RINCON RESEARCH
EMPLOYEE OWNED

Rabbit Ears 2022: Building a Phased Array using low-cost RTL-SDRs

Presented by: Spencer Shortt, Jona Soto Ramos de Oliveira, Nia Williams

Date: 26 July 2022

Presentation Outline

Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

Rabbit Ears Overview

Problem Statement

Design a *phased array* with beamforming capabilities for angle of arrival (AoA) estimation

Objective

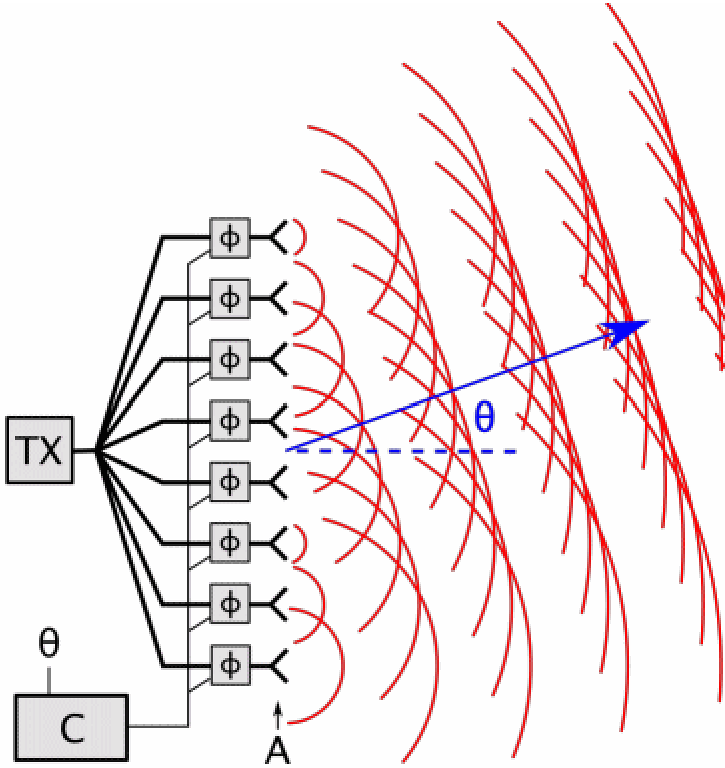
Explore possibilities and issues arising from low-cost hardware

- Timing issues
- Lack of processing power
- Inconsistency

…but inexpensive!

What is beamforming and a phased array?

- Constructive and destructive interference w.r.t EM waves
- Controllable array of several elements with tunable phase-offsets
- Phase-offsets allow for directional steering



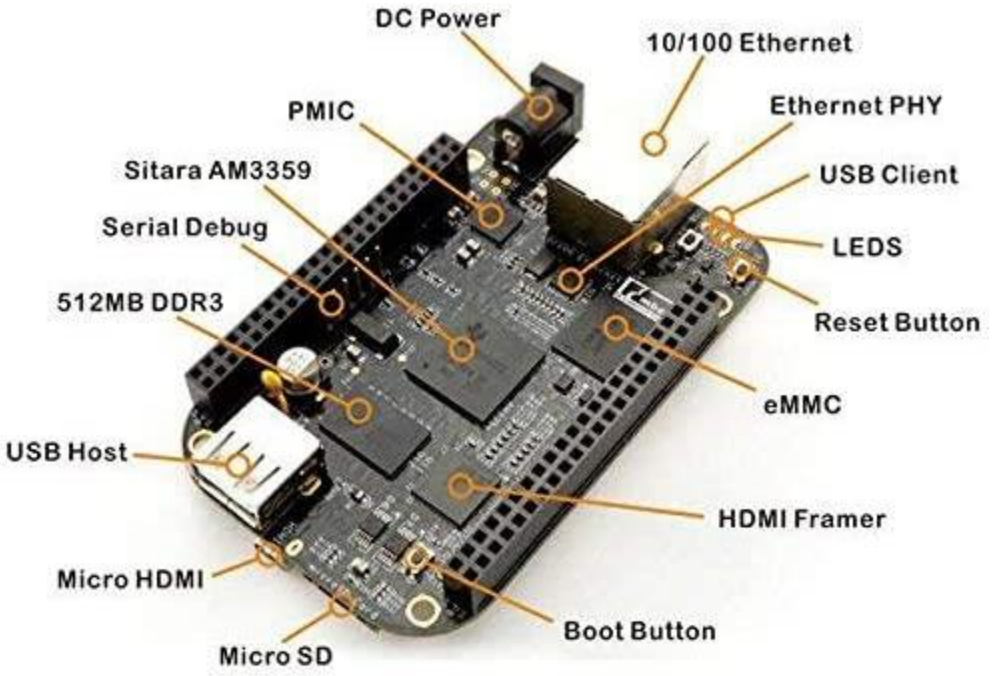
https://en.wikipedia.org/wiki/Phased_array

Hardware Overview

BeagleBone Black (BBB)

Low-cost, open-source, single board computer

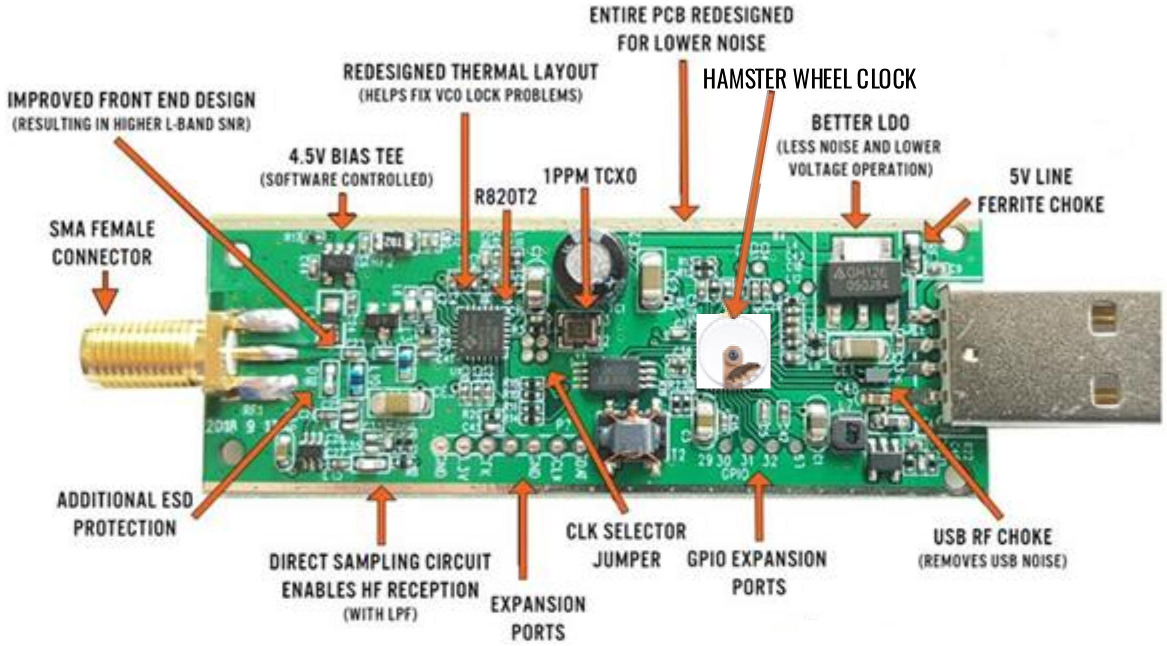
- 512 MB DDR3 RAM
- Ethernet
- 1x USB



RTL-SDR R860T

Low-cost, adapted DVB-T tuner run in debug

- RTL2832U chip collects raw I/Q data
- Shielding and heatsink (not shown)
- RX only (24-1766 MHz)



Helpful Tools

- These SDRs were not the inexpensive hardware we were investigating
- Just useful tools for transmitting and measurements

bladeRF

Mid-range consumer SDR

– 2x RX
– 2x TX

LimeSDR

Mid-range consumer SDR

– 4x TX
– 6x RX

System Overview

1. Source transmits electromagnetic signal

2. Raw data is received by antenna array

3. Data is translated by SDRs and collected on the BBBs

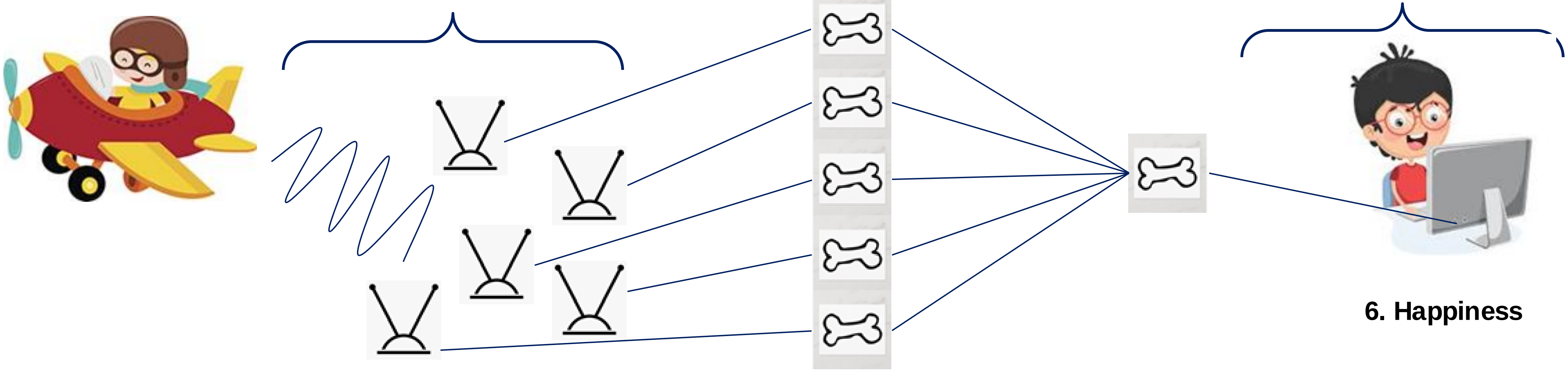
4. Master BB aggregates data

5. Beamforming to find angle of arrival (AoA)

Jona focused on BBB-BBB communication and subsystem interfacing

Nia focused on data collection and calibration

Spencer focused on beamforming approaches and implementation



6. Happiness

Presentation Outline

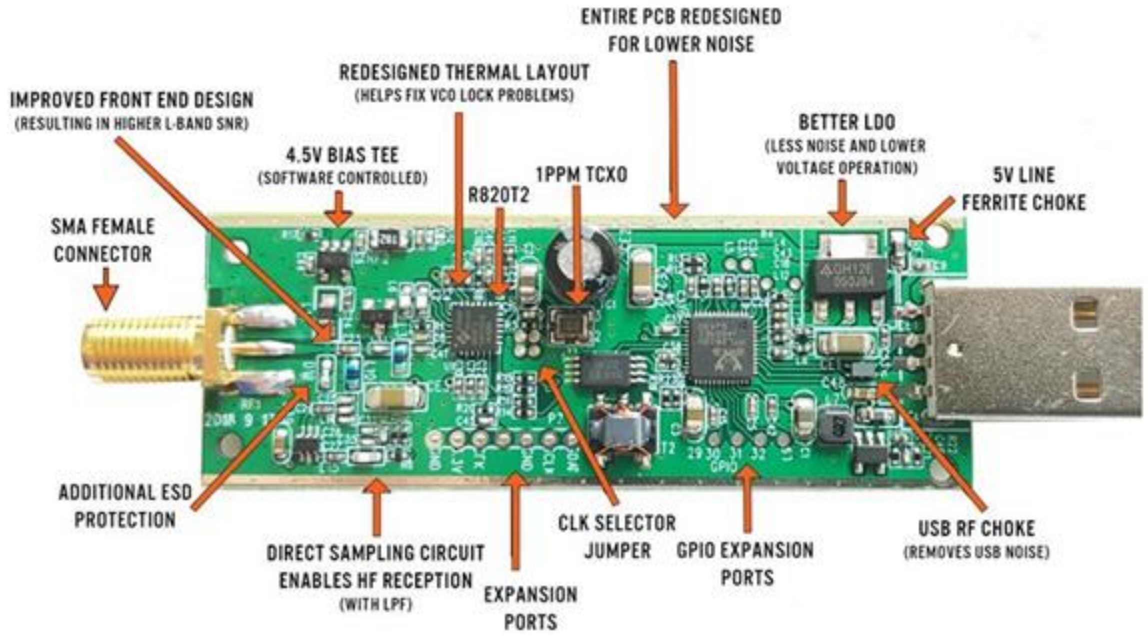
Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

RTL-SDR in Detail

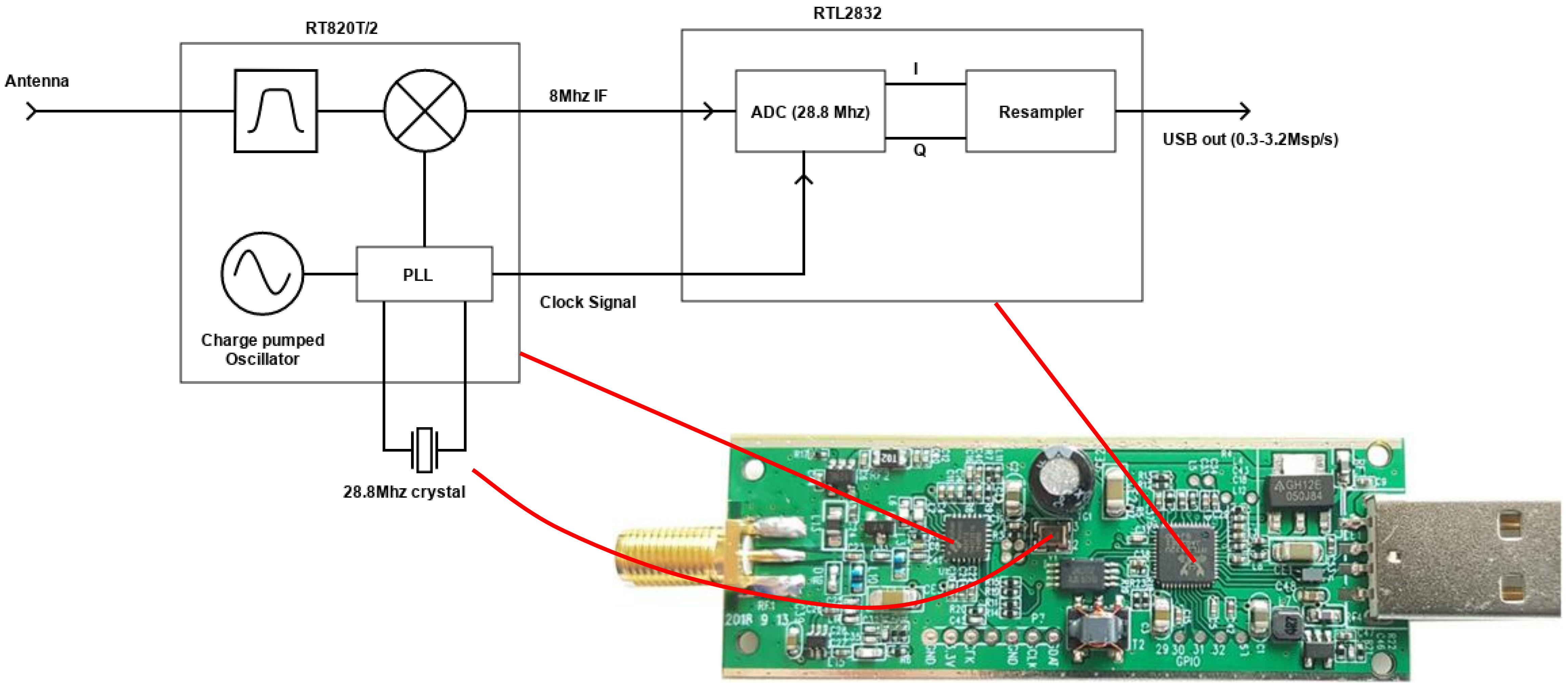
- RTL-SDRs can be mostly viewed as two major chips, the tuner, and the ADC
- Our RTLs include an R820T/2 tuner and an RTL2832 DVB-T/DAB demodulator
- The RTL chip is not actually meant to function as an SDR, instead we use 3rd-party drivers to enable debug mode and pull raw I/Q samples off of the chip

Challenges:

- Lack of proper documentation
- Mostly amateur documentation of chips
- No well defined performance tolerances

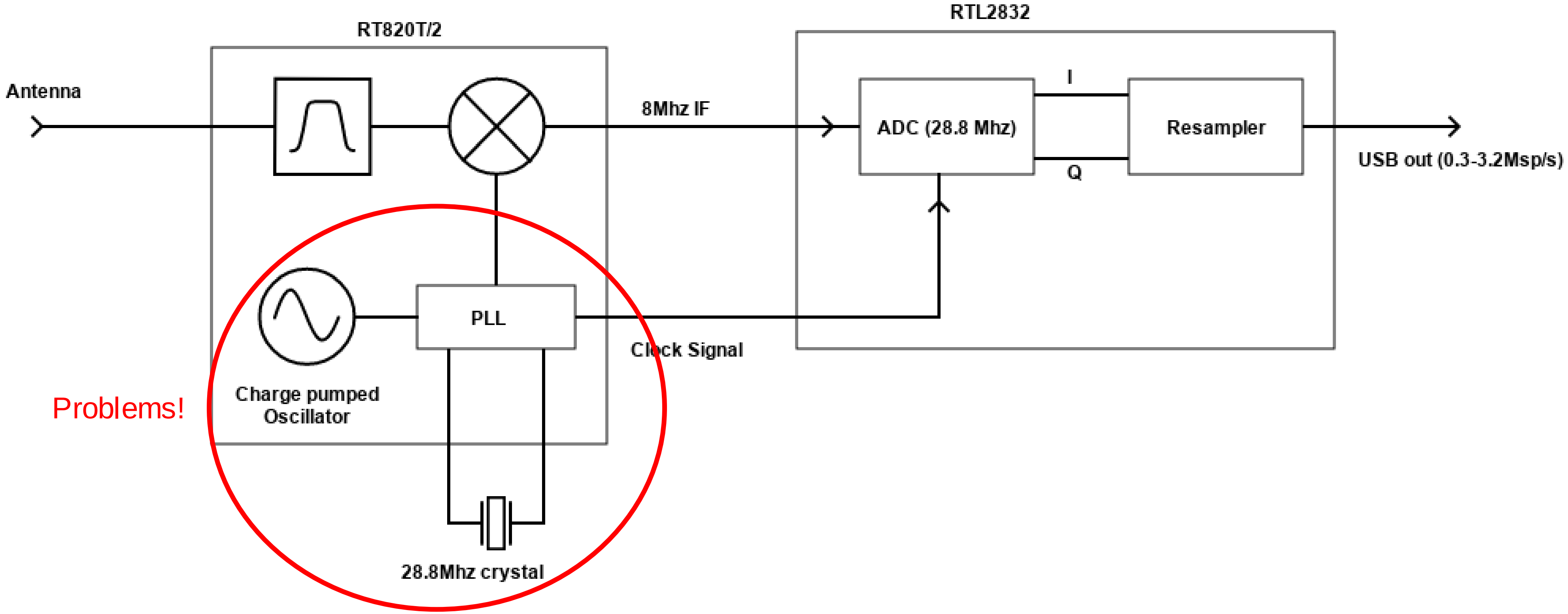


Simplified Block Diagram



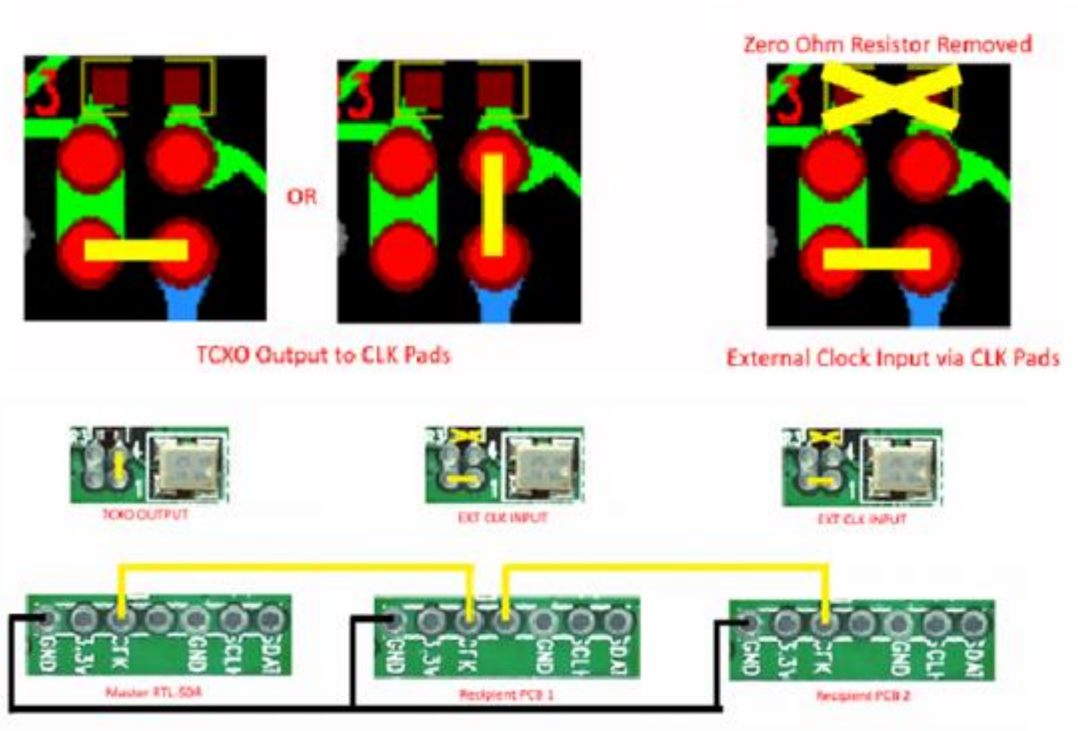
SDR Problems

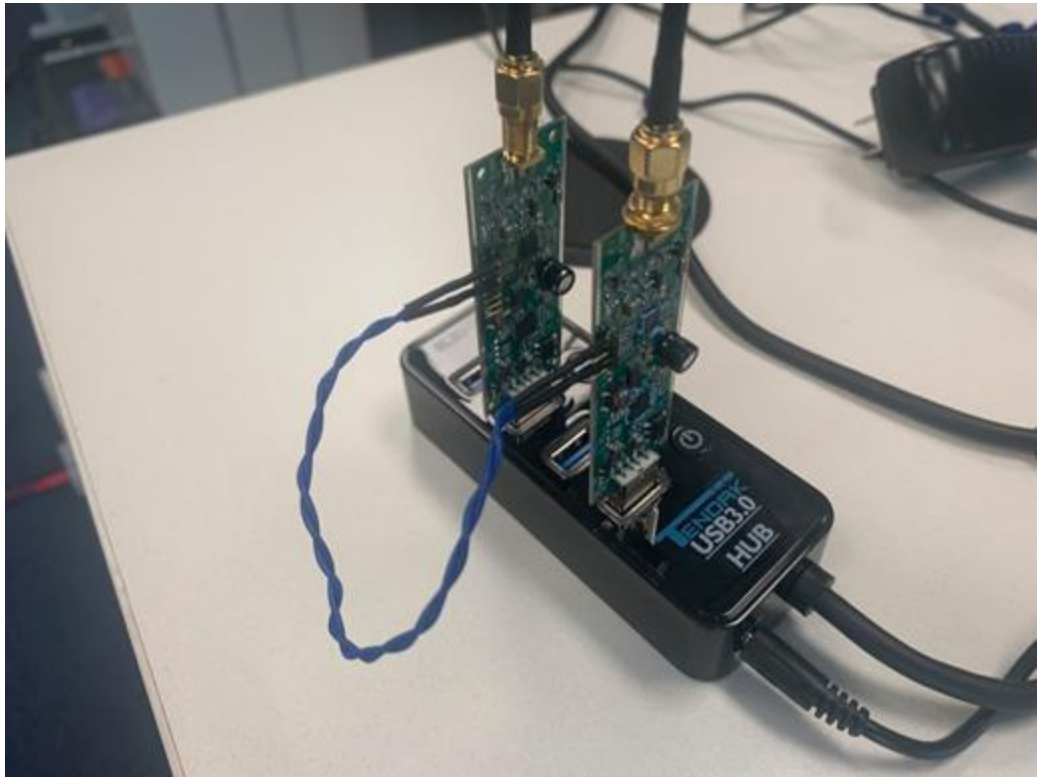
- Because each array element uses a different oscillator raw data is essentially useless
- RTL-SDR oscillators are relatively low quality, meaning we can’t trust them to stay at the same frequency
- SDR tuning frequencies are regularly off ±500Hz
- We need phase level accuracy and phase stability (Sub 1Hz accuracy)



First Attempt at Calibration

- First issue: Oscillator differences
- Each oscillator runs at 28.8Mhz which drives the sampling, but this can vary slightly between them (one may be at 28.8001Mhz and another might be 28.7999Mhz.
- Use one SDR oscillator to run another SDR

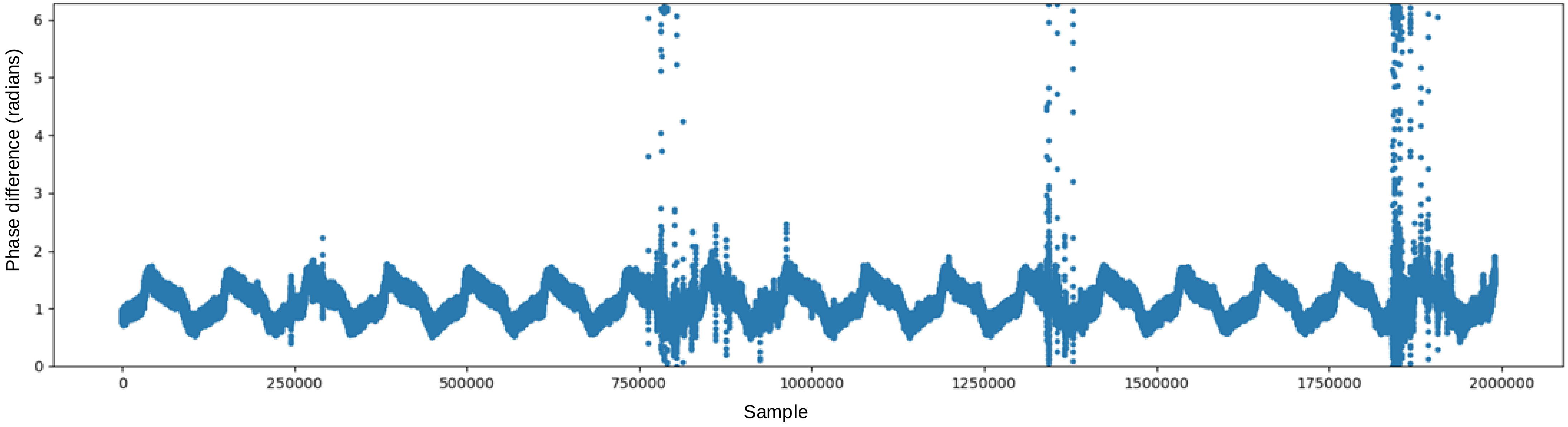




Issues with Connected Clocks

- This method causes an unexpected cyclic phase drift over time, that gets worse at higher frequencies
- Drift most likely comes from the crystal PLL struggling to drive two SDRs
- Having all the SDRs tied together is not ideal and could also require additional clock distribution circuits

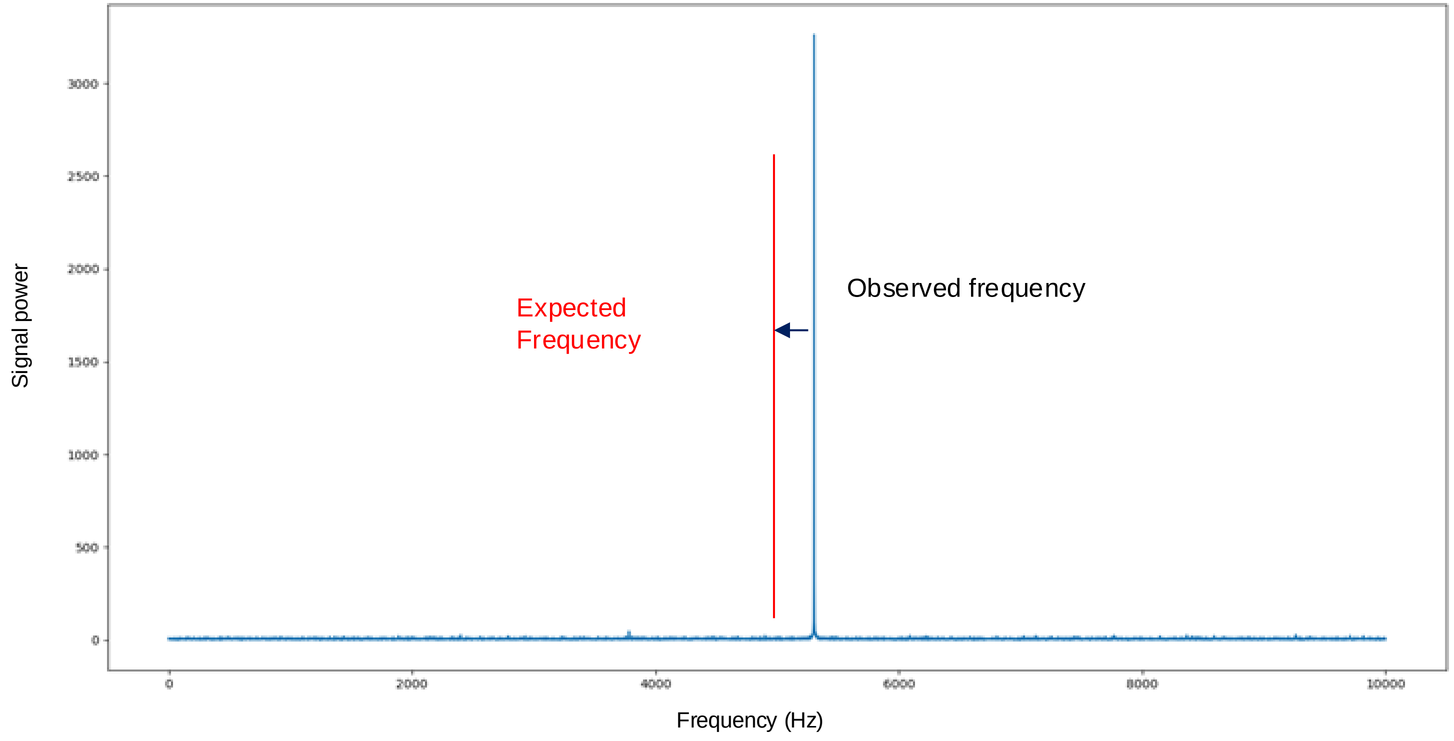
Frequency drift at 100Mhz



Correction Through Software

- First solve the clock mismatch issue
- Calibrate off of a reference tone generated by the the bladeRF
- Transform the data into the frequency domain, look at the frequency observed, then correct for the difference in the observed frequency and theoretical frequency

FFT windowed around calibration frequency

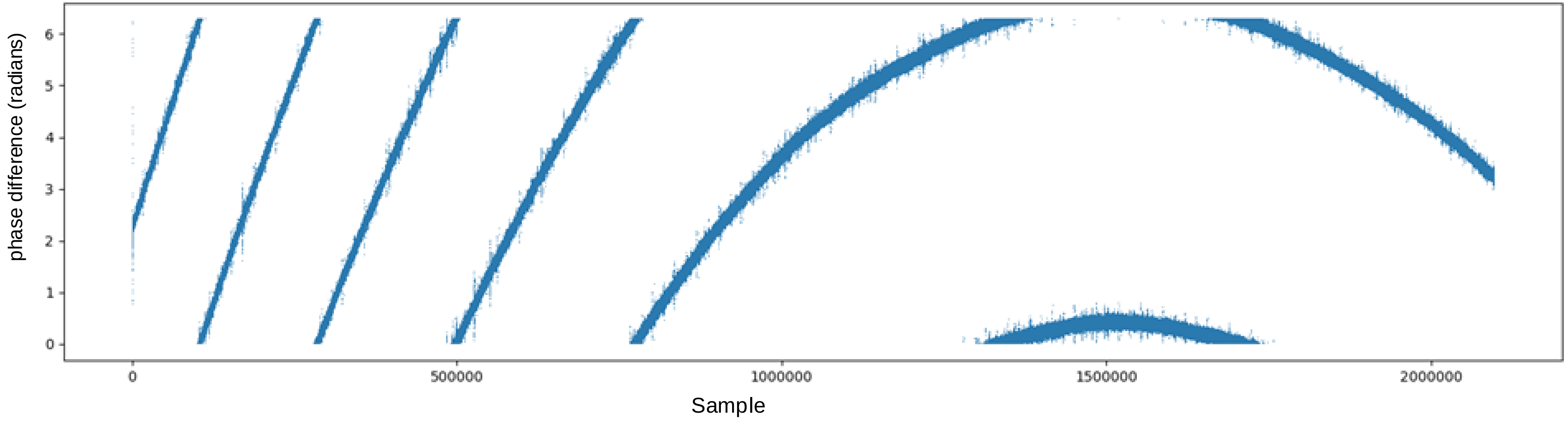


RINCON RESEARCH EMPLOYEE OWNED

Clock Drift

- As it turns out, correcting for differences in frequency is still not good enough, as the SDRs run they drift out of tune at different rates, again causing phase drift

2 Seconds of Drift at 1090Mhz

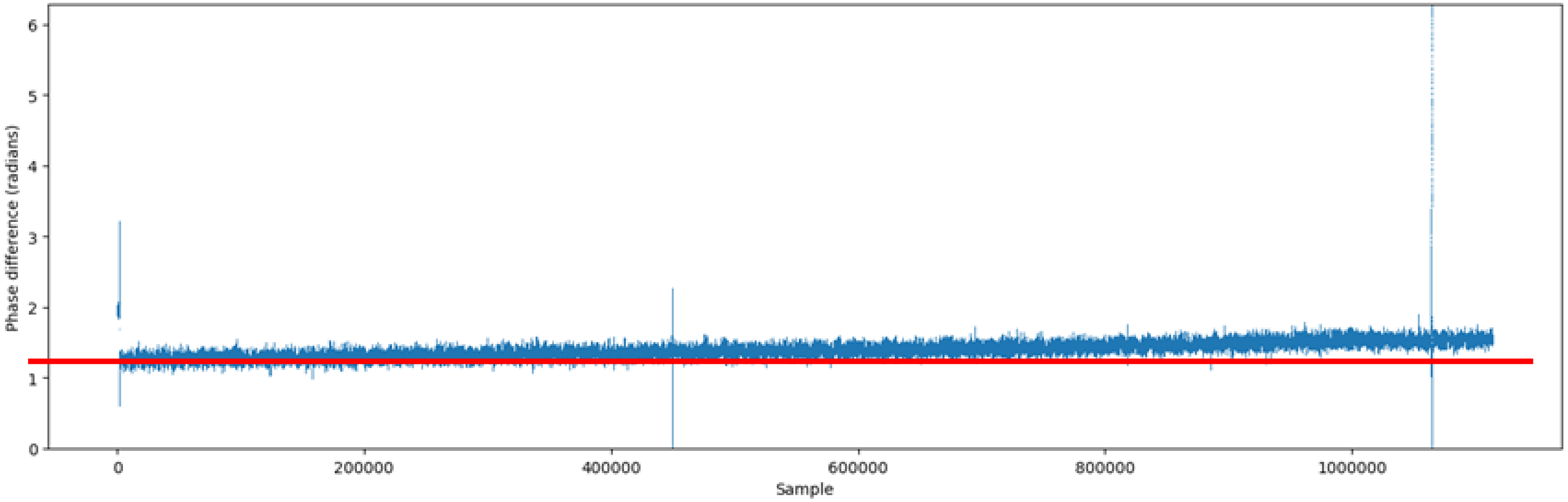


- This can still be corrected for, if we adjust the phases by the phase difference in the calibration signal we *should* get perfect timing

Using Two Calibration Signals

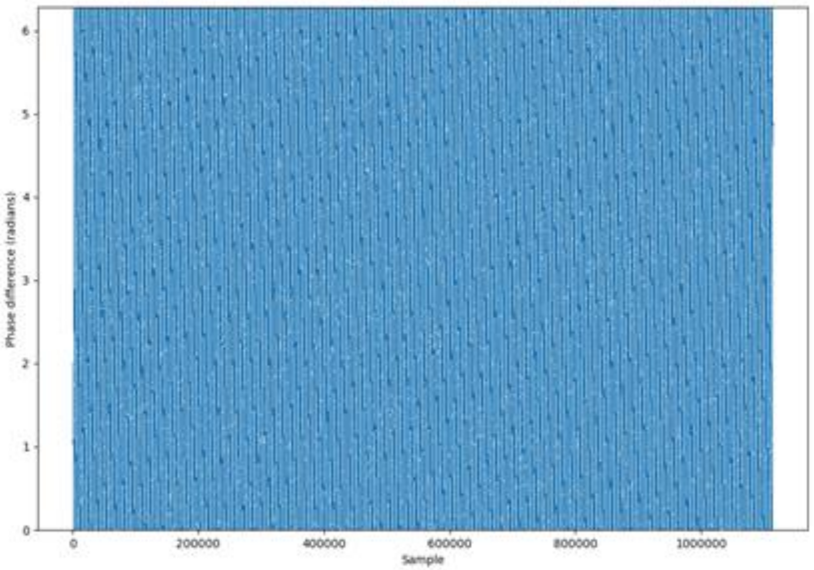
- Still some small amount of drift
- The drift was more severe the further our signal was from the observation signal

1 Second of Drift at 1090Mhz

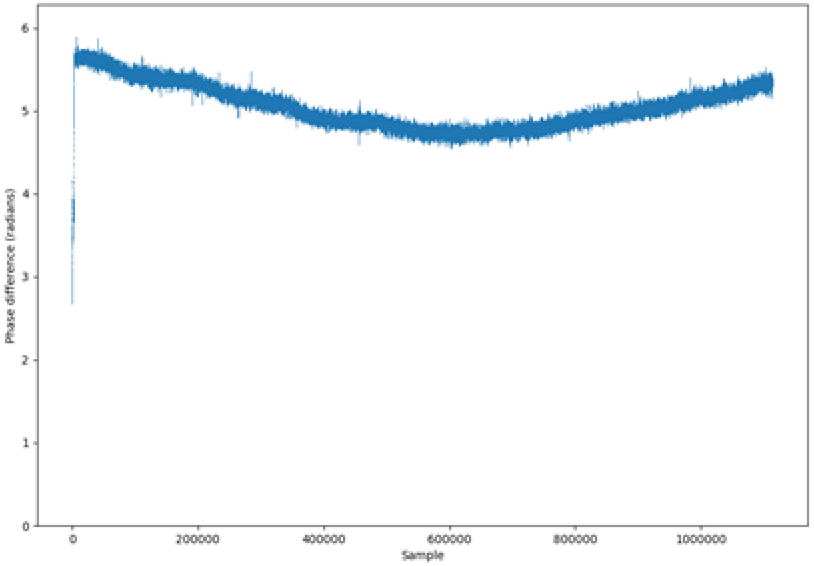


- This was solved by using two calibration signals, equally spaced from the observation signal
- By taking the average of these two calibration signals, we eliminate long term phase drift

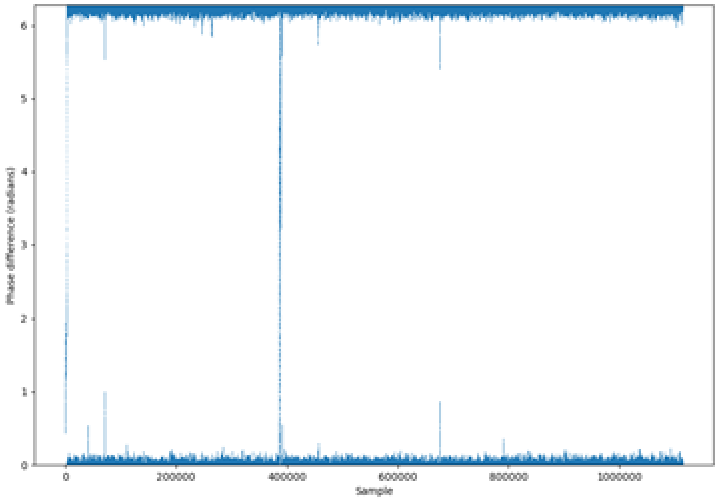
Final Calibration Routine



Frequency correction

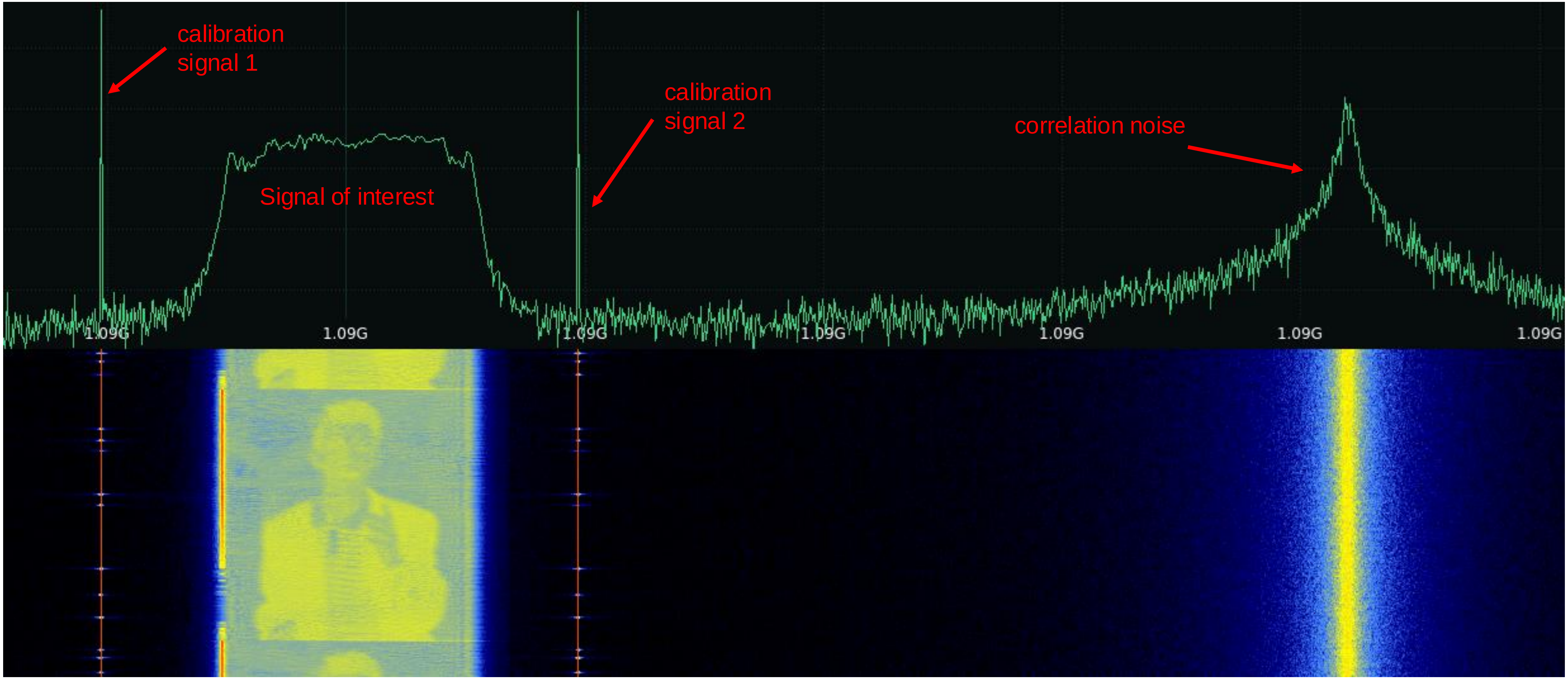


Phase correction



Calibration Signals

RTL Calibration
Nia

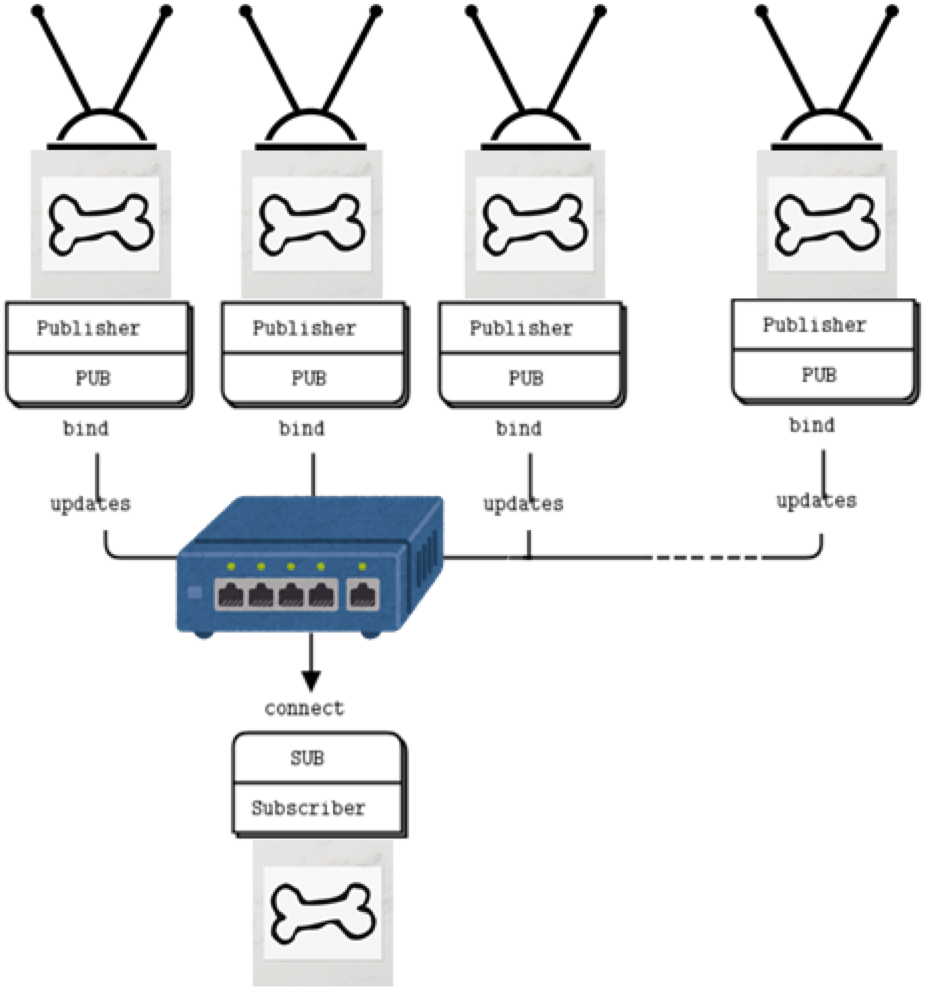


Presentation Outline

Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

Comms Subsystem Overview

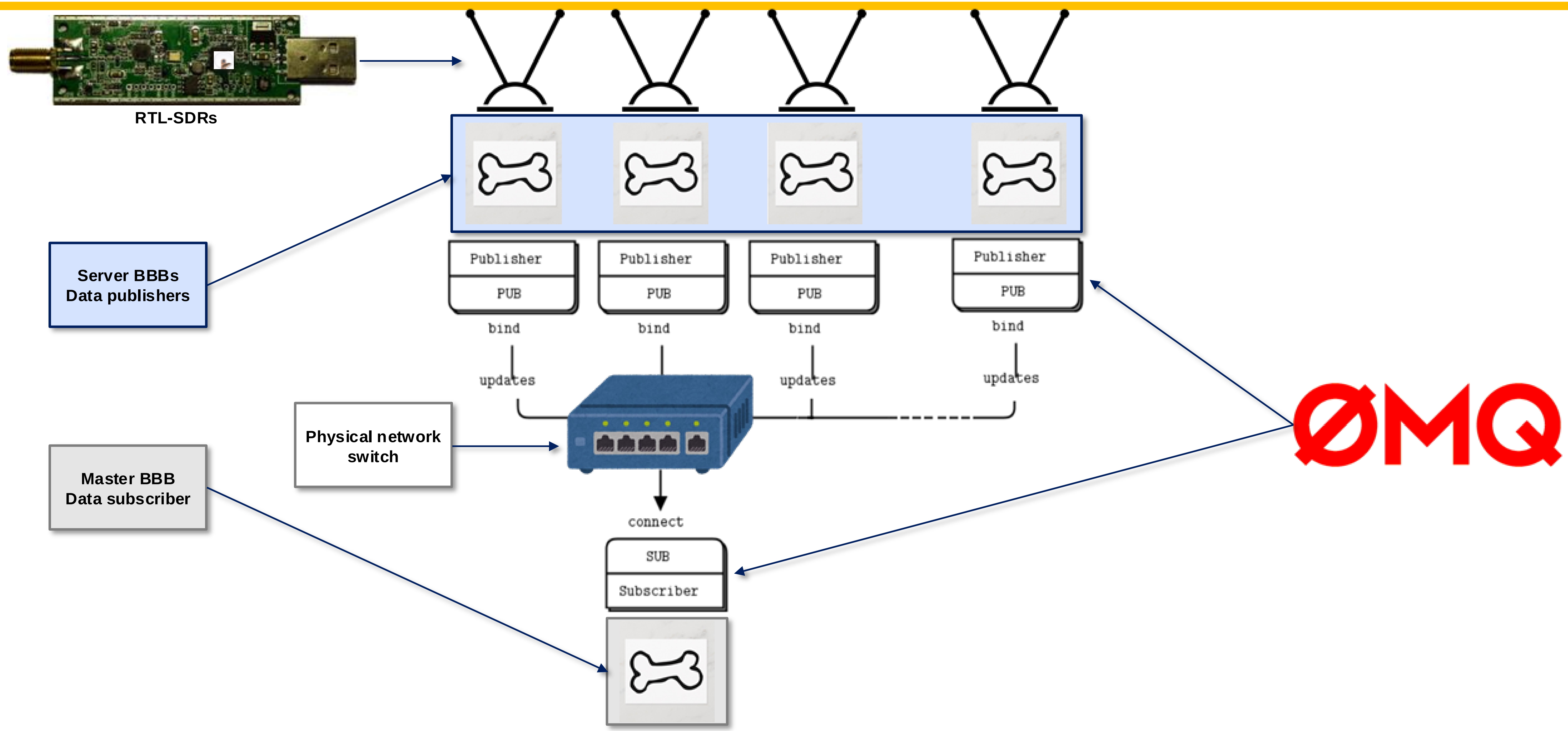
- Each BeagleBone Black (BBB) can drive only one RTL-SDR, so array data must be aggregated somewhere
- “Master” BBB serves to send commands and aggregate data
- How can BBBs transfer RX data?
  - Direct approaches like I^2C and SPI are relatively slow
  - ZMQ, a networking library, was fast and flexible
  - ZMQ can also be used over longer distances
- An inherent “not a bug, it’s a feature”-feature of ZMQ is that subscribers are liable to miss the first transmitted message



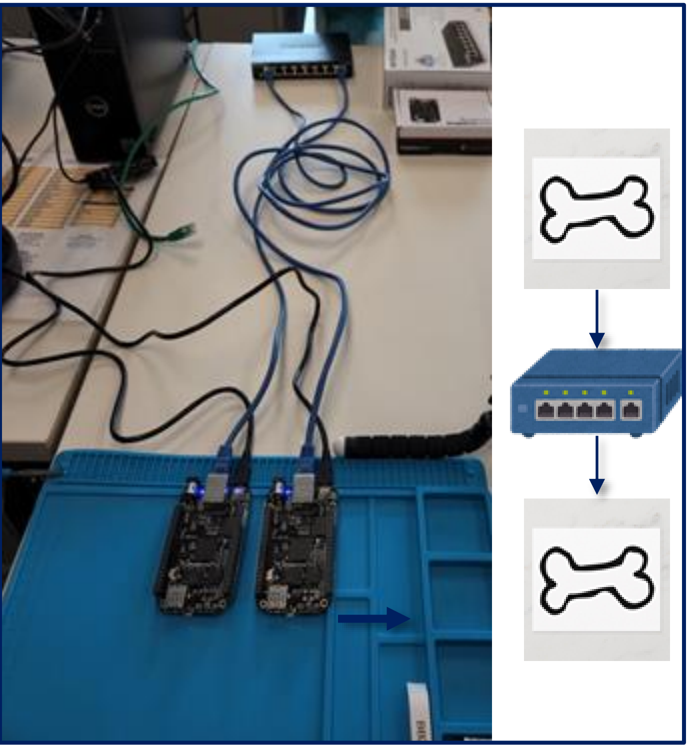
RINCON RESEARCH
EMPLOYEE OWNED

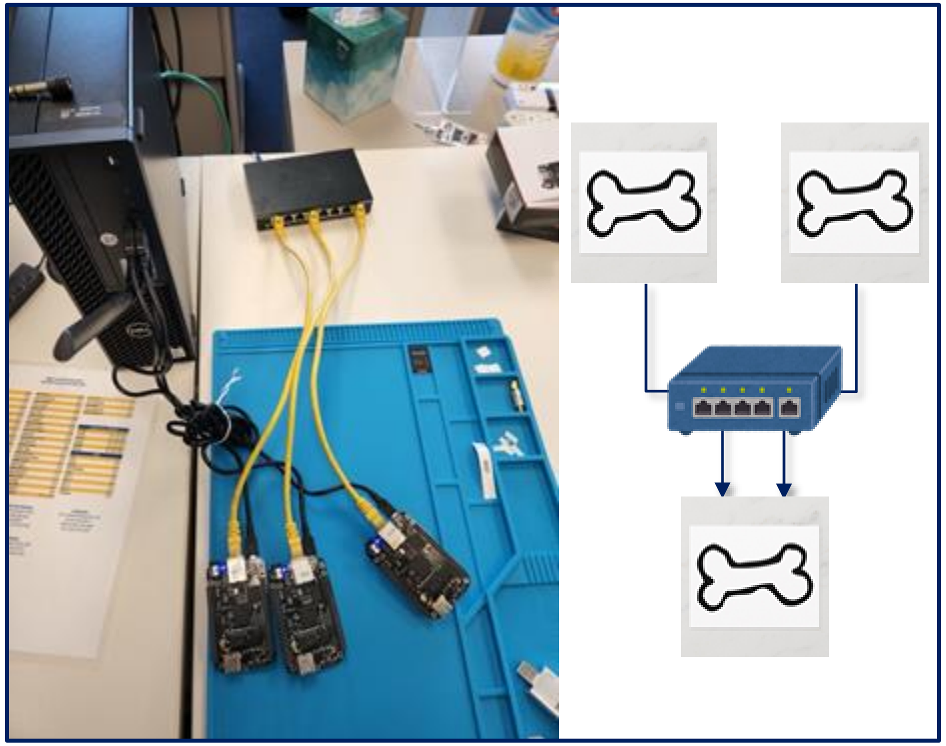
Subsystem Overview

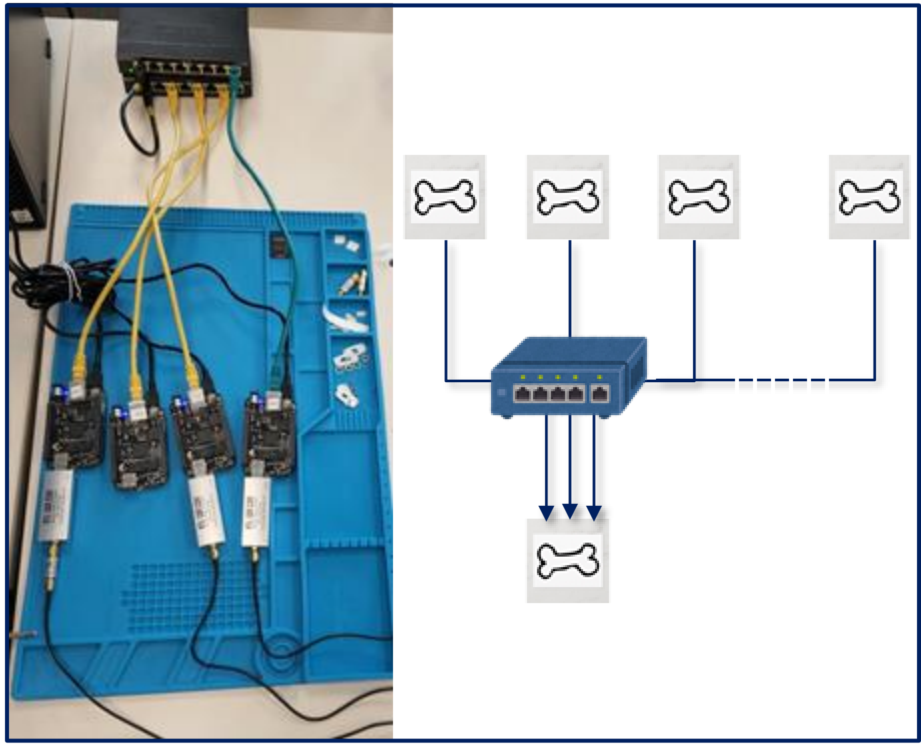
Comms System
Jona



Early BBB Comms

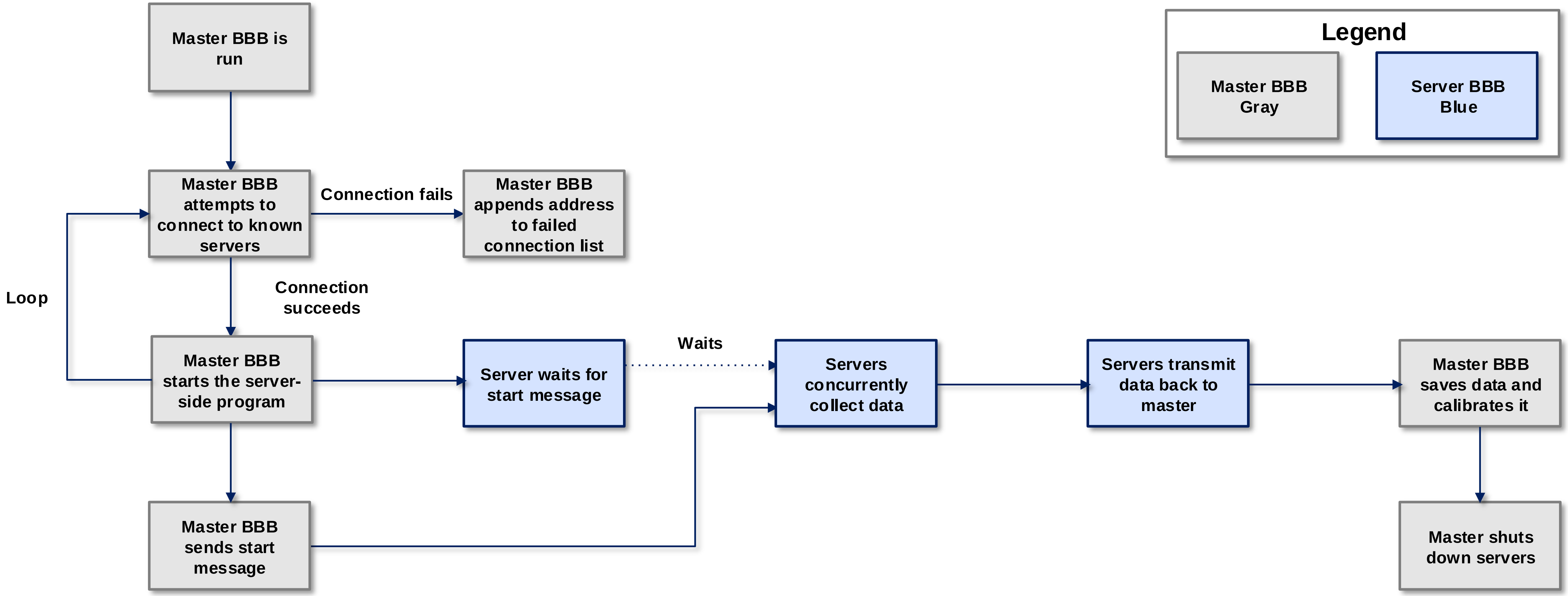






- Began as one-to-one messaging before many-to-one messaging
- Freshly set-up BBBs need only a copy of the server-side code and a unique ethernet address

BBB Program Flowchart

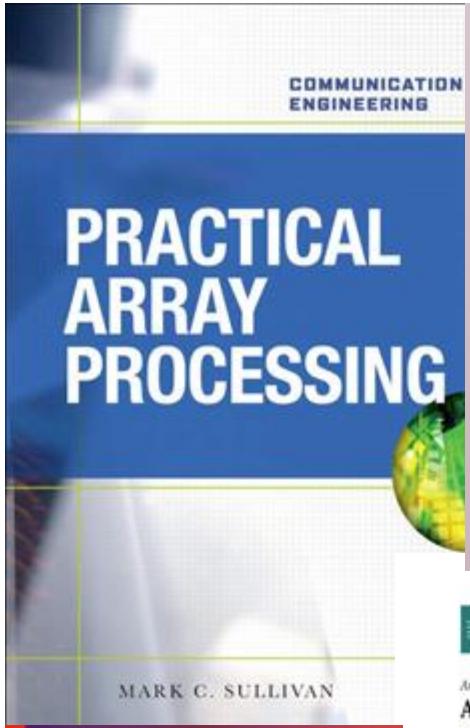


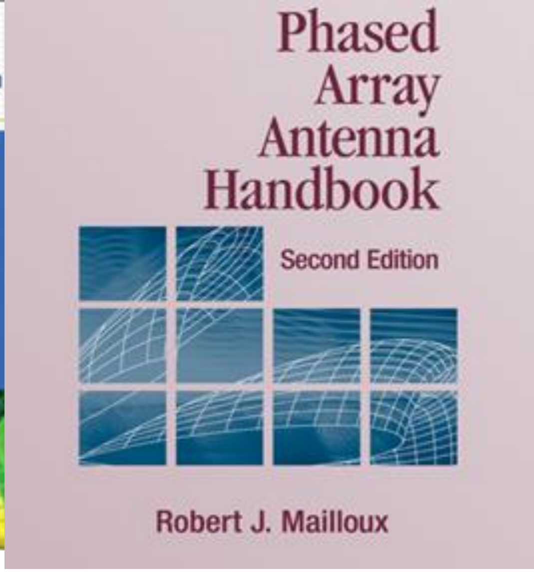
Presentation Outline

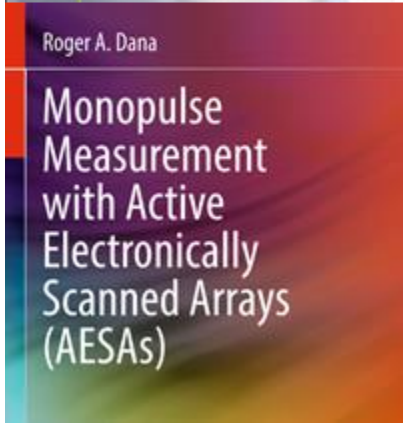
Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

RINCON RESEARCH
EMPLOYEE OWNED

Resources Used







Array Signal Processing Algorithms for Beamforming and Direction Finding

This thesis is submitted in partial fulfilment of the requirements for Doctor of Philosophy (Ph.D.)

Lei Wang
Communications Research Group
Department of Electronics
University of York

The output of the antenna array will vary based on the angle of arrival of an incident plane wave (as described here). In this manner, the array itself is a *spatial filter* - it filters incoming signals based on their angle of arrival. The output Y is a function of (θ,ϕ), the arrival angle of a wave relative to the array. In addition, if the array is transmitting, the radiation pattern will be identical in shape to the receive pattern, due to reciprocity.

Y can be written as:

$$Y = R(\theta,\phi)w_1e^{-j\mathbf{k}\cdot\mathbf{r}_1} + R(\theta,\phi)w_2e^{-j\mathbf{k}\cdot\mathbf{r}_2} + \cdots + R(\theta,\phi)w_Ne^{-j\mathbf{k}\cdot\mathbf{r}_N}$$

where **k** is the wave vector of the incident wave. The above equation can be factor simply as:

$$Y = R(\theta,\phi)\sum_{i=1}^{N} w_ie^{-j\mathbf{k}\cdot\mathbf{r}_i}$$

$$= R(\theta,\phi)AF$$

$$\boxed{AF = \sum_{i=1}^{N} w_ie^{-j\mathbf{k}\cdot\mathbf{r}_i}}$$

sensors
MDPI

Article

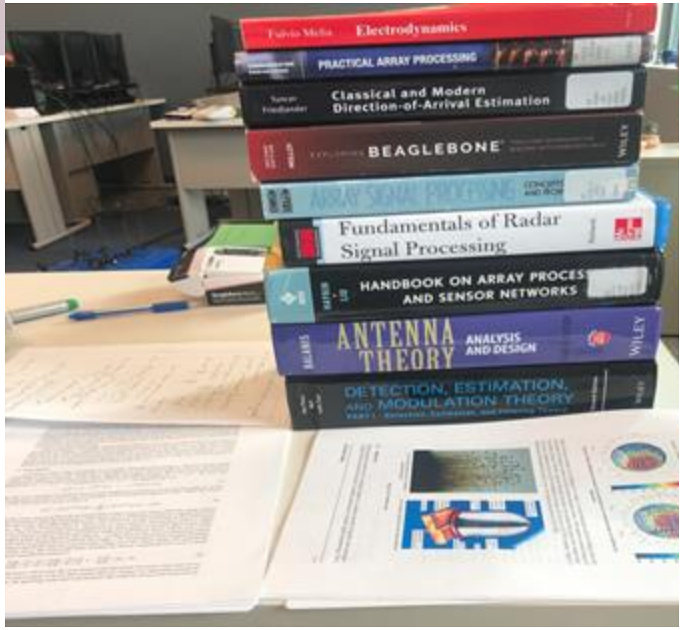
A Novel Monopulse Technique for Adaptive Phased Array Radar

Xinyu Zhang [1,3], Yang Li [1,4,*], Xiaopeng Yang [1], Le Zheng [2], Teng Long [1] and Christopher J. Baker [3]

1 Department of Information and Electronics Engineering, Beijing Institute of Technology, Beijing 100081, China; zhangxinyu90111@gmail.com (X.Z.); xiaopengyang@bit.edu.cn (X.Y.); longteng@bit.edu.cn (T.L.)
2 School of Electrical Engineering, Columbia University, New York, NY 10027, USA; lezheng84010@163.com
3 ElectroScience Laboratory, Ohio State University, Columbus, OH 43212, USA; baker.1891@osu.edu
4 Beijing Key Laboratory of Embedded Real-time Information Processing Technology, Beijing Institute of Technology, Beijing 100081, China
* Correspondence: bit_liyang@bit.edu.cn; Tel.: +86-10-6891-8390

Academic Editor: Changzhi Li
Received: 27 September 2016; Accepted: 27 December 2016; Published: 8 January 2017

Abstract: The monopulse angle measuring technique is widely adopted in radar systems due to its simplicity and speed in accurately acquiring a target's angle. However, in a spatial adaptive array, beam distortion, due to adaptive beamforming, can result in serious deterioration of monopulse performance. In this paper, a novel constrained monopulse angle measuring algorithm is proposed for spatial adaptive arrays. This algorithm maintains the ability to suppress the unwanted signals without suffering from beam distortion. Compared with conventional adaptive monopulse methods, the proposed algorithm adopts a new form of constraint in forming the difference beam with the merit that it is more robust in most practical situations. At the same time, it also exhibits the simplicity



NATO OTAN

Fundamentals of Signal Processing for Phased Array Radar

Dr. Ulrich Nickel
Research Institute for High-Frequency Physics and Radar Techniques (FHR)
Research Establishment for Applied Science (FGAN)
53343 Wachtberg, Germany
nickel@fgan.de

ABSTRACT

This section gives a short survey of the principles and the terminology of phased array radar. Beamforming, radar detection and parameter estimation are described. The concept of subarrays and monopulse estimation with arbitrary subarrays is developed. As a preparation to adaptive beam forming, which is treated in several other sections, the topic of pattern shaping by deterministic weighting is presented in more detail.

1.0 INTRODUCTION

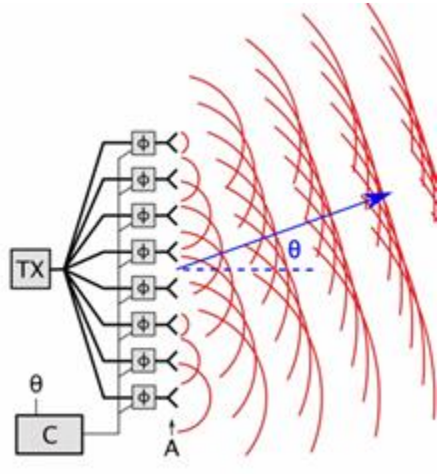
Arrays are today used for many applications and the view and terminology is quite different. We give here an introduction to the specific features of radar phased array antennas and the associated signal processing following the description of [1]. First the radar principle and the terminology is explained. Beamforming with a large number of array elements is the typical radar feature and the problems with such antennas are in other applications not known. We discuss therefore the special problems of fully filled arrays, large apertures and bandwidth. To reduce cost and space the antenna outputs are usually summed up into subarrays. Digital processing is done only with the subarray outputs. The problems of such partial analogue and digital beamforming, in particular the grating problems are discussed. This topic will be reconsidered for adaptive beamforming, space-time adaptive processing (STAP), and SAR.

Radar detection, range and direction estimation is derived from statistical hypotheses testing and parameter estimation theory. The main application of this theory is the derivation of adaptive beamforming to be considered in the following lectures. In this lecture we present as an application the derivation of the monopulse estimator which is in the following lectures extended to monopulse estimators for adaptive arrays or STAP.

Beamforming

- Generally speaking, there are two types of beamforming:
  - Conventional beamforming: fixed weights for the inputs at each antenna
  - Adaptive beamforming: data-dependent, more algorithmic approaches of increasing the SINR (Signal to Interference and Noise Ratio)

What I thought beamforming would be:



What it actually is:

$$Y = \sum_{n=1}^{N} w_n X_n$$

Conventional Beamforming

$$E(x, y, z) = e^{-j\boldsymbol{k}\cdot\boldsymbol{r}}$$
$$= e^{-j|\boldsymbol{k}|(x\sin\theta\cos\phi + y\sin\theta\sin\phi + z\cos\theta)}$$

$$|\boldsymbol{k}|^2 = (\frac{2\pi}{\lambda})^2$$

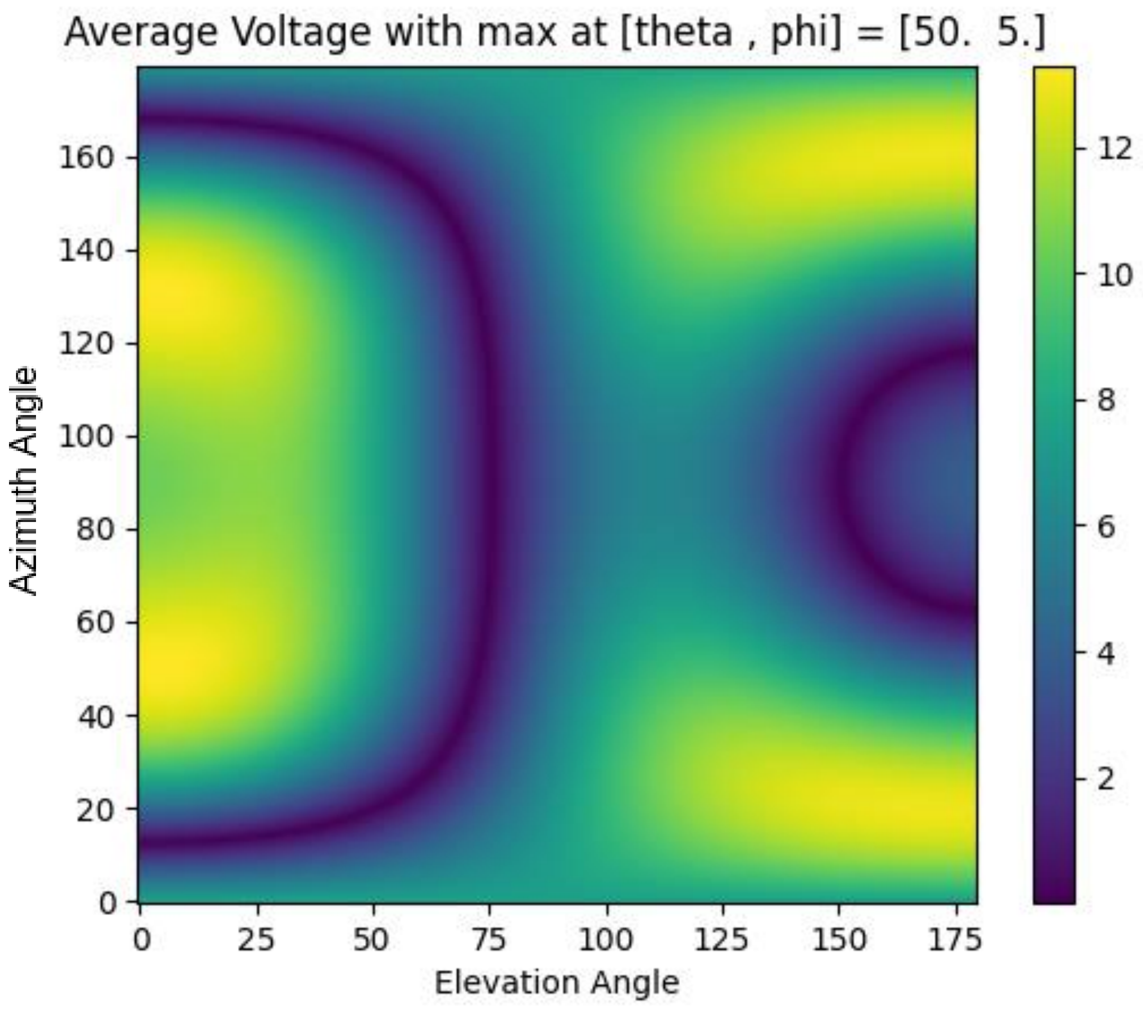
$$Y = \sum_{n=1}^{N} w_n X_n = \sum_{n=1}^{N} e^{j(\boldsymbol{k}_0 - \boldsymbol{k})\cdot\boldsymbol{r}_n}$$

$$X_n = e^{-j|\boldsymbol{k}|(x_n\sin\theta\cos\phi + y_n\sin\theta\sin\phi + z_n\cos\theta)}$$
$$w_n = e^{j|\boldsymbol{k}|(x_n\sin\theta_0\cos\phi_0 + y_n\sin\theta_0\sin\phi_0 + z_n\cos\theta_0)}$$

- This is conventional beamforming in a single direction (θ_0, ϕ_0)
- What if we form coefficients for every direction and find the angles that give maximum output? Can we detect the angle that a signal is coming from?

Conventional Beamforming

- Let’s simulate a simple case!
- Circular array with 6 antennas
- Transmitted signal: sine wave with Gaussian noise and no interference
- One drawback is that interference isn’t handled well



Capon Algorithm for Adaptive BF

- One of the algorithms that handles interference well for ULAs with a known AoA is the Capon beamformer.

$$SINR_{out} = \frac{E[\,|\boldsymbol{w}^H \boldsymbol{x}_s(k)|^2\,]}{E[\,|\boldsymbol{w}^H(\boldsymbol{x}_{int}(k) + \boldsymbol{n}(k))|^2\,]} = \frac{\sigma_0^2 |\boldsymbol{w}^H \boldsymbol{a}|^2}{\boldsymbol{w}^H \boldsymbol{R}_{i+n} \boldsymbol{w}}$$

$$\boldsymbol{R}_{i+n} = E[\,(\boldsymbol{x}_{int} + \boldsymbol{n}(k))(\boldsymbol{x}_{int} + \boldsymbol{n}(k))^H\,]$$

$$\min_{\boldsymbol{w}} \ \boldsymbol{w}^H \boldsymbol{R}_{i+n} \boldsymbol{w} \quad \text{subject to } \boldsymbol{w}^H \boldsymbol{a} = 1$$

$$\boldsymbol{R}_x = E[\,\boldsymbol{x}(k)\boldsymbol{x}^H(k)\,] = \sigma_0^2 \boldsymbol{a}\boldsymbol{a}^H + \boldsymbol{R}_{i+n}$$

$$\boldsymbol{w}^H \boldsymbol{R}_x \boldsymbol{w} = \boldsymbol{w}^H \boldsymbol{R}_{i+n} \boldsymbol{w} + \sigma_0^2 |\boldsymbol{w}^H \boldsymbol{a}|^2$$

$$\min_{\boldsymbol{w}} \ \boldsymbol{w}^H \boldsymbol{R}_x \boldsymbol{w} \quad \text{subject to } \boldsymbol{w}^H \boldsymbol{a} = 1$$

$$\boldsymbol{w} = \frac{\boldsymbol{R}_x^{-1} \boldsymbol{a}}{\boldsymbol{a}^H \boldsymbol{R}_x^{-1} \boldsymbol{a}}$$

$$\hat{\mathrm{R}}_x = \frac{1}{N} \sum_{k=1}^{N} \boldsymbol{x}(k)\boldsymbol{x}^H(k)$$

ARL Python

arlpy

latest

Search docs

Signal processing
Communications
Beamforming and array processing
Stable distributions
Geographical coordinates
Underwater acoustics
Underwater acoustic propagation modeling
Plotting utilities
Common utilities
Digital Towed Array
ROMANIS
High frequency data acquisition system
Unet modem & network stack

Docs » ARL Python Tools

Edit on GitHub

ARL Python Tools

Packages such as *numpy* and *scipy* provide excellent mathematical tools for scientists and engineers using Python. However, these packages are still young and evolving, and understandably have some gaps, especially when it comes to domain-specific requirements. The *arlpy* package aims to fill in some of the gaps in the areas of underwater acoustics, signal processing, and communication. Additionally, *arlpy* also includes some commonly needed utilities and plotting routines based on *bokeh*.

General modules

The following modules are general and are likely to be of interest to researchers and developers working on signal processing, communication and underwater acoustics:

- Signal processing
- Communications
- Beamforming and array processing
- Stable distributions

arlpy.bf.capon(*x, fc, sd, complex_output=False*)

Frequency-domain Capon beamformer.

The timeseries data must be 2D with narrowband complex timeseries for each sensor in individual rows. The steering delays must also be 2D with a row per steering direction.

If the timeseries data is specified as 1D array, it is assumed to represent multiple sensors at a single time.

The covariance matrix of x is estimated over the entire timeseries, and used to compute the optimal weights for the Capon beamformer.

Parameters:
- **x** – narrowband complex timeseries data for multiple sensors (row per sensor)
- **fc** – carrier frequency for the array data (Hz)
- **sd** – steering delays (s)
- **complex_output** – True for complex signal, False for beamformed power

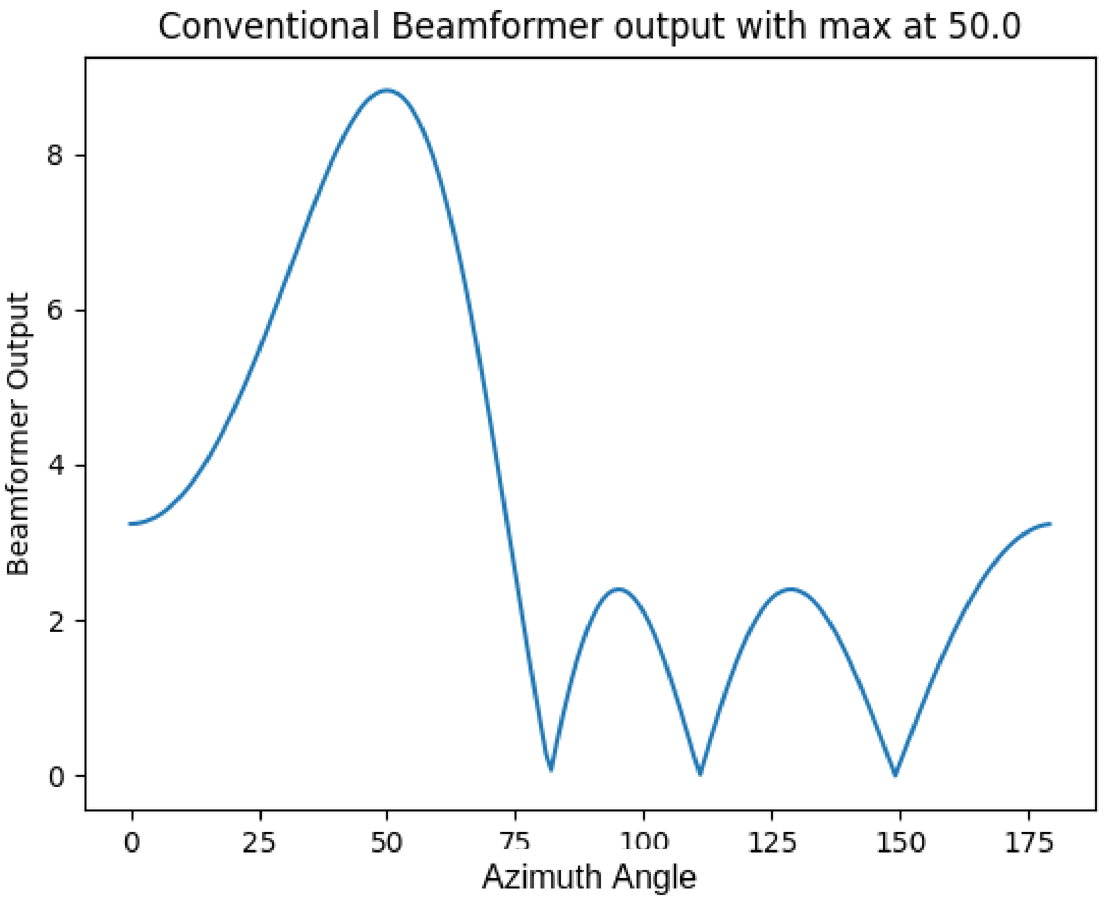
Returns: beamformer output averaged across time

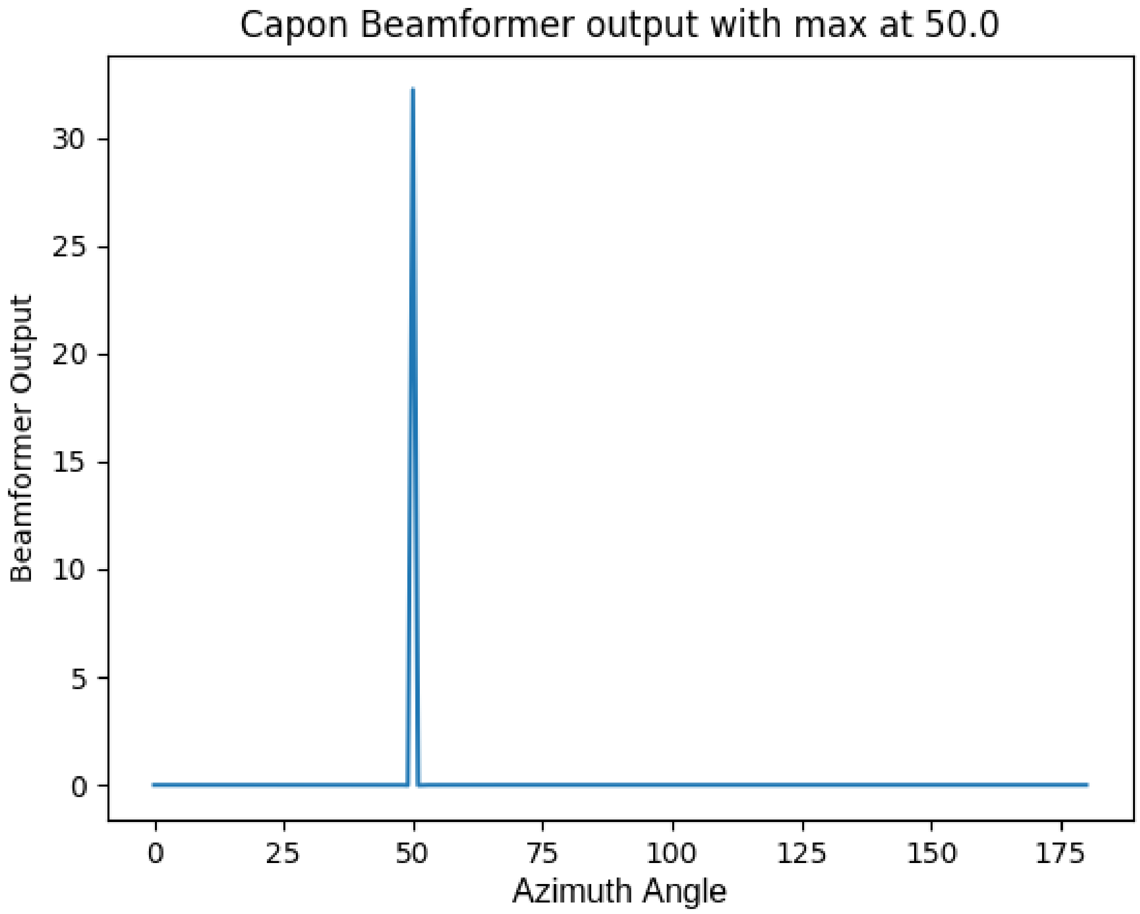
```
>>> from arlpy import bf
>>> import numpy as np
>>> # narrowband (1 kHz) timeseries array data assumed to be loaded in x
>>> # sensor positions assumed to be in pos
>>> y = bf.capon(x, 1000, bf.steering(pos, 1500, np.linspace(-np.pi/2, np.pi/2, 181)))
```

Simulating With No Interference

$$Y = \sum_{n=1}^{N} w_n X_n$$

$$\frac{1}{\mathbf{w}^H \mathbf{R}_x \mathbf{w}}$$

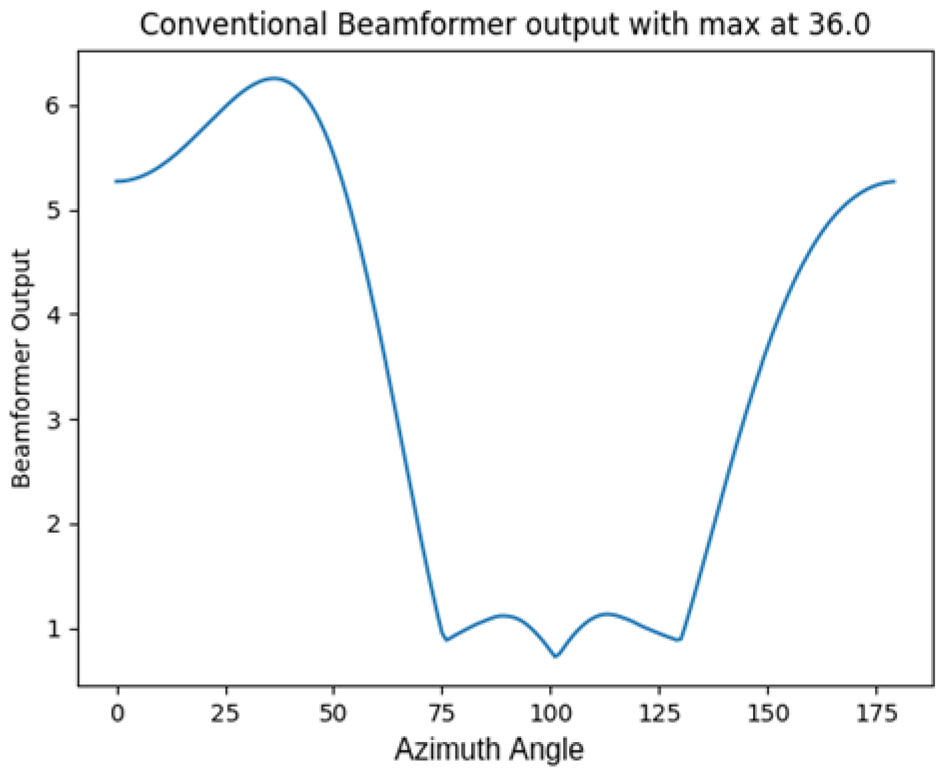


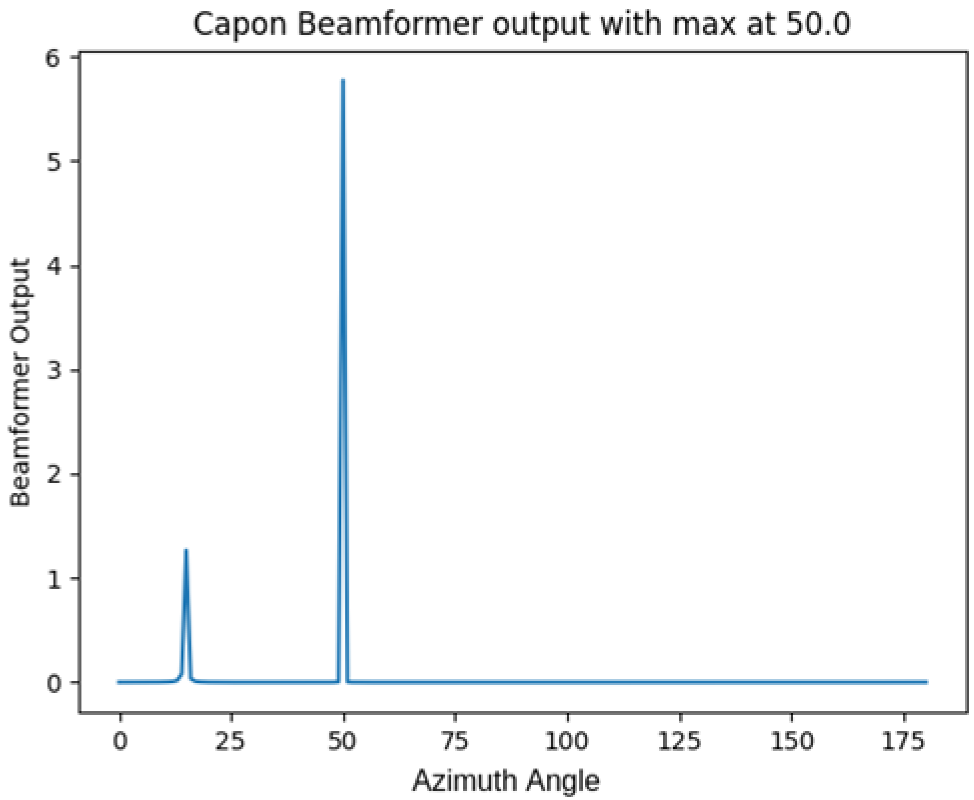


Simulating With One Interfering Signal

$$Y = \sum_{n=1}^{N} w_n X_n$$

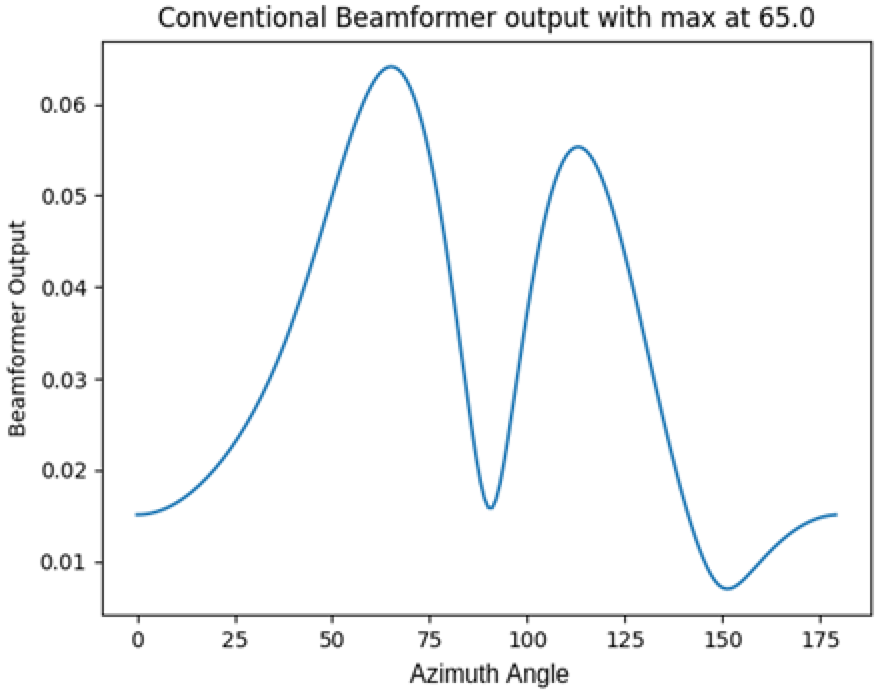
$$\frac{1}{\mathbf{w}^H \mathbf{R}_x \mathbf{w}}$$



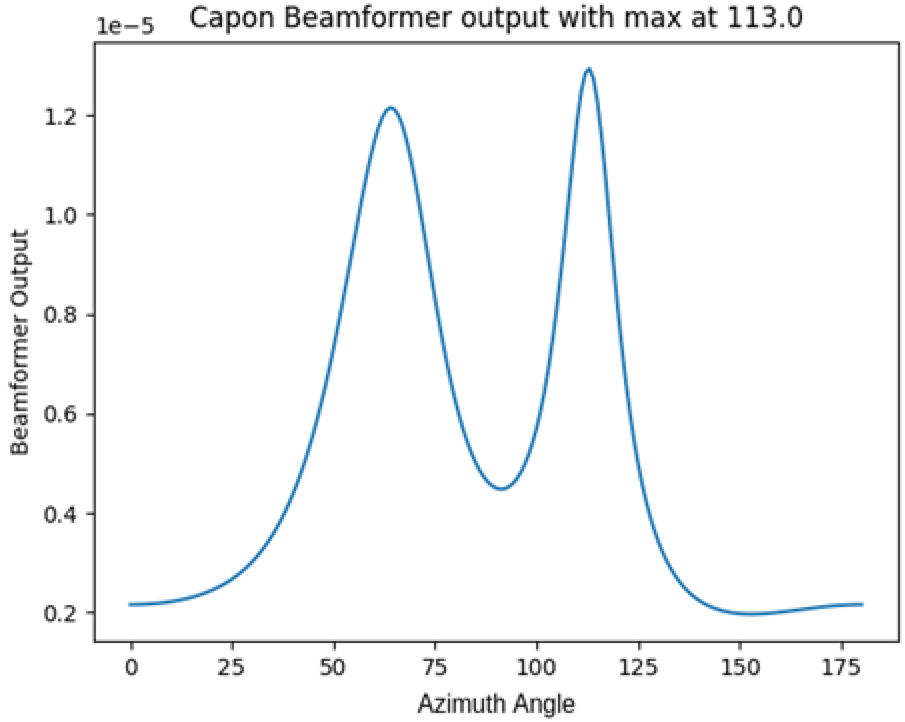


Beamforming on RTL Data

$$Y = \sum_{n=1}^{N} w_n X_n$$



$$\frac{1}{\mathbf{w}^H \mathbf{R}_x \mathbf{w}}$$



- These angles are **incorrect**!
- Let's try receiving a few more times
- Oh no! We get wildly different angles every run, even though we didn't move the transmitter

Presentation Outline

Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

Sidequesting Begins!

Sidequest Goals

- To test and refine our beamforming approach, we wanted to remove the RTLs from the equation
- Used a LimeSDR to collect dual-channel receive data
- The LimeSDR was chosen mostly because it was available (this is foreshadowing)

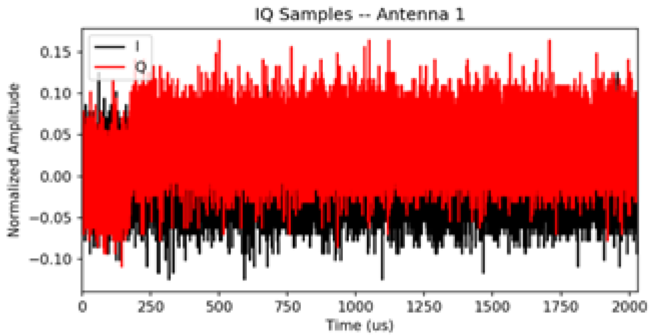
Side quest accepted!
(Week 5+)

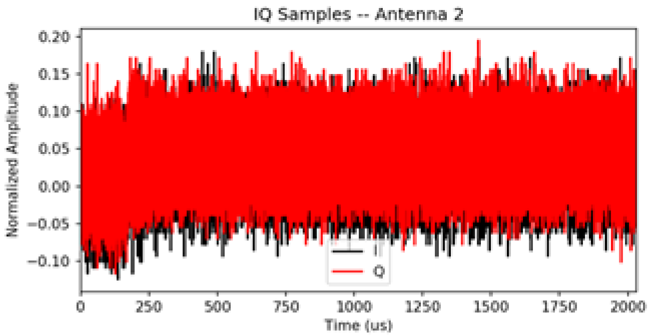
Objectives:
❑ Set-up LimeSDR
❑ Set-up dual Rx
❑ Collect data
❑ Find AoA

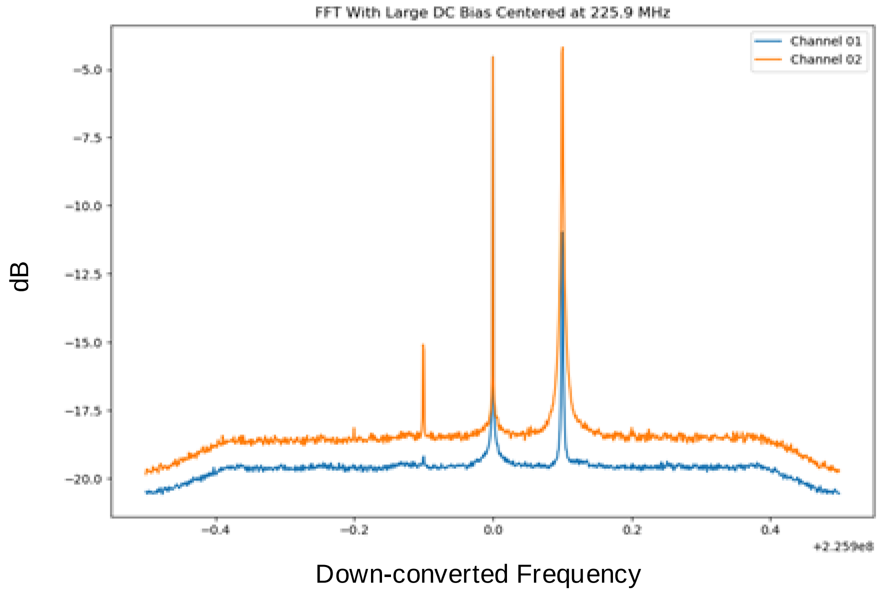
Reward: Happiness.
(and 100 exp.)

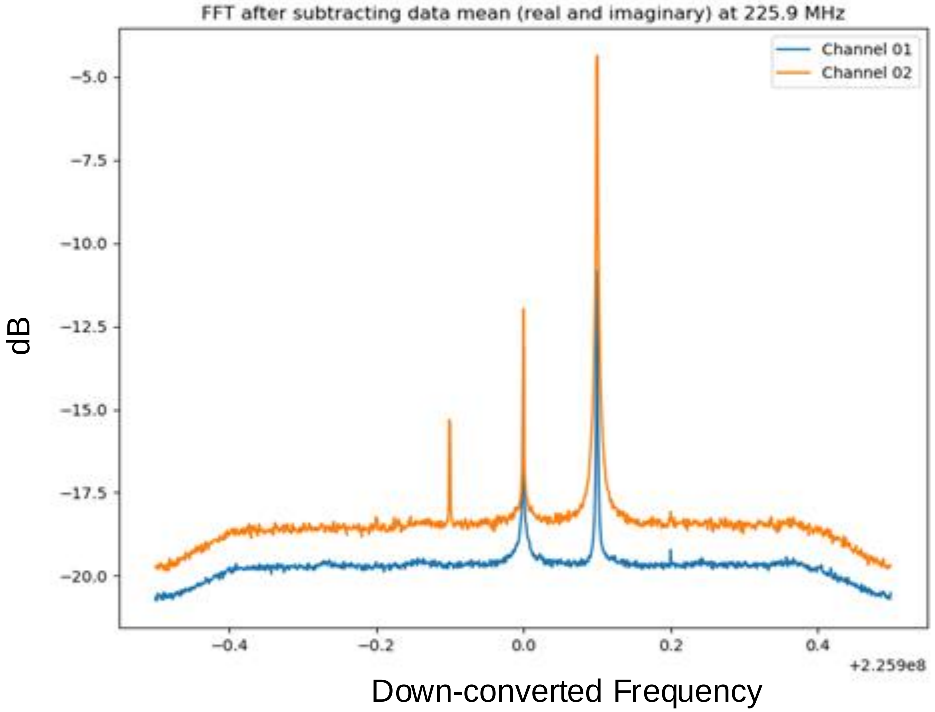
Preparation

- Learned how to transmit and receive from the LimeSDR with GNURadio
- However, dual-receive on GNURadio proved to be quite difficult, so we switched to SoapySDR









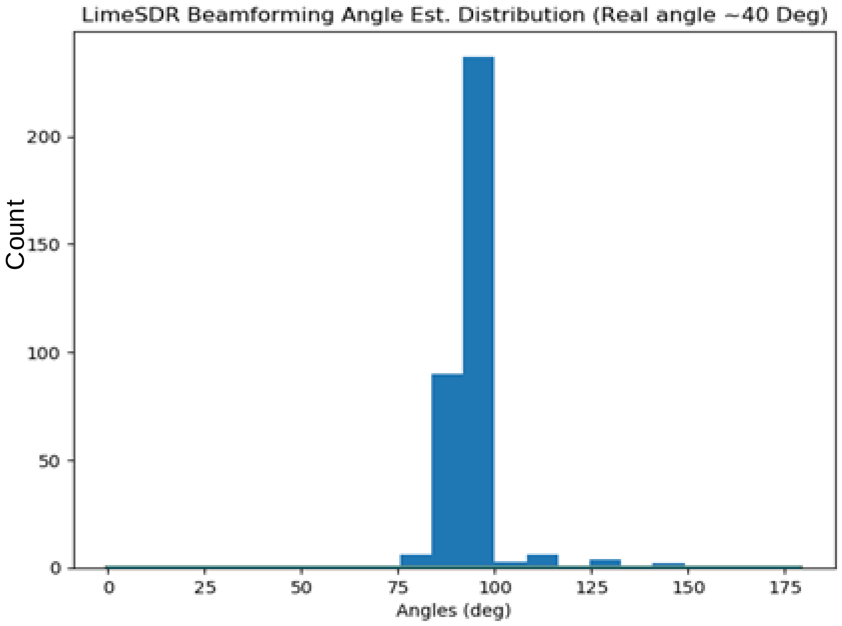
LimeSDR Beamforming

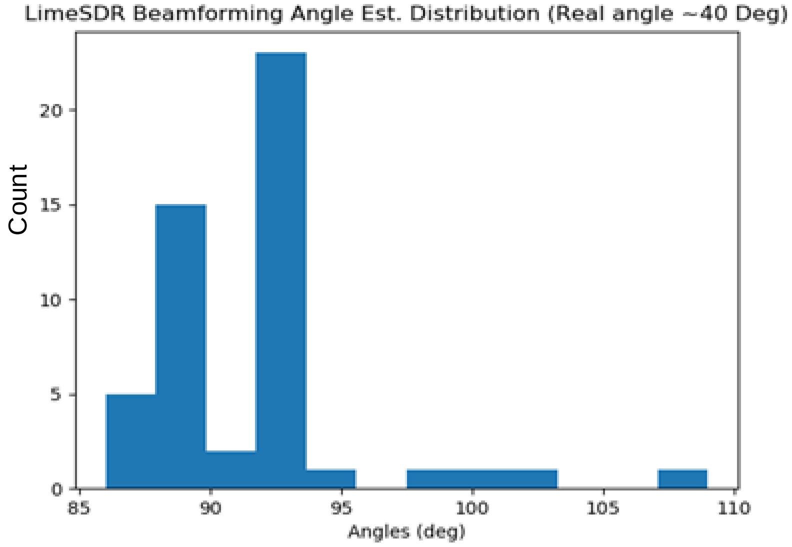
Side quest... failed?
Wait what happened?

Objectives:
☑Set-up LimeSDR
☑Set-up dual Rx
☑Collect data
☒Find AoA

Reward: confusion.
(and 100 exp.)

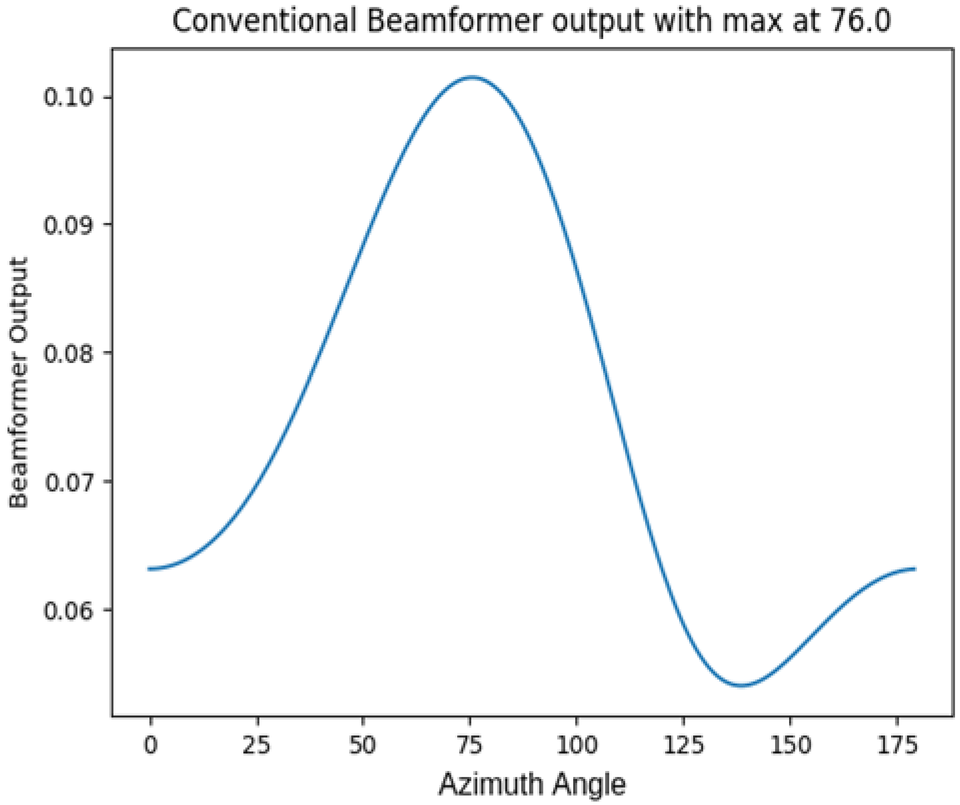
- AoA estimation histograms show serious problems
- With more testing, we found that the angle also changed with respect to sampling rate, so something must be wrong here



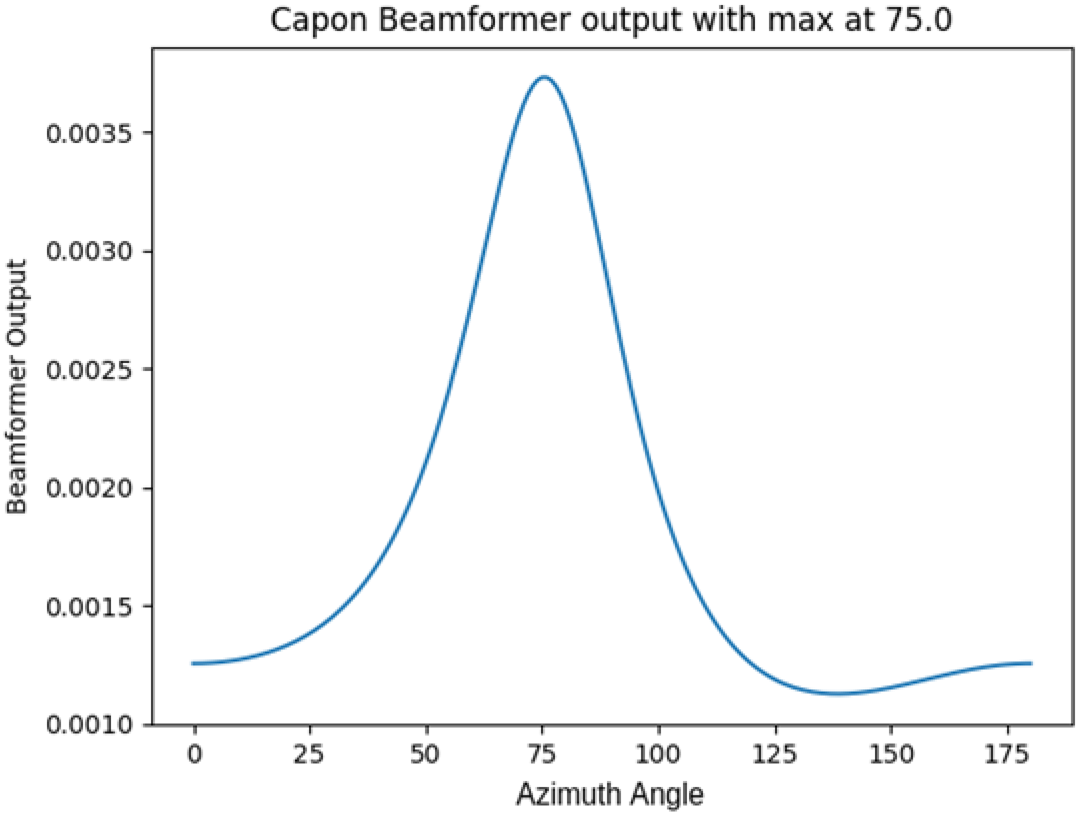


Beamforming Methods

$$Y = \sum_{n=1}^{N} w_n X_n$$



$$\frac{1}{\mathbf{w}^H \mathbf{R}_x \mathbf{w}}$$

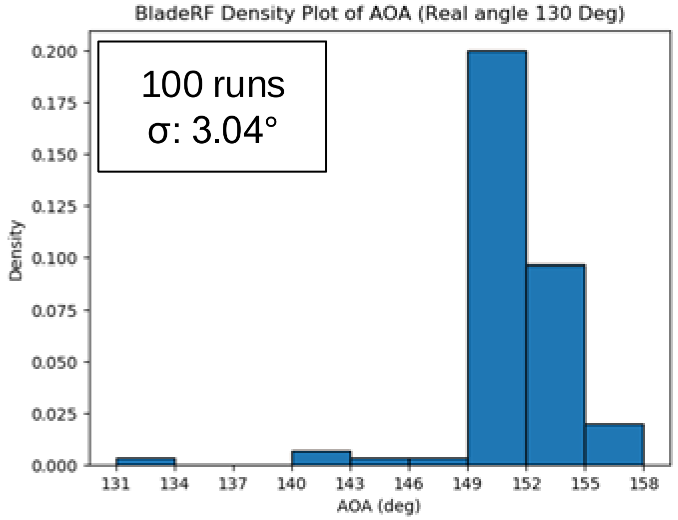


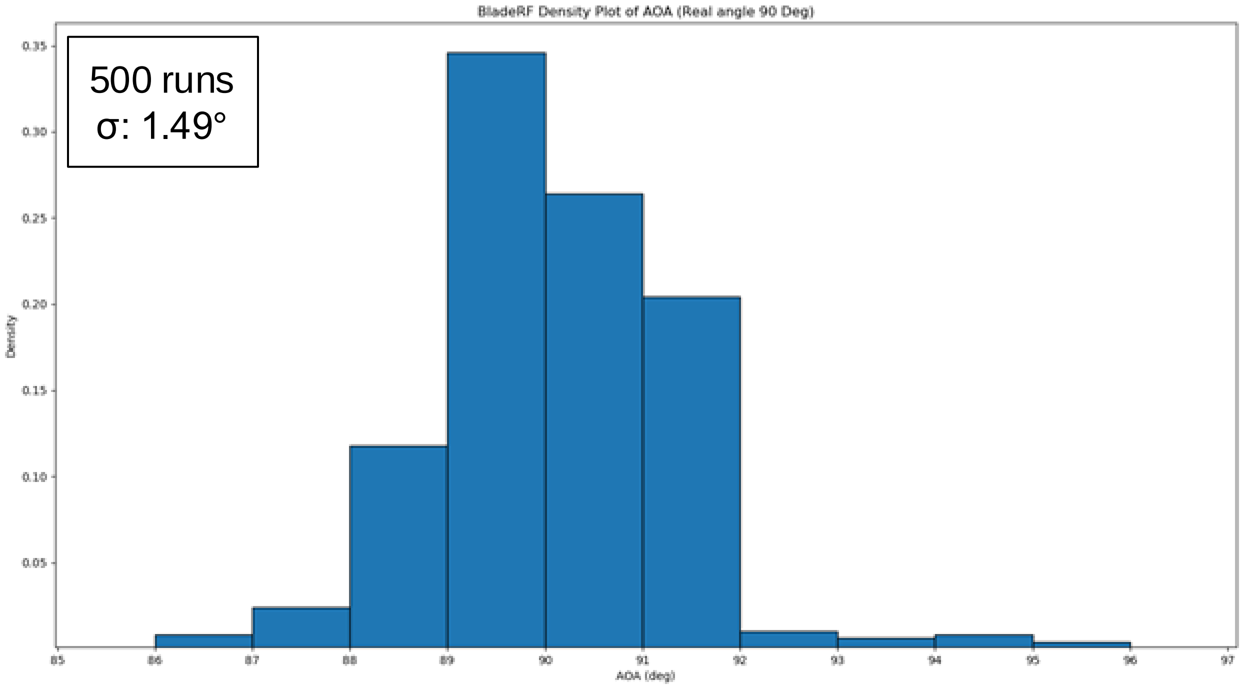
- Both beamforming approaches seem to agree, so the issue might lie somewhere else
- This made us think the LimeSDR may have been at fault

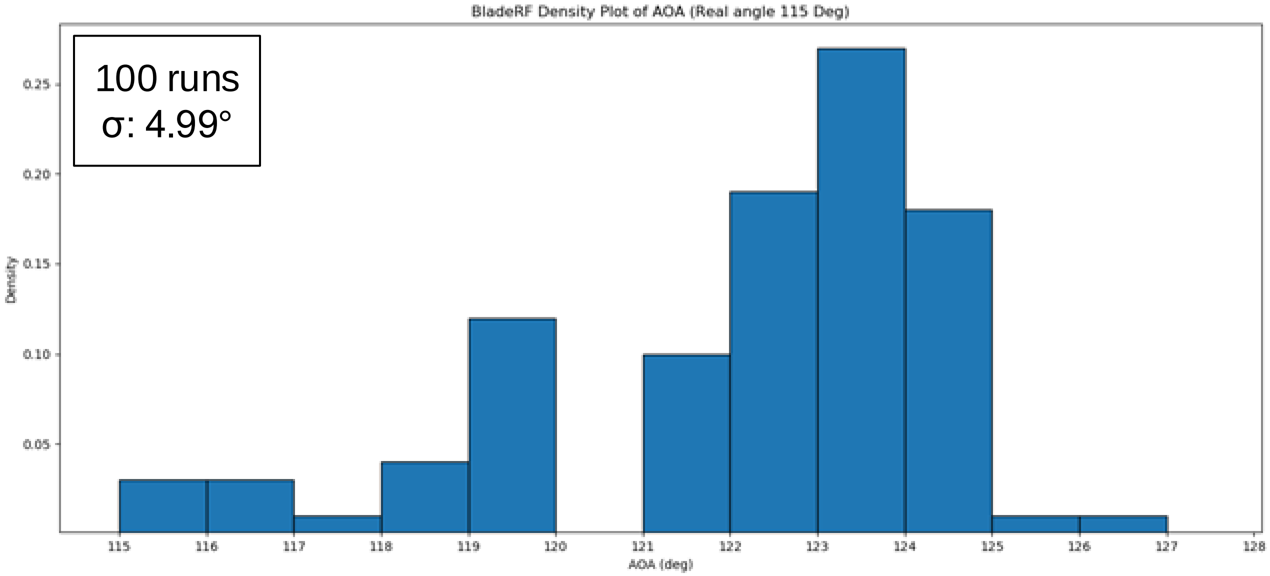
bladeRF Beamforming

- After some serious confusion, we got an extra (shielded) bladeRF to set up and experiment with and…

- Yup. It was the LimeSDR’s fault all along.



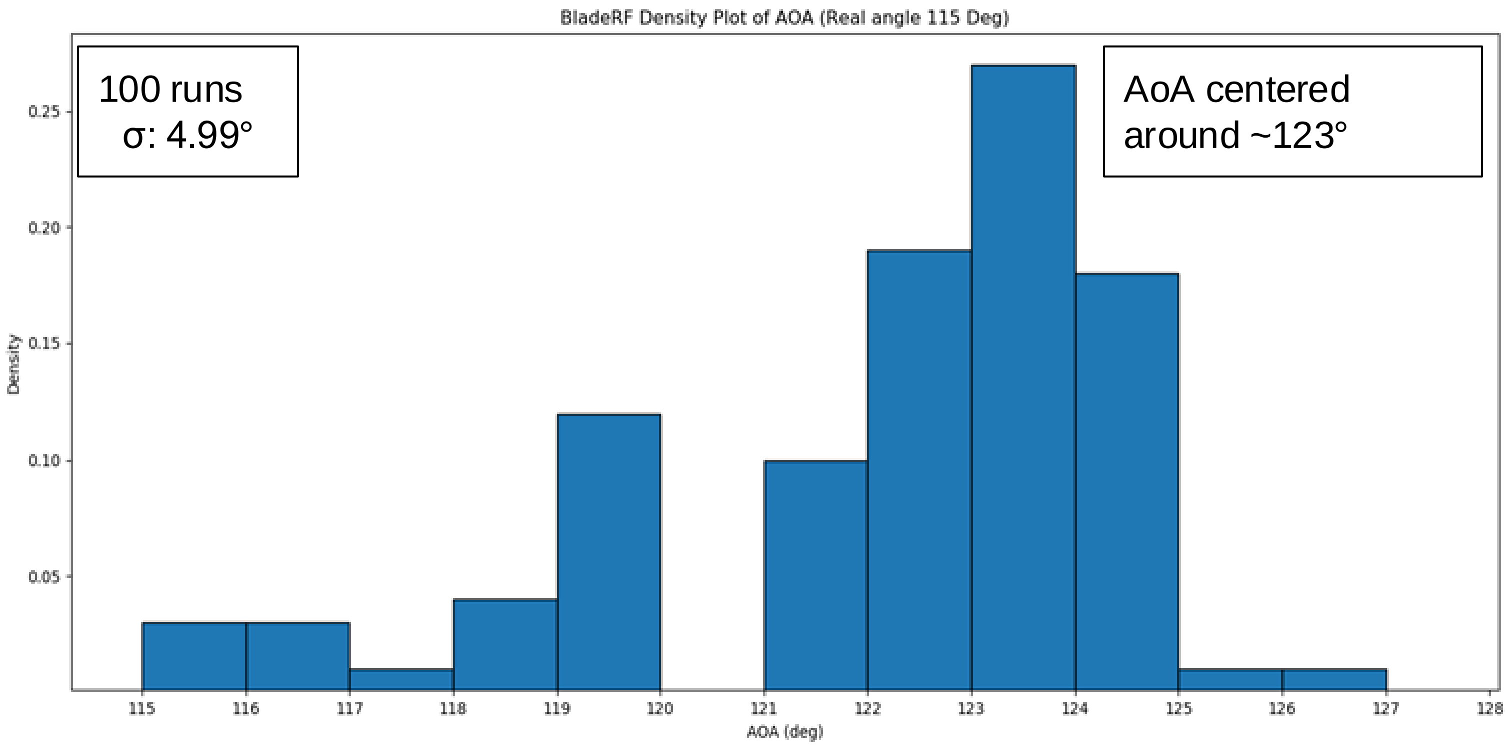




bladeRF Beamforming Issues

Sidequesting
Spencer + Jona

- However, not everything is perfect
- We are susceptible to multipath a nearfield
- This is expected given the context and equipment (e.g. only two RX channels)
- This means we're also bound to have similar issues on the RTL-SDRs

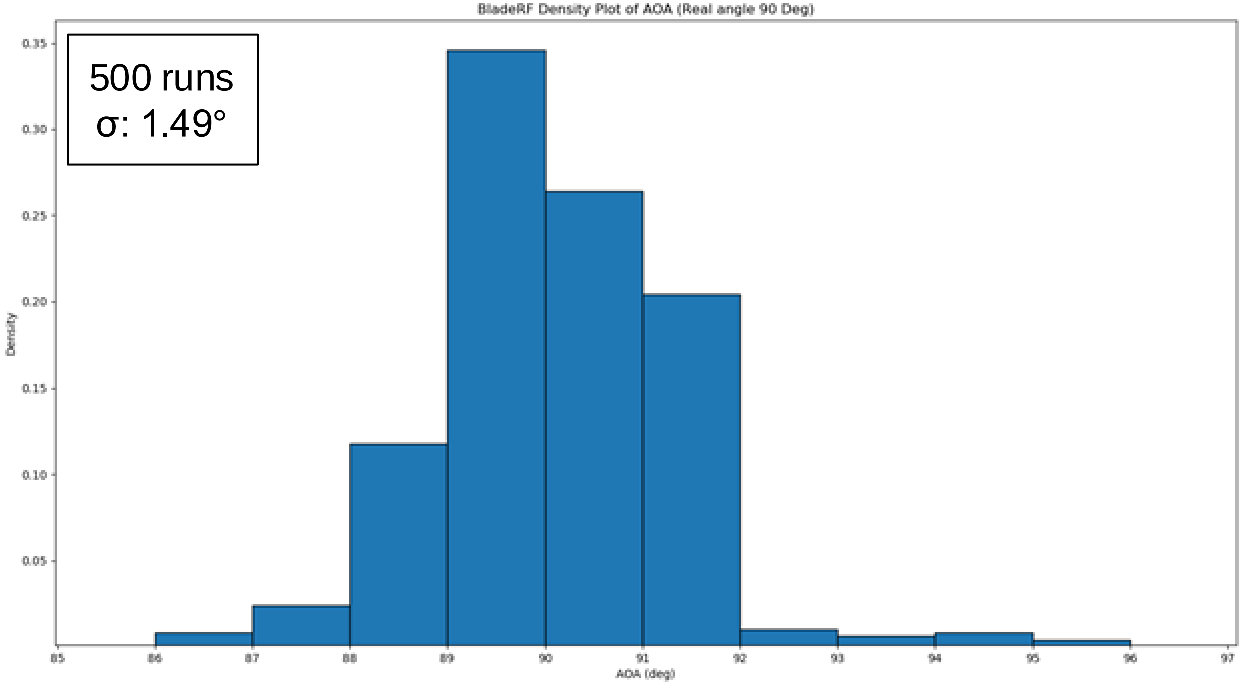


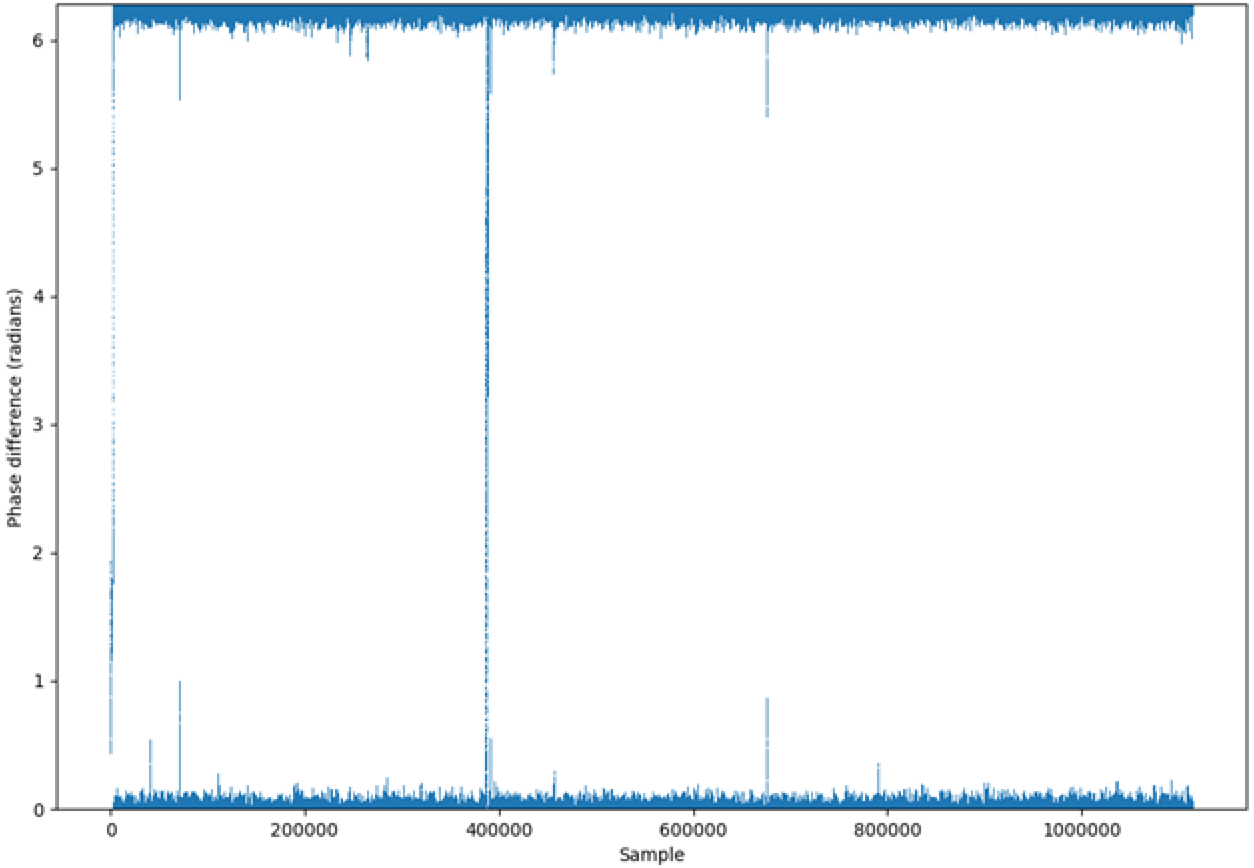
Presentation Outline

Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

Combining Prototypes

- Having verified our beamforming algorithm and perfected the calibration, we were able to beamform using the RTLs





Presentation Outline

Intro and Overview	RTL Calibration Nia	Comms System Jona	Beamforming Spencer	Sidequesting Spencer + Jona	Results Nia	Conclusion

Conclusion

What did we learn?

- Digital signal processing basics, RF basics, and other technical skills
- How to approach a large, complex problem that we are generally unfamiliar with

What would we do differently?

- More prototyping
  - LimeSDR/bladeRF sidequesting taught us a lot

Future work

- Fix correlation issues when we combine all parts of the project
- Extend to disaggregated array
- Self Localization