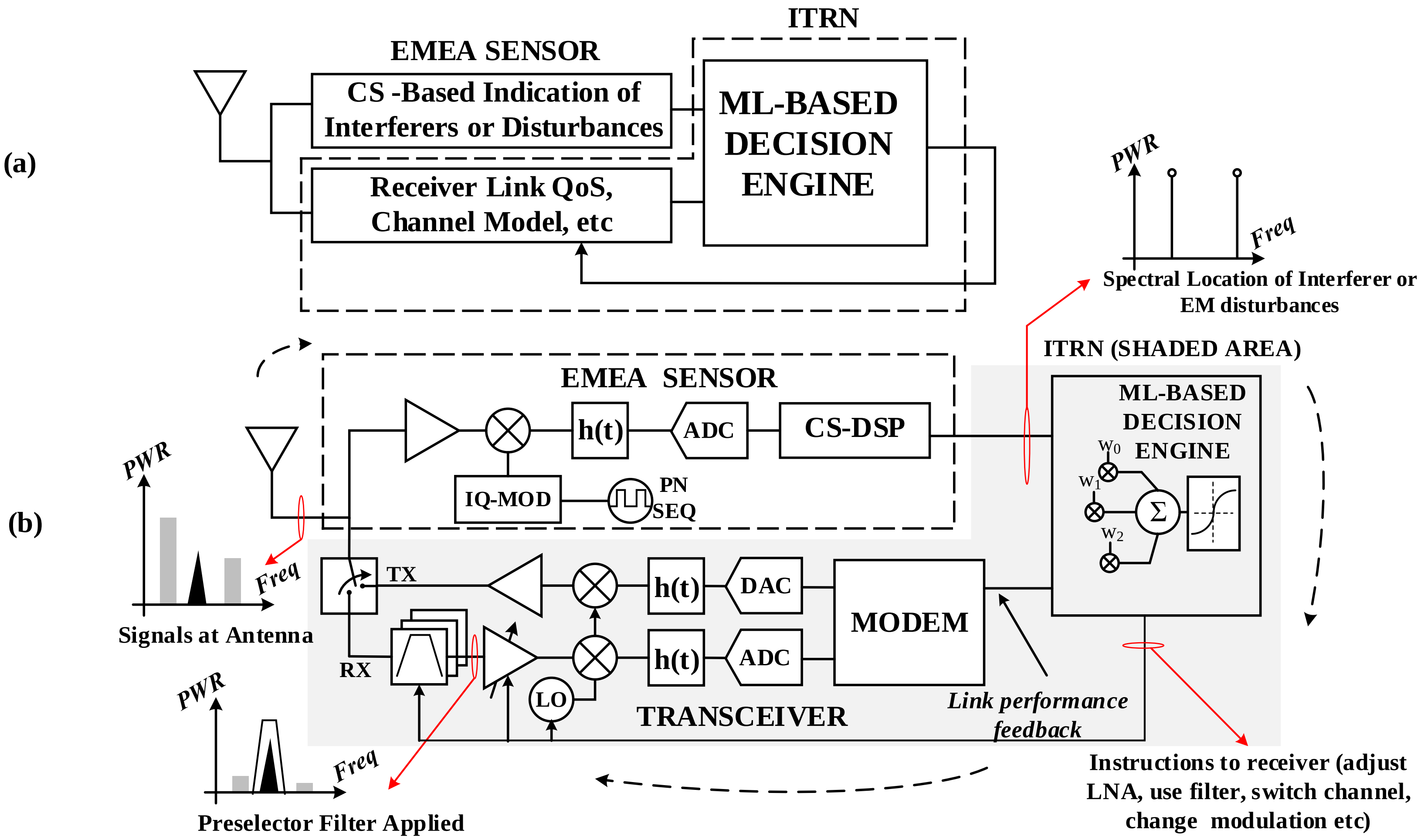
Problem 1: The more data available, the harder it is to rapidly exploit (sensing bottleneck).

Radio Outputs

Problem 2: Optimal decision situational dependent (decision bottleneck).

Strategy to Mitigate Bottlenecks: Combine Compressed (CS) Sensing with Machine Learning (ML)

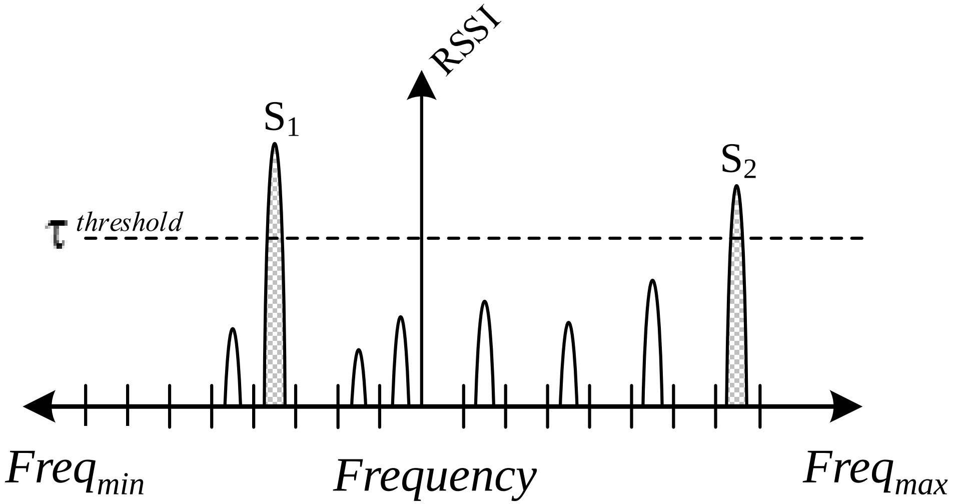
High Level System Diagram



Problem Solution

Proposed Solution (and Assumptions about the EM Environment):

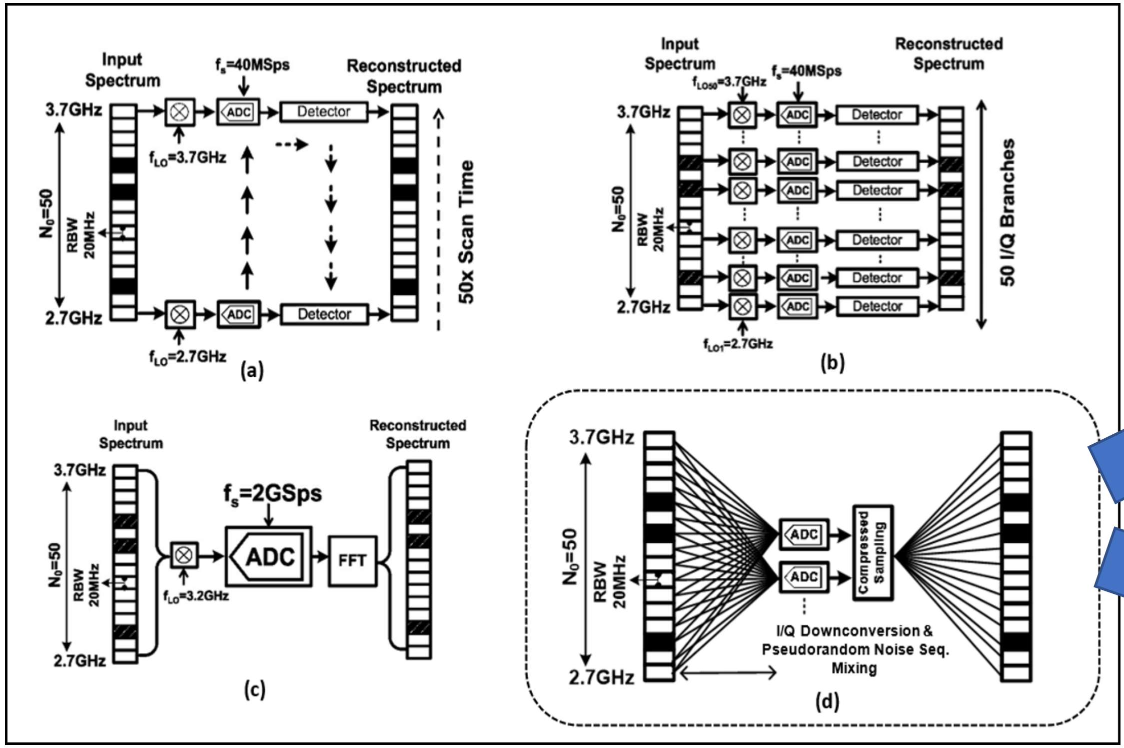
- **Compressed Sensing (CS):** By assuming sparsity (that there are much less jammers than possible signal locations (K<<N)), compressed sensing can be used to drastically reduce the time it takes to perform a spectrum scan. It is also much more scalable than current sensing methods.



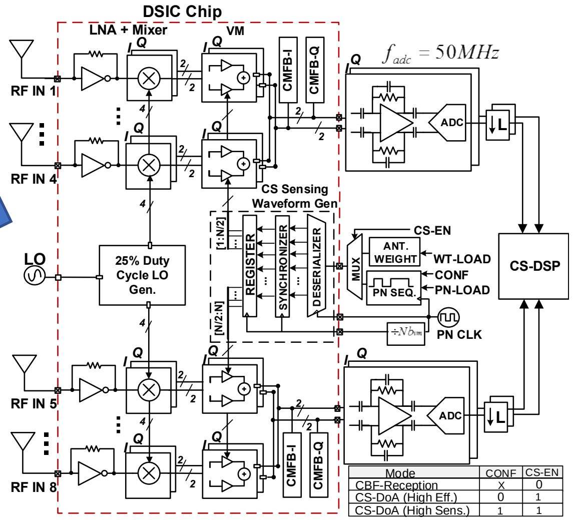
Reinforcement Learning:

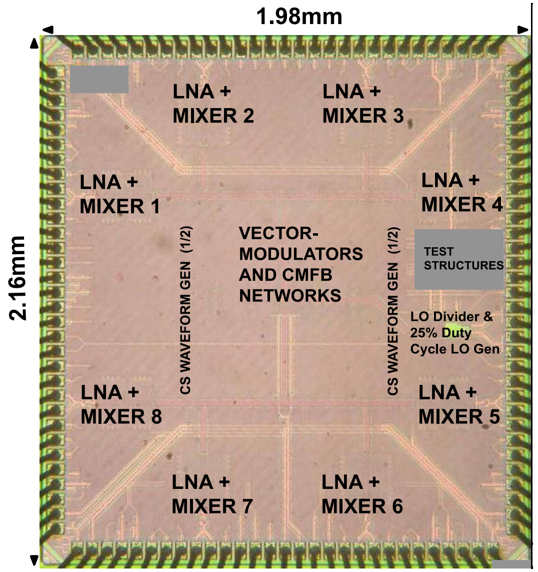
- Using a ML based decision engine, we can make adaptive countermeasure decisions much more effectively than a lookup table (LUT) based approach. It is also more scalable than a complicated logic circuit or LUT.

Electromagnetic Environment Aware (EMEA) Sensor



- **A fully custom, CS enabled RF-ASIC with a single HW branch is used, called the DSIC (Direct Space to Information Converter) [4].**
- **The DSIC uses LO modulation to sense the frequency spectrum on 1 or 8 antennas.**



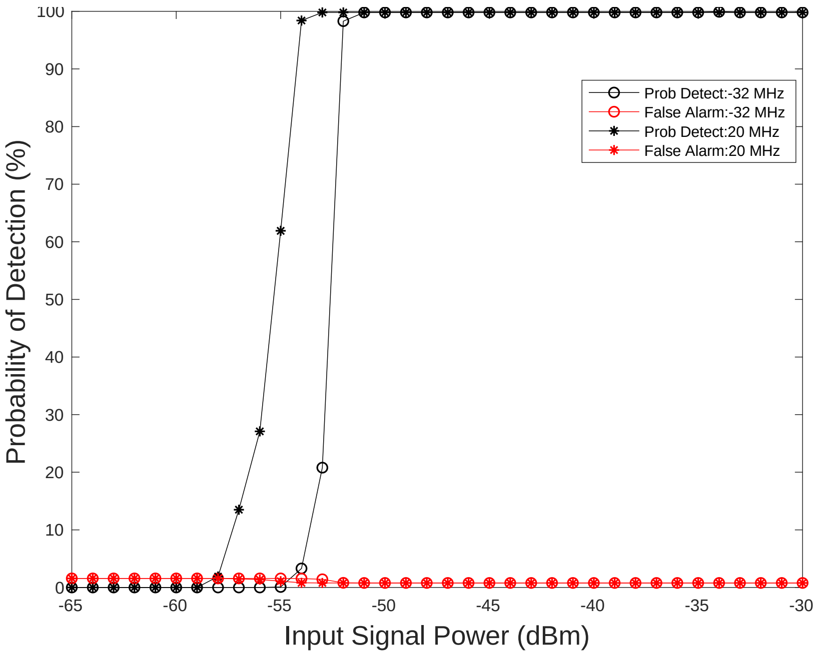


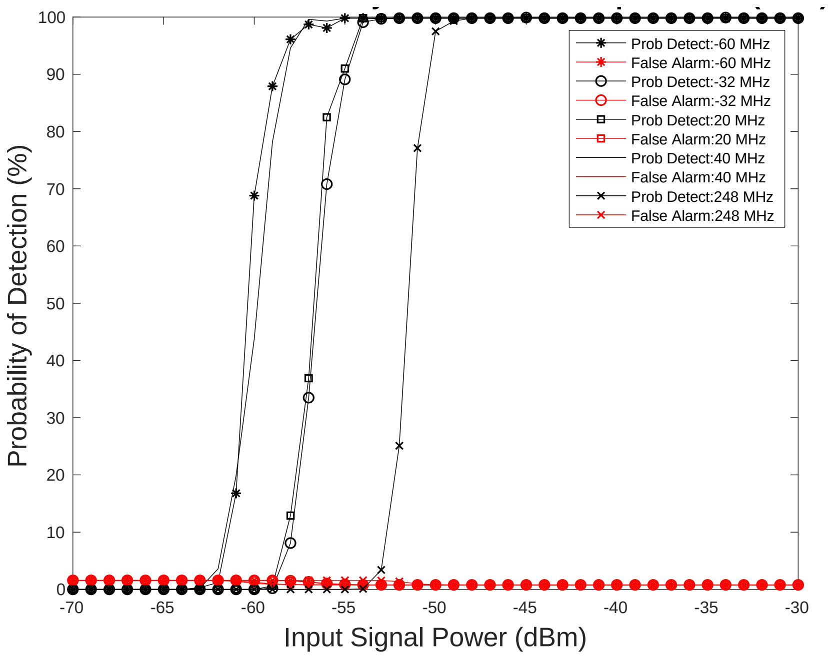
- All spectrum sensing methodologies are ultimately limited by the Nyquist rate e.g. N measurements for N possible signal locations.
- By exploiting sparsity in the spectrum, CS can be used to take as few as m measurements where $m << N$ [5].

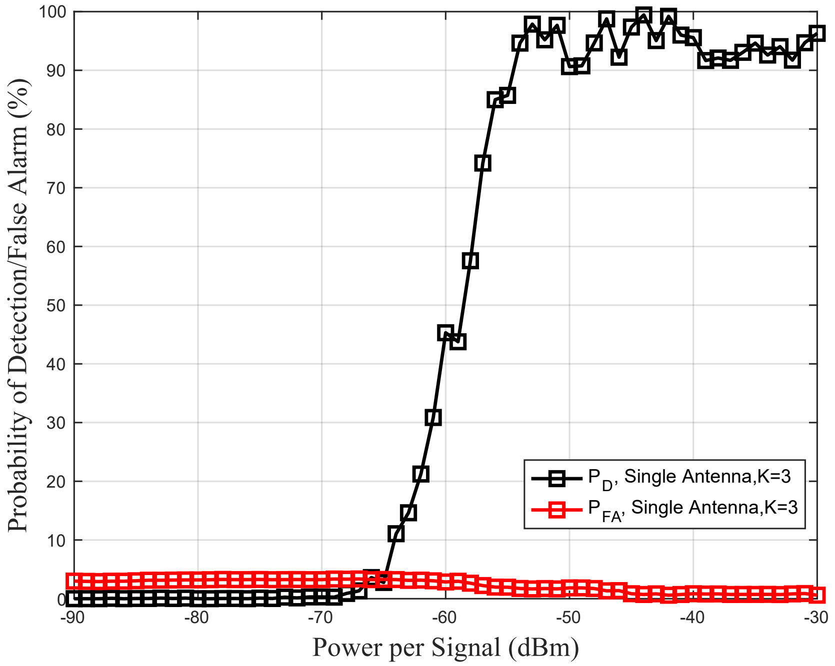
$$m = KC_o \log(N/K)$$

Won 3rd place at RFIC 2018!

Standalone Performance of the EMEA Sensor







Each Point for K=1:

- 10 samples per experiment
- 1000 experiments
- 9 measurements (1 PN seq. → 9 by branch expansion)

OMP Iterations = 2

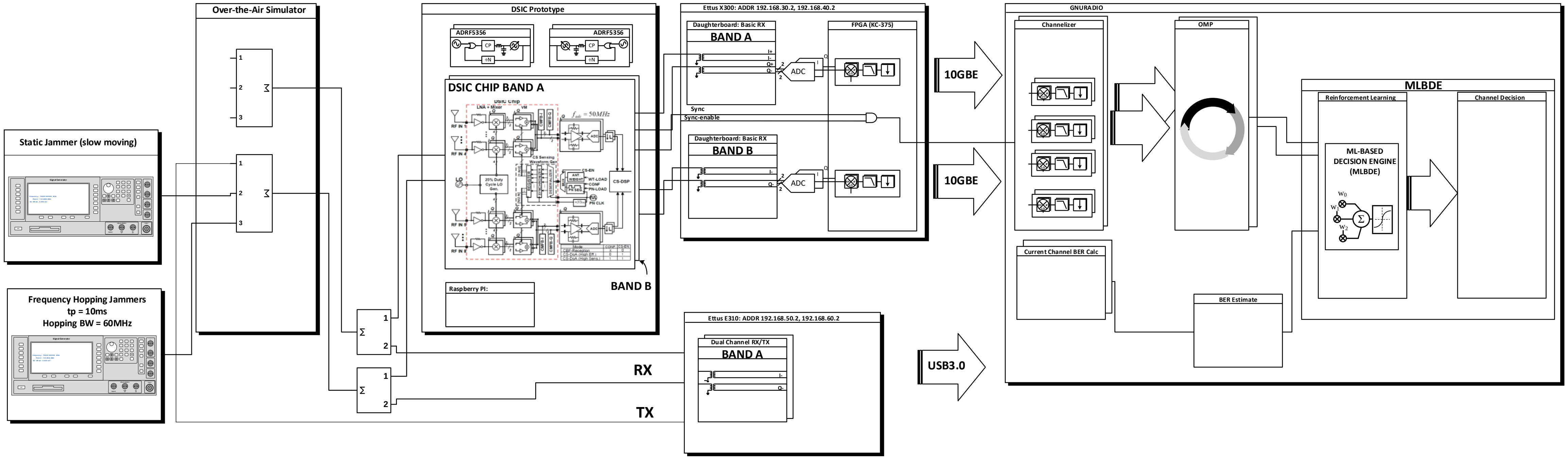
Each Point for K=3:

- 10 samples per experiment
- 1000 experiments
- 19 measurements (1 PN seq. → 19 by branch expansion)

OMP Iterations = 4

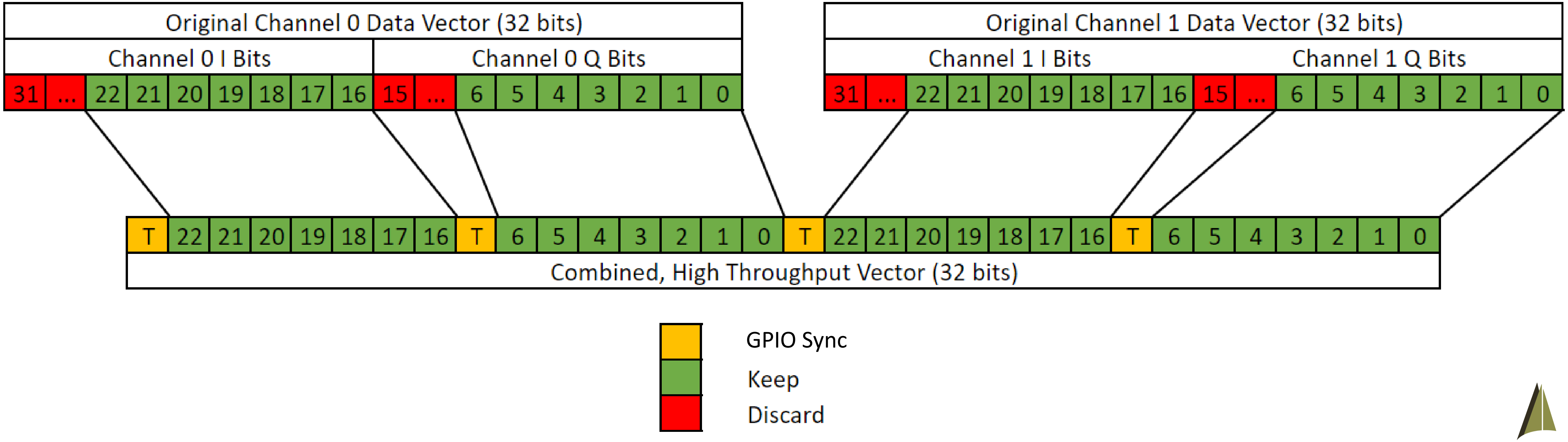
ASPEN
CONSULTING GROUP INC

Hardware Diagram



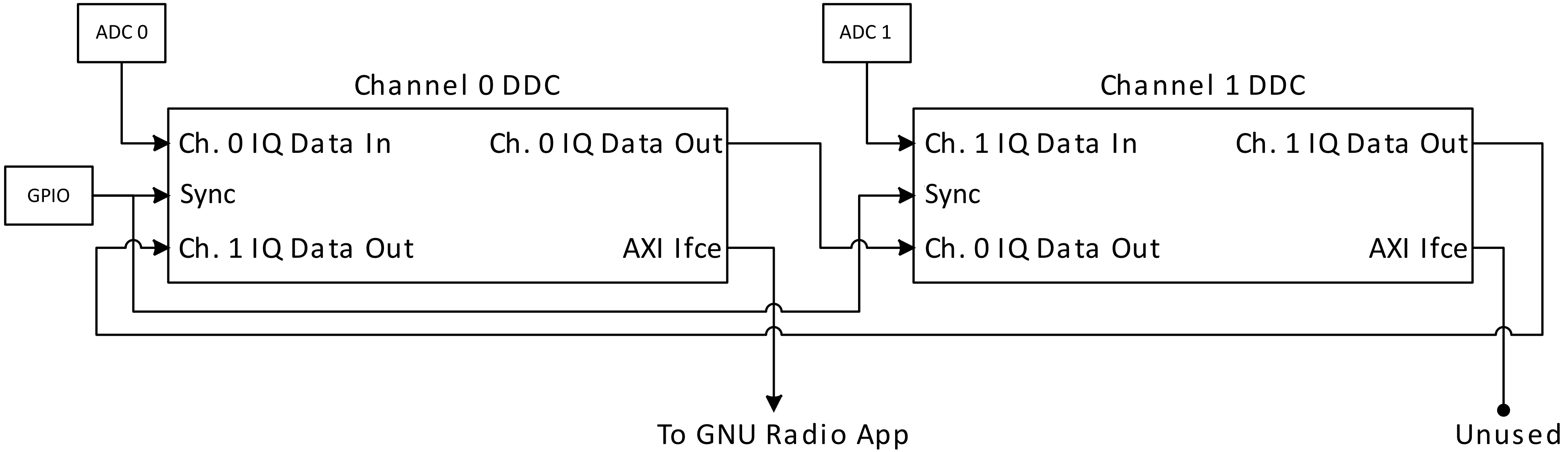
FPGA Modification Goals

- Embed sync input from GPIO port into IQ data to use as a time reference
- Reformat IQ data vector to include both channels' IQ data and sync signal in 32 bits to increase throughput to GNU Radio application



FGPA Modification Overview

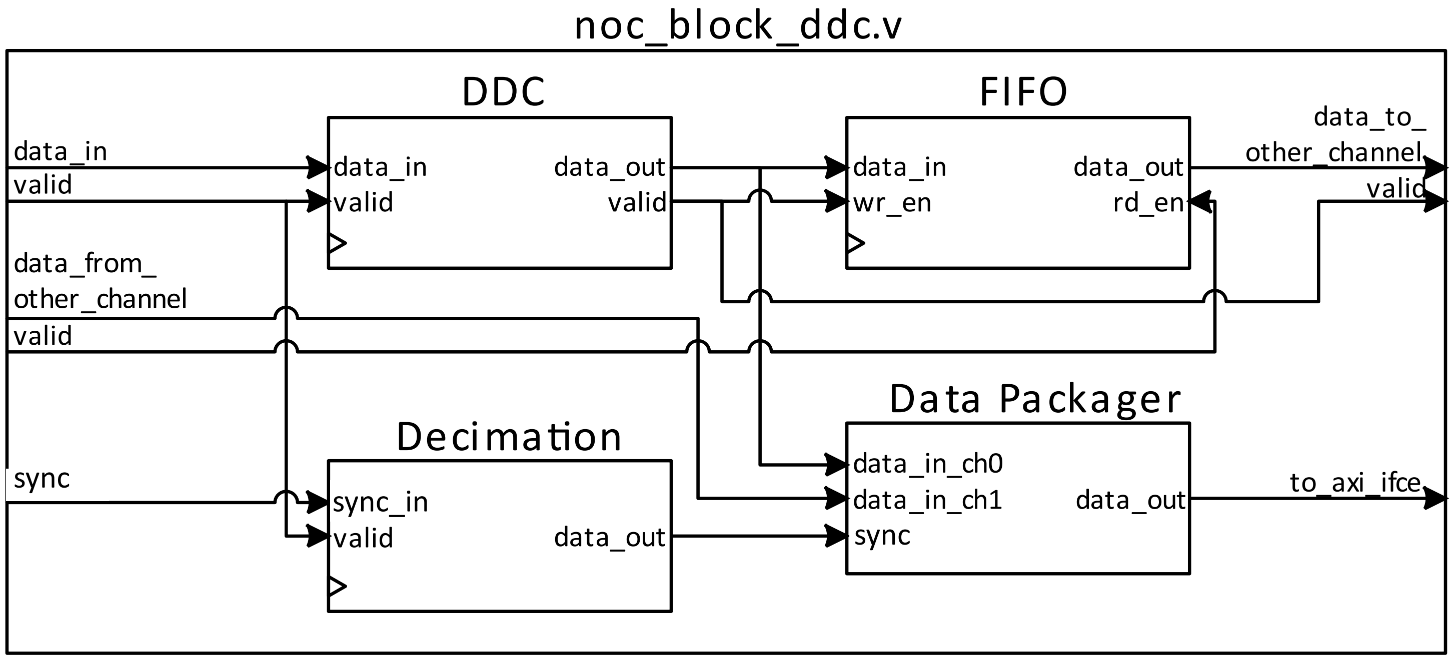
- Adapted from UHD 3.15 official release
  - Most changes were made in noc_block_ddc.v
- Channel 0 and 1 DDCs (noc_block_ddc) will share baseband IQ samples and repackage the IQ data and sync.



Detailed FPGA Modifications

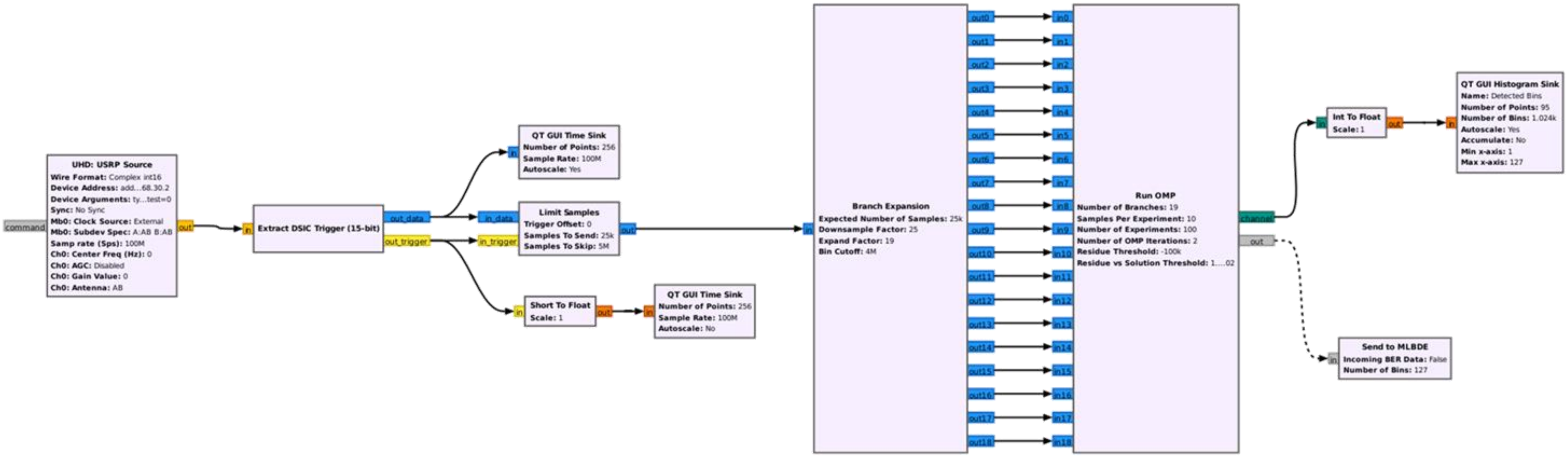
Decimation, FIFO and Data Packager are new components

- **Decimation**
  - Decimates sync signal at the same rate as the DDC and buffers the decimated sync with a FIFO to preserve time alignment with IQ data
- **FIFO**
  - Buffers baseband IQ data to be sent to the other channel's noc_block_ddc instantiation
  - Data is read out of the FIFO when the other channel's data is valid, ensuring the packaged data is valid on both channels
- **Data Packager**
  - Packages IQ data from both channels and the sync into a single 32 bit vector

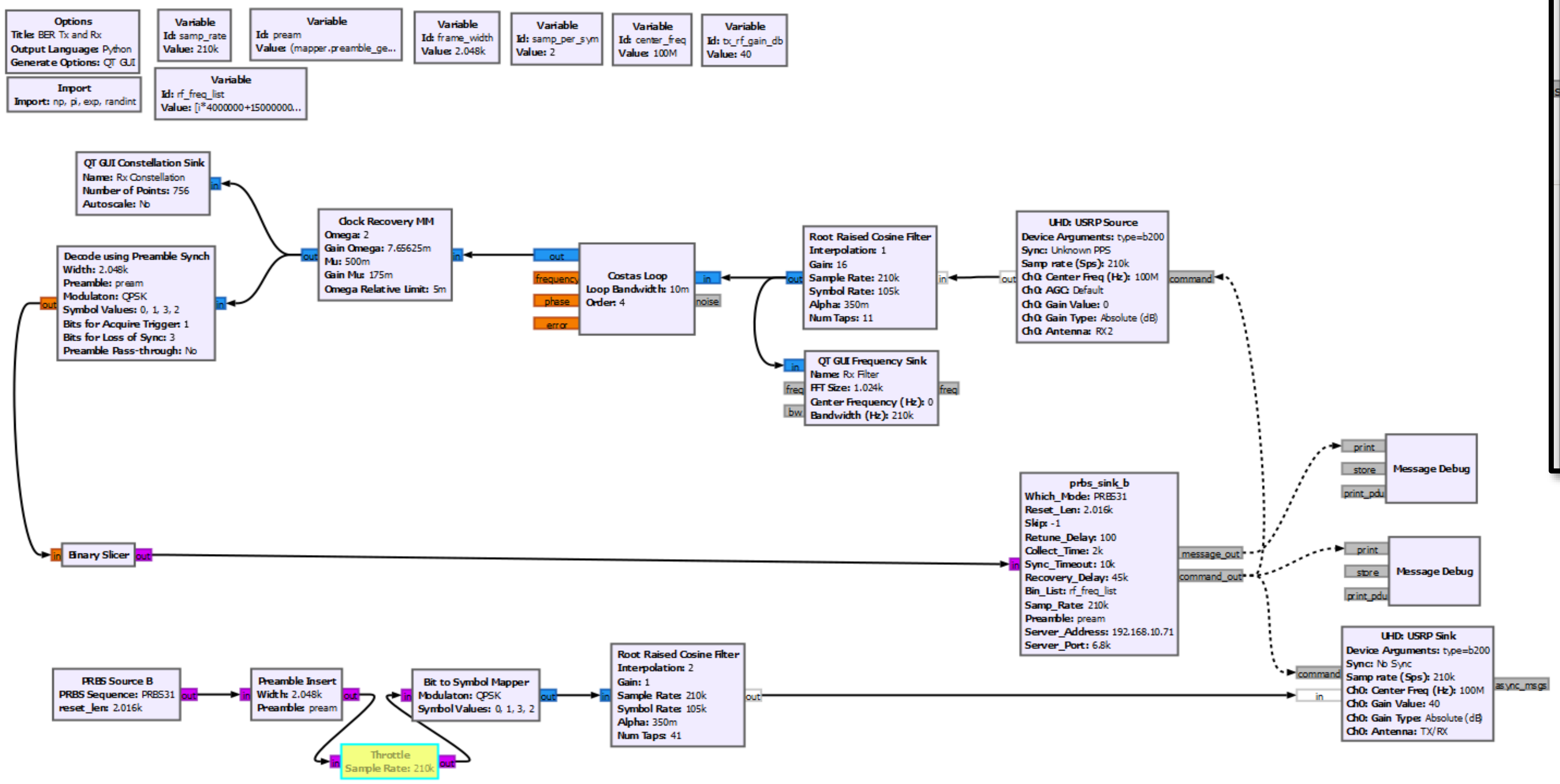


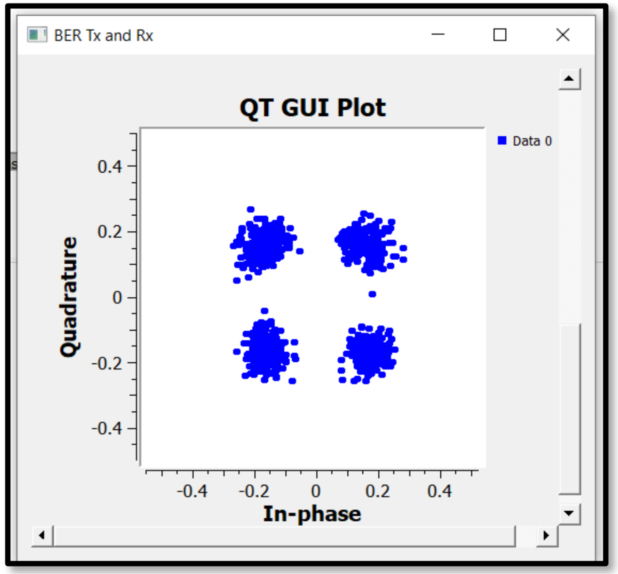
Compressed Sensing Implementation

- Digital branch expansion is used in GNU Radio to create *m* virtual branches
- Each *virtual branch* corresponds to a CS measurement in the frequency domain: $m = KC_o \log(N/K)$
- Orthogonal Matching Pursuit (OMP) is used to recover the supports (e.g., Signals) -no signal reconstruction is required



MLBDE Input BER Cost Function Circuit

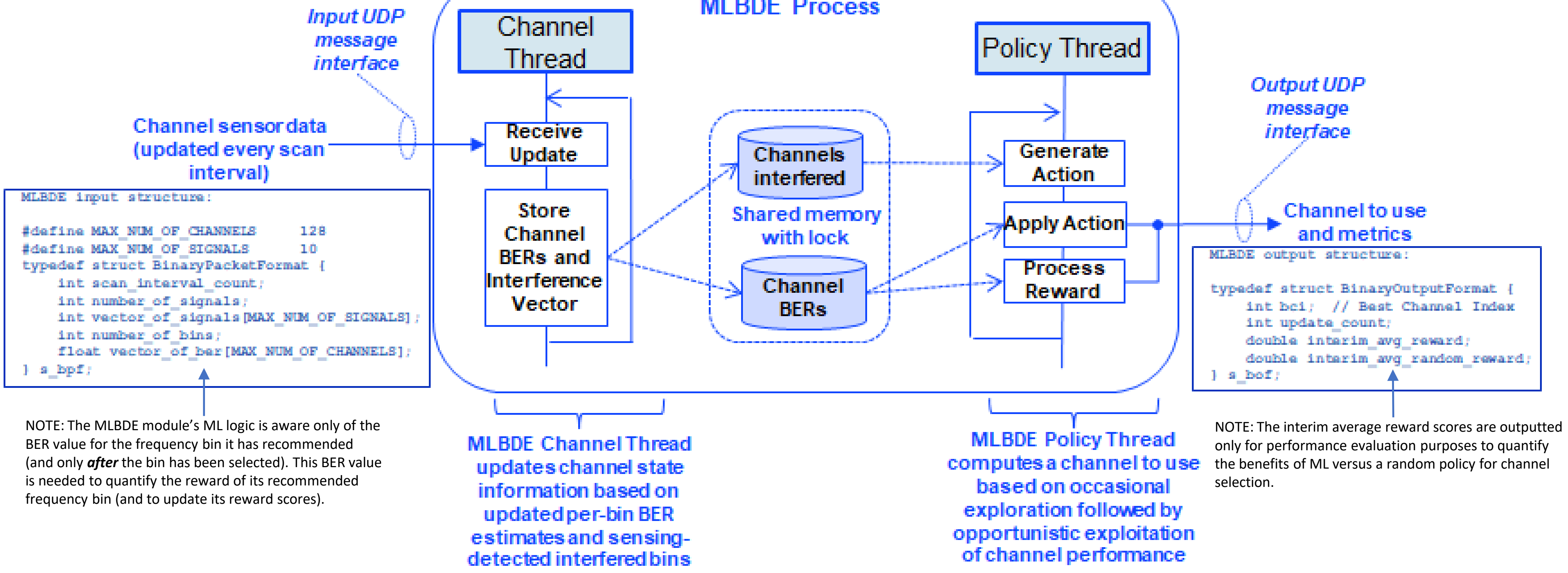




Machine Learning-Based Decision Engine (MLBDE) Overview

- The goal of the ITRN system's Machine Learning-Based Decision Engine (MLBDE) module is to compute in real-time an optimal recommended frequency bin that is both robust and high-performing
- The MLBDE module processes spectrum sensing data streamed to it in real-time over a UDP socket interface that contains the set of frequency bins that are interfered (and estimates of channel BER values used to "score" the MLBDE recommended frequency bin)

MLBDE Software Architecture:



ASPEN CONSULTING GROUP INC

Multi-Armed Bandit (MAB) ML Problem Formulation

- The MLBDE casts the ITRN frequency bin selection challenge as an instance of the contextual multi-armed bandit (MAB) problem
- The reward obtained by the action (A_t) of selecting frequency bin $A_t = k \in \{1,2,\ldots,K\}$ at each step t is the complement of BER:
  - I.e. $R_t(A_t = k) = 1 - q_t(k)$ where $q_t(k)$ is the BER for frequency bin k over the interval associated with time step t
  - The MLBDE performance metric evaluated over N steps is the average step-by-step reward ($\bar{R}_N$):
    - $\bar{R}_N = \frac{1}{N}\sum_{t=1}^{N} R_t = \frac{1}{N}\sum_{t=1}^{N}\left(1 - q_t(A_t)\right)$
  - The average reward yielded when selecting frequency bin k:
    - $\bar{R}_N(k) = \frac{\sum_{t=1}^{N} R_t \cdot \mathbf{1}_{A_t=k}}{\sum_{t=1}^{N} \mathbf{1}_{A_t=k}}$
- Defining S_t as the set of frequency bins experiencing interference and $s_N(k)$ as the fraction of steps (out of the N steps $\{1,2,\ldots,N\}$) for which frequency bin k was interfered, the ML decision logic selects a high-performing bin (A_{N+1}) according to $\bar{R}_N(k)$ **while avoiding bins in S_t:**
  - $A_{N+1} = \max_{\substack{k\in\{1,2,\ldots,K\},\\ \{k\}\cap S_{N+1}=\emptyset}} \bar{R}_N(k) - s_N(k)$

I.e. the ML decision logic uses the sensing-derived side information about interfered frequency bins as context

- MAB policies / heuristics implemented as part of the MLBDE policy suite to process the channel information and output a recommended frequency bin:

Policy / Heuristic	Description of Policy/Heuristic
Random	Periodically select a random frequency irrespective of the interfered frequency set or its past selections
Sticky Non-Interfered	A new frequency is not explored unless the current frequency is under interference
Random Non-Interfered	Periodically select a random frequency as long as the frequency is not under interference
ε-Greedy	Perform random exploration with probability ε (exploration) but use best frequency o/w (exploitation)
ε-First	Perform pure exploration for first εN trials and then pure greedy exploitation for remaining (1 – ε)N trials
ε-Decreasing	Similar to ε-Greedy, but uses a decreasing ε value as the experiment progresses
Epoch-Greedy	Experiment proceeds as a sequence of epochs where, in each epoch, exploration of new frequency bin(s) is pursued first followed by exploitation of the best frequency bin for the remainder of the epoch

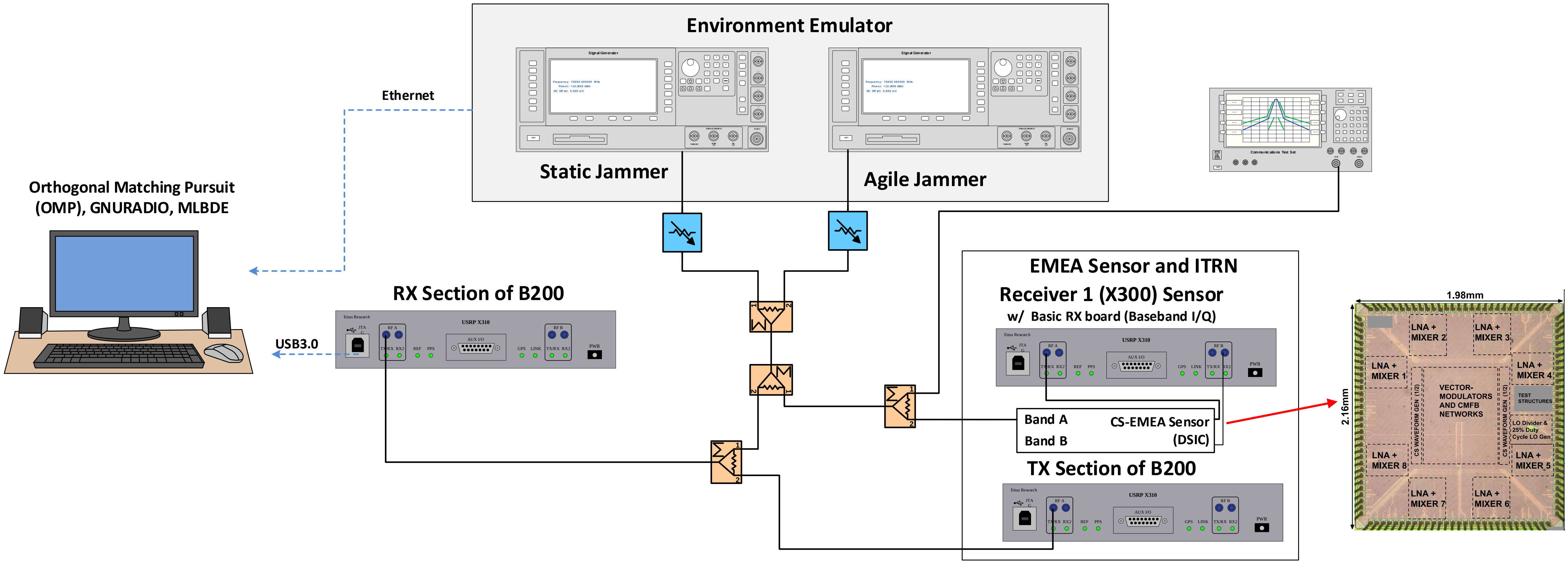
- Random policy procedures implemented primarily for comparison purposes

MLBDE Proof-of-Concept Validation Results

- The MLBDE software was validated in standalone mode using channel sensor input data collected offline and saved to file
  - MLBDE software compiled with *g++ 7.5.0* (Ubuntu 7.5.0-3ubuntu1~18.04)
  - Tested on a "modest" Ubuntu 18.04 machine (Intel(R) Core(TM)2 Duo CPU   P8700 @ 2.53GHz, 2 GB RAM)
- At experiment time, the sensor data was read from file and sent at "high speed" (e.g. ~662 Hz ←→ ~1.51 ms intervals) by a unit-test driver module to the MLBDE component via UDP socket API
  - Higher channel sensor input data rates are potentially supportable and are an area of further investigation
- ***Epoch-Greedy*** was the ML heuristic enabled in the experiments behind these results

Test Purpose	Result / Key Finding
Compare Random policy versus Greedy heuristic	ML with Epoch-Greedy heuristic in use for 2 distinct input data sets reduced avg. BER of selected frequency bin by factors of 7.76 and 26.9 versus avg. BER achieved by a baseline Random policy
Process fast channel sensor updates	For a representative experiment with 101 frequency bins and 10 interfering signals per sensor update (@ ~662 Hz), MLBDE processed sensor updates originated every ~1509 μs in real-time
Validate use of interference side information (versus not)	Epoch-Greedy ML use of interfering signal side information yielded a reduction in avg. BER by a factor of ~2 (versus w/ no side information) → Example of spectrum sensing benefit
Verify shift to robust bins by using $s_N(k)$ in decision logic	Using a compound metric ($\bar{R}_N(k) - s_N(k)$) to select next frequency bin (A_{N+1}) versus pure reward metric ($\bar{R}_N(k)$) shifts selected the bin preference to less-interfered bins by up to 74%

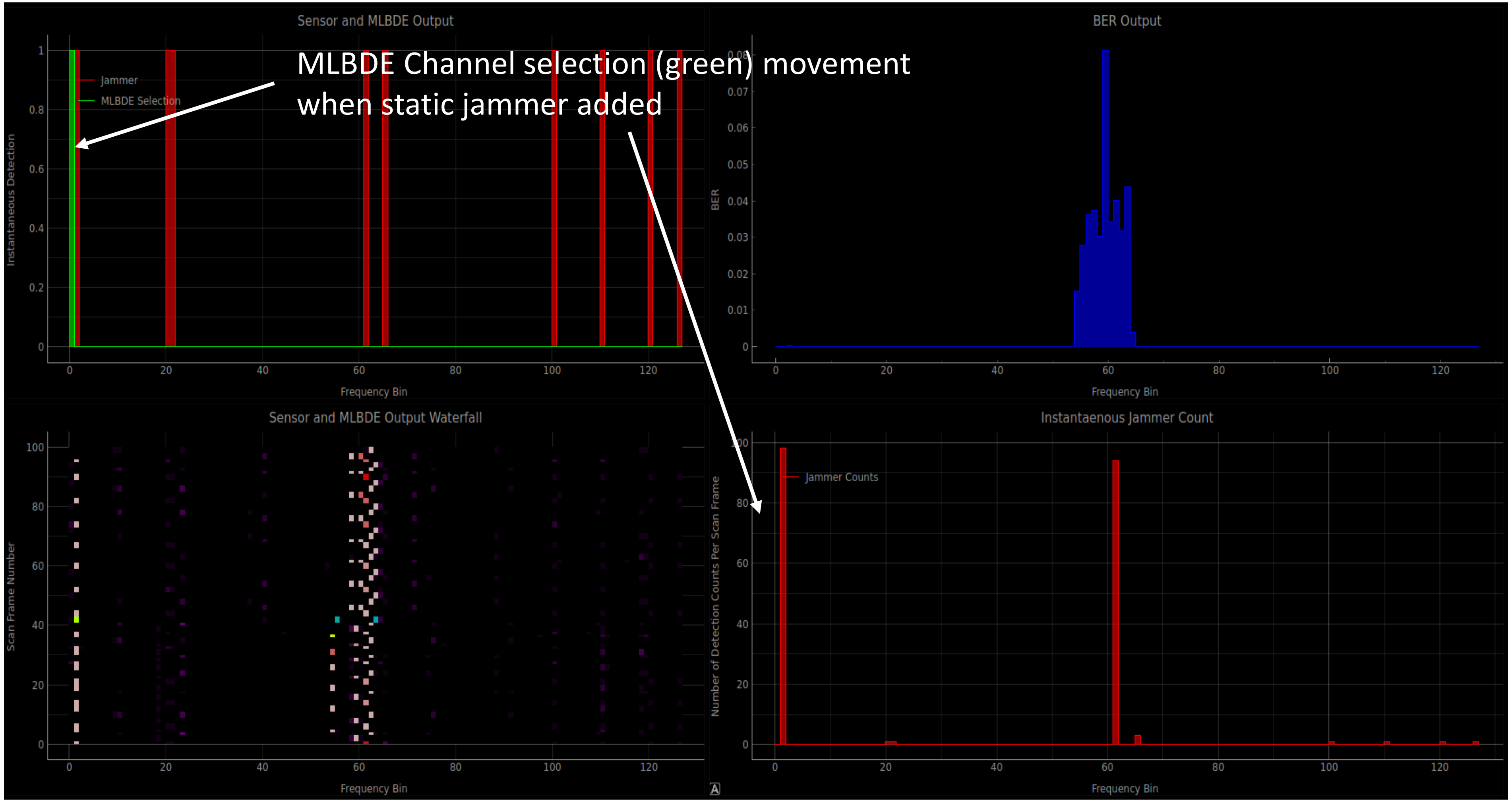
Simulated EM Environment



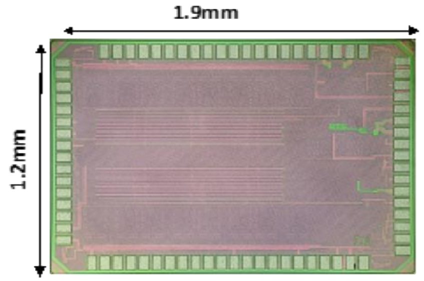
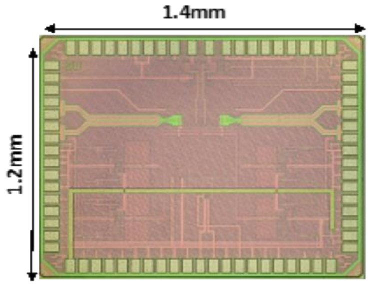
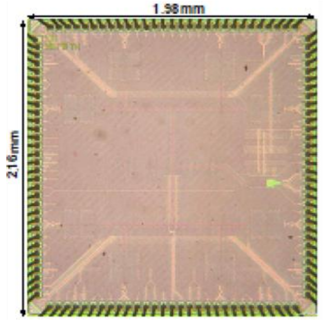
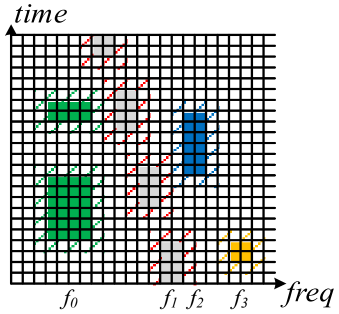
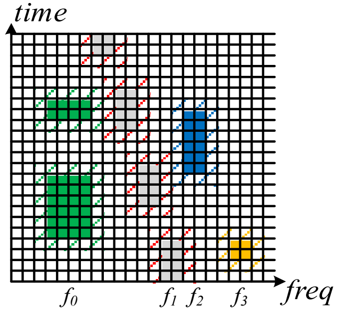
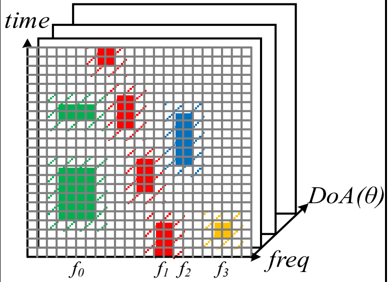
ASPEN CONSULTING GROUP INC

Demo Video

System GUI and Example Data



Future Work Sensor Work

Name	TS-QAIC	DRF2IC	DSIC
Reference	[26]	[27]	[67]
			
Year	2015	2017	2018
Process	TSMC 65nm	TSMC 65nm	TSMC 65nm
Does CS-Spectrum sensing?	Yes	Yes	Yes
Num Antennas	1	1	8
Does CS-DoA Sensing?	No	No	Yes
Key Features	Signal detector, 16 physical branches, Time Segmentation	Signal detector and receiver, noisce canceling, 4 physical branches, branch expansion and time segmentation.	Signal detector and beamforming receiver, 8 antennas in 2 subarrays of 4, branch expansion, interleaved CS-DOA and CS-Spectrum sensing.
Resource Cube Visualization			

Conclusions and Future Work

- A Compressed Sensing (CS) driven receiver architecture can be used to sense the RF spectrum in a fraction of the time as current state-of-the-art techniques

- As a proof-of-concept, a CS-enabled EM environment aware sensor was successfully integrated into the GNU Radio framework, the output of which indicates the spectral position of jammers or interferers

- Jammer location is sent to a machine learning-based decision engine (MLBDE) which in turn takes appropriate action (retunes, switches band, etc.) by making the optimal corrective decision based on reinforcement learning

- We are currently benchmarking the ITRN and comparing it against other EMEA sensing architectures.

References

[1] J. Mitola, “Cognitive radio for flexible mobile multimedia communications”, in Proc. IEEE Int. Workshop Mobile Multimedia Communications (MoMuC’99) (Cat. No.99EX384), Nov.1999, pp. 3–10.

[2] International Working Group “IEEE p802. 22/d1. 0 standard for wireless regional area networks part 22: Cognitive wireless ran medium access control (mac) and physical layer (phy) specifications: Policies and procedures for operation in the tv bands”, IEEE docs, pp. 22–06, 2008

[3] E. Candes, “Compressive sampling.” Proc. Int. Congress of Math, vol. Aug., pp. 67–94, 2006.

[4] M. Bajor et al., “An 8-Element, 1-3GHz Direct Space-to-Information Converter for Rapid, Compressive-Sampling Direction-of-Arrival Finding Utilizing Pseudo-Random Antenna-Weight Modulation,” in RFIC. IEEE, 2017.

[5] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” IEEE Trans. Inf. Theory, pp. 4655–4666, 2007.