

# Wide Band Receiver Calibration

Nikhil Challa

May 30, 2023

# 1 Introduction

The object is to identify methods for calibration and perform calibration for multiple receiver system using an open source hardware. By calibration we mean, estimating the phase and amplitude changes to signal due RF signal conditioning units such as low-noise amplifiers, active bandpass filters and other components within the HW at transmitter and receiver for received signal. Once the HW contributions have been estimated, we can estimate the channel between the transmitter and receiver. We will be utilizing the below components for attempting calibration processes.

- Single Ettus USRP X410 unit ([1])
- Single omnidirectional antenna (Tx Antenna)
- Patch antenna with atleast 4 patches (Rx Antenna)
- RF cables to connect the USRP to the antennas
- 20db attenuators
- CPU for sample collection
- Ethernet cable to connect USRP to CPU

Since we are restricted to using patch antennas which are not omnidirectional, we will focus only on phase estimation and not amplitude. The reason for selecting patch antenna is covered in section 2. We will be utilizing 1 Transmit antenna (omnidirectional) and 4 receive antennas (patch antenna) to transmit/receive reference signals for the calibration process.

The following terms will be used interchangably

source : transmitter

sensor : receiver

Tx : Transmitter

Rx : Receiver

B2B : Wired Tx to Rx direct Transmission (Back To Back)

OTA : Wireless Tx to Rx Transmission (Over The Air)

## Sequence of steps:

- Determine layout requirement of Tx and Rx antennas
- Collect wireless transmission data between Tx and Rx and physical measurements
- Collect wired transmission data with direct connection between the Tx and Rx port of the USRP
- Perform frequency error estimation and frequency correction
- Determine stable points in the calibration data after phase unwrapping
- Fix errors in unwrapping using outlier detection
- Perform SAGE algorithm to extract LoS component (Time Delay and Phase+Amplitude of signal)
- Using the LoS signal phase to perform final calibration (and determine phase contributions from the receiver)
- Using MUSIC algorithm to verify the correctness of the calibration data

## 2 Layout Requirement of Tx and Rx antennas

It is important to understand the layout requirements for reliable calibration process. We begin by first identifying the reference signal properties and the physical dimensions.

- Center Frequency  $f_c$  : 5.725 Ghz
- Wavelength of source signal  $\lambda = \frac{c}{f_c} = \frac{3 \times 10^8}{5.725 \times 10^9} \approx 0.052$  meters
- Separation between sensors  $L_a$  : 0.2 meter

We also need to measure the Tx antenna base-plate dimensions, but will be covered in next section. We now need to determine the antenna height, Tx-Rx separation and other such requirements

We consider the assumptions listed in Appendix A, that will be factored in for layout requirement. It is recommended to follow through the details provided in the Appendix to recognize the reason for these assumptions.

- Assumption 4 requires us to use sensors in a single array, hence we use the patch antenna as seen in Figure 1a. Also utilizing the patch antenna (instead of single Rx) gives us the possibility of using MUSIC algorithm to verify calibration data accuracy, covered in section 9.
- Assumption 1 requires us to ensure the source and sensor is in the same plane. For this arrangement to be met, we ensure that the Tx and Rx are placed on the table with equal height. How do you measure height for Tx side. Its not a single point. For Rx side, the patch center is the height I assume
- Assumption 2 requires us to ensure the separation between the source and sensor are such that we can utilize the far field approximation. Referring to the equation provided in Appendix B, the Tx-Rx separation or Rayleigh distance must be greater than 1.54 meters.

$$\text{Rayleigh distance } d_R = \frac{2L_a^2}{\lambda} = \frac{2(0.2)^2}{0.052} \approx 1.54 \text{ meters}$$

- Assumption 3 requires no-mutual coupling between the receivers. The USRP X410 unit is assumed to have decent shielding (negligible coupling) between the different receivers.
- Assumption 6 requires us to ensure a clear LoS path. The quality of the LoS signal also depends on the Fresnel Zone clearance. Referring to the equation provided in Appendix C, the distance from the floor to the LoS path or the Fresnel Zone radius must be greater than 0.15 meters.

$$\text{Radius of first Fresnel Zone } F_1 = 8.656 \sqrt{\frac{d_R \times 10^6}{f_c}} = 8.656 \sqrt{\frac{1.629 \times 10^6}{5.725 \times 10^9}} \approx 0.15 \text{ meters}$$

The LoS requirement implies we need to separate out the LoS component from the multipath using the SAGE algorithm. More details are provided in section 7



(a) Patch Antenna

(b) Cabling from patch to USRP

Figure 1: Path Antenna Setup utilizing middle 4 patches

For our setup as seen in Figure 2, the Tx bench to Rx bench separation was set to 1.629 meters and the height from floor to Tx/Rx bench level is 0.5 meters. The Tx-Rx separation is greater than bench separation, and the LoS height is greater than the bench height. This ensures both Rayleigh distance ( $> 1.54m$ ) and Fresnel Zero radius ( $> 0.15m$ ) are met.

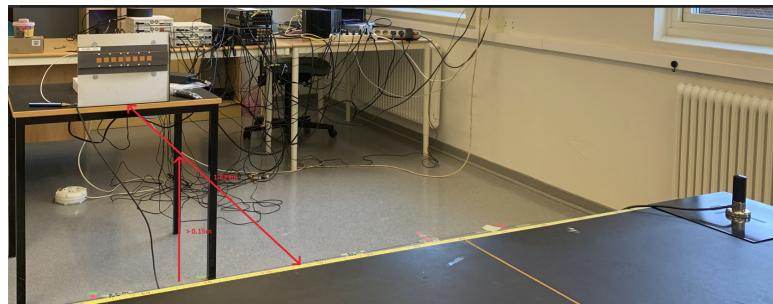


Figure 2: Measurements to check layout requirements

Apart from the critical aspects, there are some non-critical considerations that will help with calibration process. When we do unwrapping of the phase for visualization and error spotting, it will be useful to ensure that the propagation path distance for adjacent positions does not exceed 0.9 wavelength (or  $1.8\pi$  in radians). Based on the adjacent position phase value, we can know if unwrapping is required or not. The other aspect is to ensure we have a minimum difference of atleast 0.1 wavelength (or  $0.2\pi$  in radians) to reduce impact of human error on calibration data. The wavelegnth is approximately 5.24cm, and we expect the error to not exceed 0.5 cm, hence the 10% margins.

Reference Code : ArrangementCalculator.m

[Room for Improvement 1] While deciding the arrangement for calibration data, maximum difference was missed resulting in higher error at the extremes. We opted for 10cm sliding length, when the calculations suggest value higher than 13cm; but fortunately we dont see huge errors near the mid. Given that we use 10cm as sliding length, the max length on either sides of the table calculates to be 104cm. We limited ourselves to 100cm but this is also too close to the upper limit. If we opted for 13cm sliding length, we will be limited to 77cm limits on either side of the mid point.

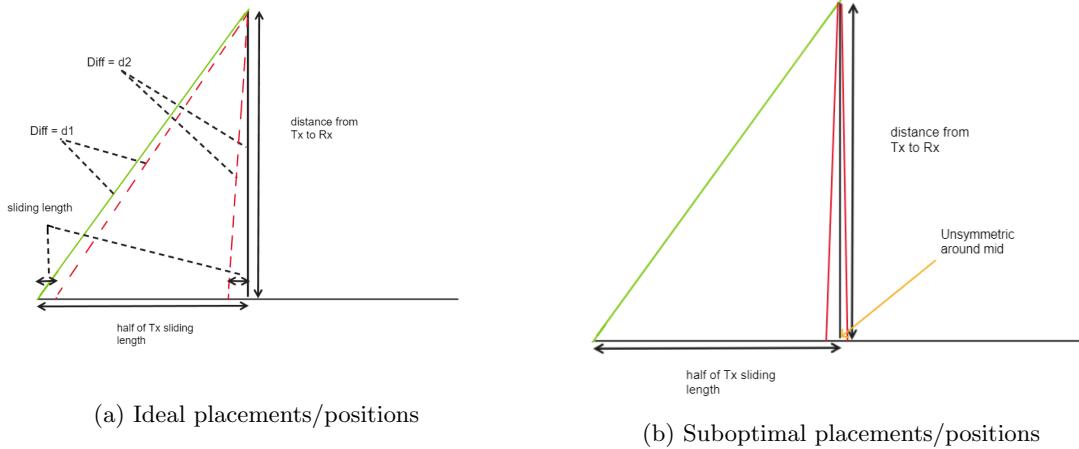


Figure 3: Visual understanding of min/max difference  $d_2/d_1$

Another useful placement strategy is to have symmetric placements. This will all make more sense when you cover subsequent sections.

### 3 Data Collection

#### 3.1 Tx-Rx Antenna setup

Having identified the apparatus setup requirements, now we attempt to collect data to perform the calibration. Collecting data for a single position of Tx and Rx antenna wont give us confidence on accuracy of calibration results nor will it give us calibration data for wide angle of operation. Rather we can attempt to collect multiple data points and ensure we see similar calibration results for all data points. For this we will attempt to slide the Tx antenna in parallel to patch antenna board orientation and collect the data at periodic position intervals. Arrange a couple of benches as shown in Figure 1b or Figure 4. Ensure that the center of the row of benches is in same position as the center of the patch antenna; to obtain equal measurements on either sides. Note down the measurement from one end to the center of the bench row setup as shown in Figure 4 ( 106cm)

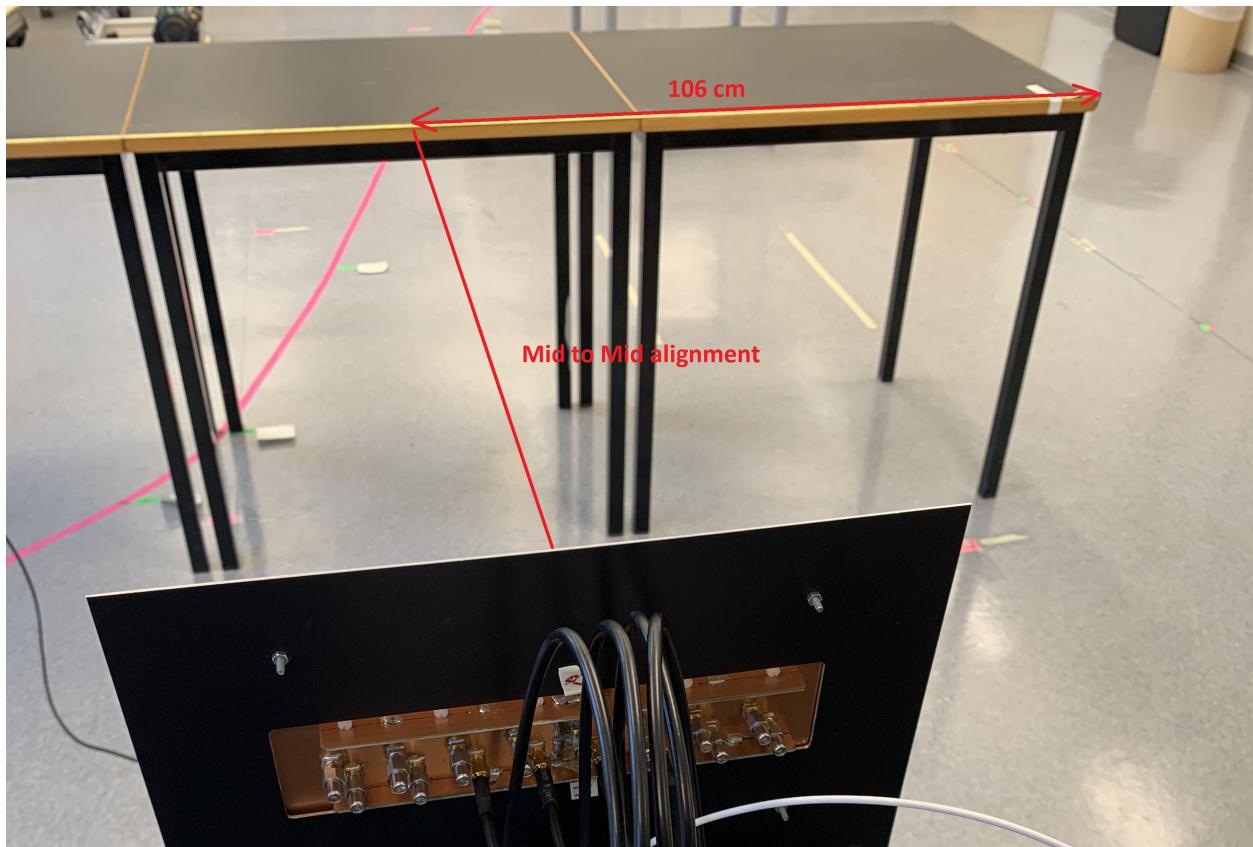


Figure 4: Mid point measurement

Once you have the bench row arrangement completed, we measure the Tx antenna bottom plate physical dimensions. The measurement value are shown in Figure 5. We need these measurements as propagation path distance is from Tx to Rx, not from bench to bench.

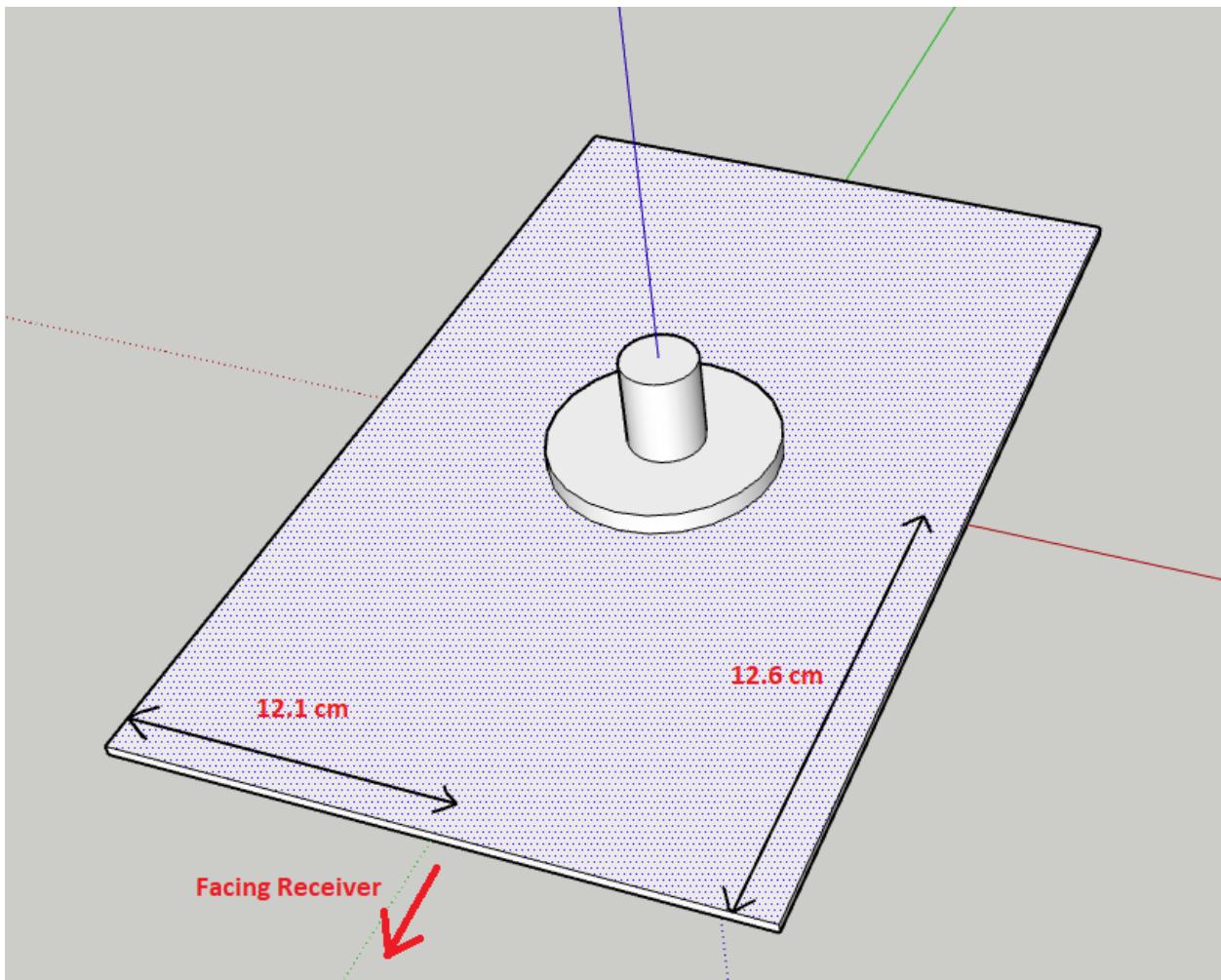


Figure 5: Tx antenna on metal plate

We will utilize the sliding approach to collect multiple data points, as shown in Figure 6. For this, we mark from the left to right, a 10cm interval and attempt to collect a set of measurements for every 10 cm shift in Tx antenna. The more the measurements, the more datapoints we have, the more is our confidence on the calibration data.

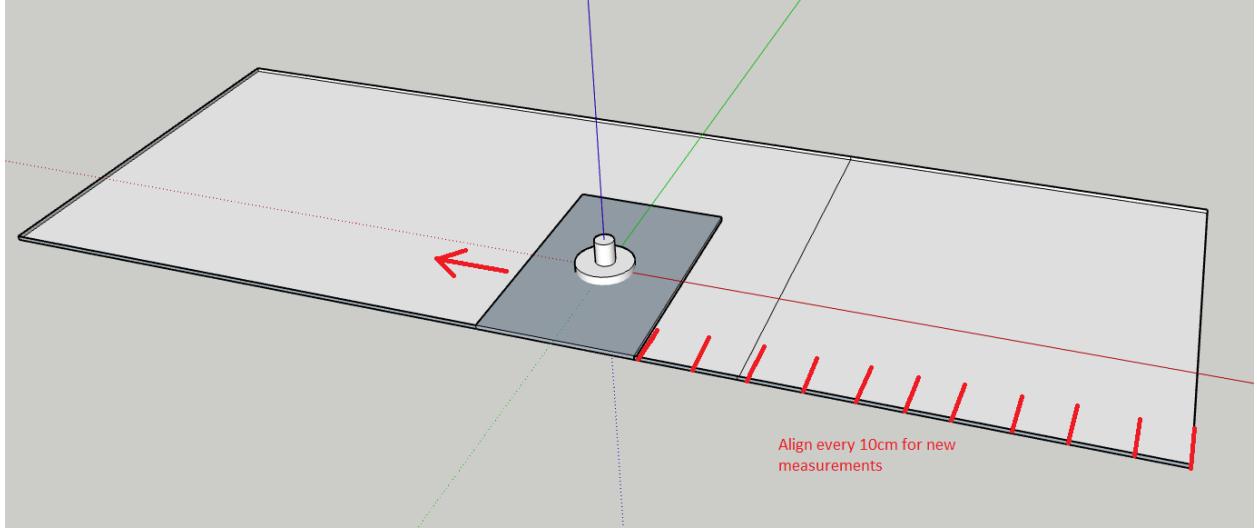


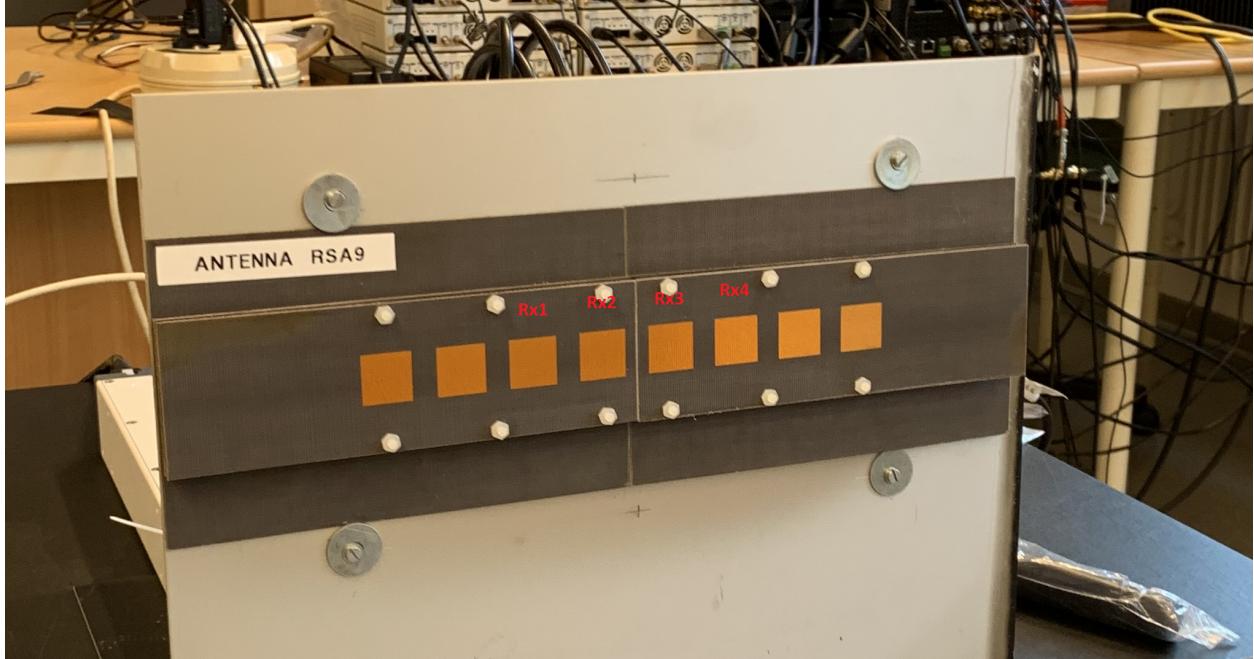
Figure 6: Sliding Tx Antenna

[Room for Improvement 2] One important consideration (to ensure lower error in the calibration data around mid) is while placing the Tx antenna at various positions, we have an absolute mid position also. Having symmetry in placement gives us pair of observations/measurement data on either side of mid position; to perform more corrective action if necessary. You can see from Table 1, for Sr No 10, 11 the mid value of DtMFN was not as close to 0 as possible. Also we should take odd number positions in total (mid + equal positions on either sides)

[Room for Improvement 3] Continuing from previous point, selecting numbers that don't result in non-terminating decimal numbers (due to division) will result in more accurate estimation results (avoid rounding the number). Selecting number of stable points as 11 (5 on each side and 1 mid point) is useful. Other good numbers are 9 ( $4 \times 2 + 1$ ), 5 ( $2 \times 2 + 1$ ), 17 ( $8 \times 2 + 1$ ), 21 ( $10 \times 2 + 1$ ) and many more but also keep in mind there is an upper limit on the number of points (to ensure we meet "Room for Improvement 1" conditions). All these numbers (number of points on each side) are not multiples of 3, 7, 11 or other such numbers as they can result in non-terminating decimals. I will call these as "mathematically friendly" numbers, for reference in other parts of this document

### 3.2 Cabling details

Now we turn our attention to rf cable connections between USRP and Tx-Rx antennas. Refer to Figure 7 for the details. The Tx port will be connected directly to the omnidirectional antenna.



(a) Selecting patch antenna



(b) USRP Connections

Figure 7: Patch antenna to USRP (Port numbers marked in red)

### 3.3 USRP Configuration

Center frequency : 5.725 Ghz

Waveform Type : OFDM (Appendix E )

Reference Signal : Zadoff Chu sequence of length 813 (Appendix D)

Subcarrier spacing : **FILL**

Number of subcarriers : 1024

Sampling rate : 2.519ns

Tx gain : 55 dB

Rx gain : 45 dB

We will use the above configuration to collect OTA signal (Over The Air) as well as direct Tx to Rx feed for BB data (Back-to-Back). To collect B2B data, refer to Figure 8 as an example for Tx to Rx3 case. We need to collect the same for Tx to Rx0, Tx to Rx1 and Tx to Rx2.

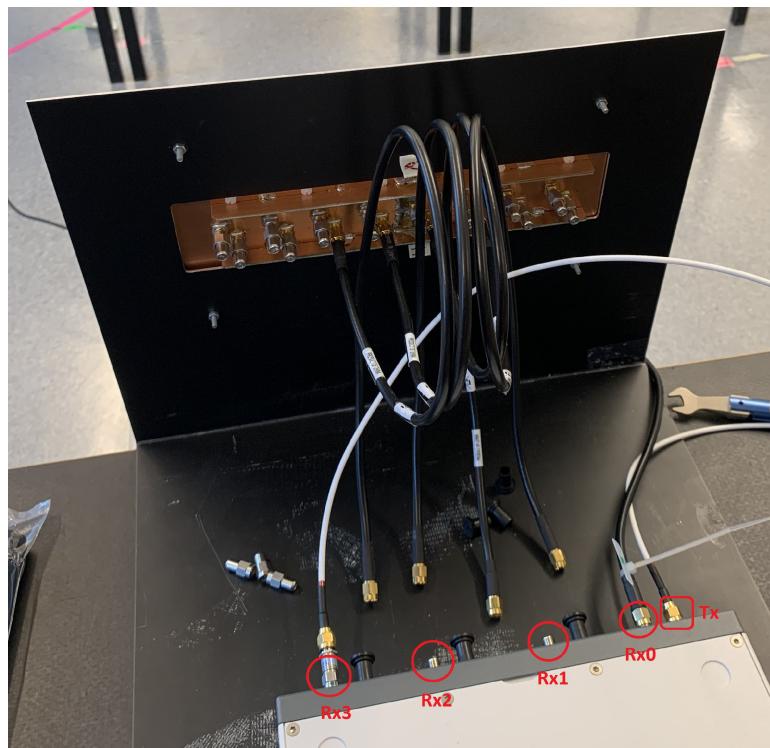


Figure 8: B2B Data collection (Tx to Rx3 connection)

### **3.4 Table of positional data**

Using the sliding approach (Figure 6), we collect all the necessary information shown in Table 1.

#### **Patch Antenna and Tx Antenna bottom plate info :**

Distance from left of Tx Antenna plate to Tx Antenna = 12.1 cm

Distance from front of Tx Antenna plate to Tx Antenna = 12.6 cm

Distance from left of Table to Center = 106cm

Shortest Distance from the center of Rx panel to Tables = 162.9cm

Shortest Distance from the center of Rx panel to Tx Antenna =  $162.9 + 12.6 = 175.5$  cm

Distance between each patch in Rx Patch Antenna = 2.6 cm

#### **Acronyms :**

DfL : Distance from Left of the tables to tx antenna bottom plate

DtAfL : Distance to AntennaTx from Left of the tables

DtMfN : Distance to Mid from Tx Antenna position (+ve if Tx Antenna location to left of Mid else -ve)

DtMoR : Distance to Mid of Receiver (Hypotenuse) from Tx Antenna

DoA : Direction of Arrival w.r.t direction  $\perp$  to patch antenna plane

Dt1P : Distance to first Rx Patchh Antenna from Tx Antenna

Dt2P : Distance to second Rx Patch Antenna from Tx Antenna

Dt3P : Distance to third Rx Patch Antenna from Tx Antenna

Dt4P : Distance to forth Rx Patch Antenna from Tx Antenna

SrNo	DfL (cm)	DtAfl (cm)	DtMfn (cm)	DtMoR (cm)	DoA (radian)	DoA (degree)	Dt1P (cm)	Dt2P (cm)	Dt3P (cm)	Dt4P (cm)
1	0	12.1	93.9	199.0414	0.9781	-33.959	197.2315	198.4314	199.6579	200.9107
2	10	22.1	83.9	194.5237	1.0526	-29.6904	192.8737	193.9665	195.0879	196.2373
3	20	32.1	73.9	190.4244	1.1229	-25.6625	188.9451	189.9237	190.9327	191.9716
4	30	42.1	63.9	186.7711	1.1897	-21.8352	185.473	186.3304	187.2199	188.1411
5	40	52.1	53.9	183.5905	1.2536	-18.174	182.4836	183.213	183.9763	184.7731
6	50	62.1	43.9	180.9073	1.3153	-14.6388	180.0007	180.5963	181.2272	181.8931
7	60	72.1	33.9	178.7441	1.3752	-11.2068	178.0456	178.5021	178.9952	179.5246
8	70	82.1	23.9	177.1199	1.4338	-7.8492	176.6359	176.9492	177.3	177.6882
9	80	92.1	13.9	176.0496	1.4914	-4.549	175.7847	175.9517	176.157	176.4004
10	90	102.1	3.9	175.5433	1.5486	-1.2717	175.5	175.5193	175.577	175.6732
11	100	112.1	-6.1	175.606	1.6056	1.9942	175.7847	175.6559	175.5656	175.5138
12	110	122.1	-16.1	176.2369	1.6628	5.2715	176.6359	176.3605	176.1229	175.9235
13	120	132.1	-26.1	177.4302	1.7206	8.5832	178.0456	177.626	177.2436	176.8985
14	130	142.1	-36.1	179.1744	1.7794	11.9522	180.0007	179.4408	178.917	178.4295
15	140	152.1	-46.1	181.4537	1.8397	15.4071	182.4836	181.7884	181.1278	180.5023
16	150	162.1	-56.1	184.2484	1.9018	18.9652	185.473	184.6483	183.8567	183.0986
17	160	172.1	-66.1	187.5352	1.9663	22.6608	188.9451	187.9974	187.081	186.1964
18	170	182.1	-76.1	191.2889	2.0338	26.5283	192.8737	191.8098	190.7755	189.7712
19	180	192.1	-86.1	195.4826	2.105	30.6077	197.2315	196.0587	194.9135	193.7965
20	190	202.1	-96.1	200.0886	2.1806	34.9393	201.9907	200.7162	199.4675	198.245

Table 1: Antenna position details

Full table of details can be referred to at DATA SHEET

Few comments :

- The calculations are limited to 4 decimal places
- The Excel sheet in Git as more columns, only first 11 columns are posted above due to size restrictions
- The 35th column as configurable variables, do not modify any other contents as they auto populate based on configurable variables.

## 4 Frequency Error Detection and Correction

Reference Code : C1\_FrequencyErrorAndCorrection.m

It is always good to check if we have frequency errors prior to calibration process. Frequency error can be easily identified by checking the phase difference between consecutive time samples. If we observe a linear increment or decrement in phase difference, then the contribution to this error in phase is the frequency error. The error from each channel using B2B Data is shown in Figure 9

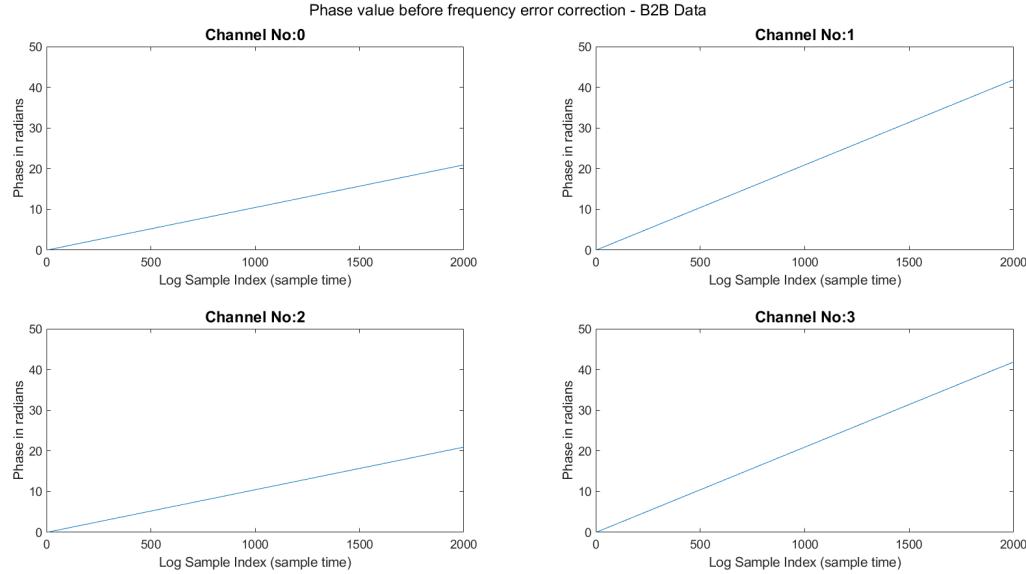


Figure 9: Phase error before frequency error removal

Notice that phase is usually measured within the range of  $[-\pi \ \pi]$  or  $[0 \ 2\pi]$ . We used a concept of phase unwrapping to see a clear increment and plot beyond the limits if the frequency error is high. Phase unwrapping makes it easier to calculate a single slope value for each Channel. The longer is the data sample duration, the more accurate is the slope value, as we do line fitting.

We first determine the slope of the line which can be easily computed using *polyfit* function from matlab. Using the slope value and knowing the sampling time between the points, we can negate the phase error by reversing the effect. The phase error after removing frequency error is shown in Figure 10

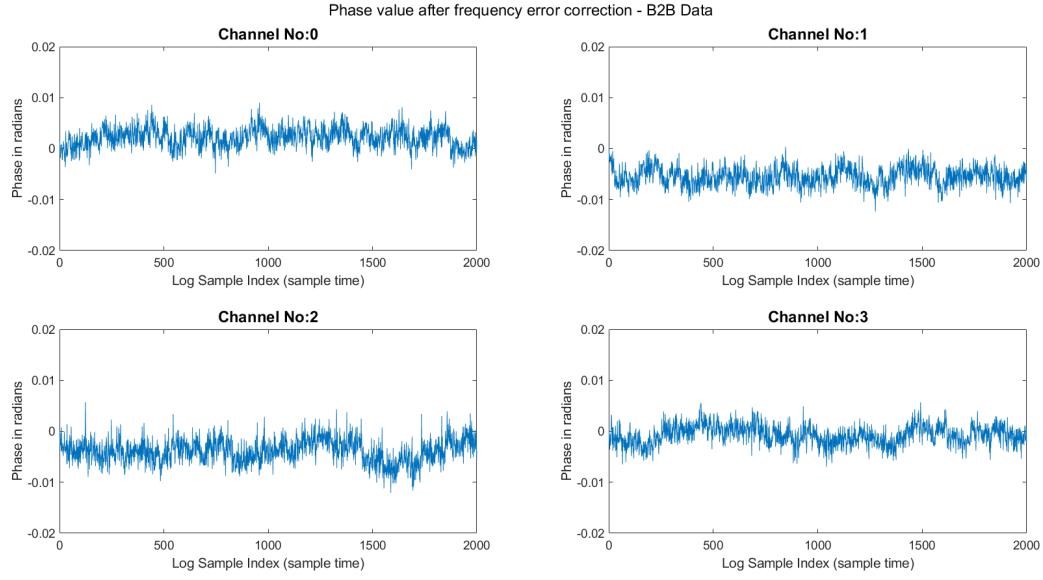


Figure 10: Phase error after frequency error removal

As we can see, the phase error has now dropped to max of  $10^{-2}$  and has stochastic behavior over time, concluding that frequency error is removed. We need to remove this frequency error for OTA Data also.

## 5 Stable stationary point selection for Channel Estimation

Reference Code : C2.KNN\_Clustering.m

While sliding the Tx antenna, the received signal will observe some phase jitter either due to physical obstruction, or actual antenna movement. It takes few secs to stabilize after completing the antenna placement hence selecting the stable points for calibration at every step is necessary. The Figure 11 gives a good view of phase jitters.

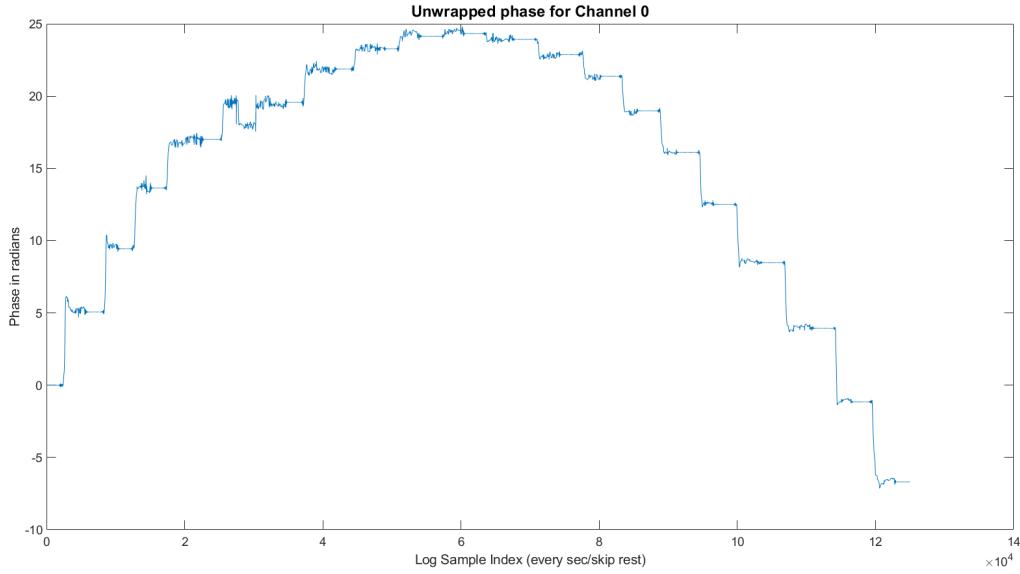


Figure 11: Sample of Phase Jitter

The stable points can either be chosen manually through visual inspection or you can use a customized KNN Clustering algorithm. This algorithm requires you to provide the percentage of time the antenna was in a stationary position with no interference/obstruction and the number of stable points. This corresponds to variable name *RatioStableVsMovement* and *NumCenters* in the Reference Code provided earlier. Below is the summary but you can refer to Appendix F for more details.

Summary of Algorithm :

- Provide two values, Number of stable points (eg 20) and Percentage of time in stable region (eg 0.5). The "Acceptance Rate Limit" should be less than percentage of time in stable region.
- Introduce the concept of window, which is 1 or more secs of consecutive ofdm samples. Window width is in multiples of 1 sec. Each sec has 200 sample points in our dataset.
- Start with a window width, utilize sliding window (minimum slide duration = 1sec) to pick different set of ofdm samples for each window (window sample).
- Calculate the mean and variance of the phase value of all ofdm samples in a window (mean value, variance value)
- Perform KNN Clustering using the mean value of all window samples and number of centroids equal to number of stable points. (This needs to be modified for symmetric positioning)

- Reject bad window samples based on variance of all the window samples. Higher the variance, higher is instability of points in window, worst is the window.
- Increase strictness of sample quality (eg using sum of points to centroid distance or sumd) to meeting Acceptance Rate Limit. Higher strictness will reject more samples in previous step.
- Try multiple window width and pick a window width based on highest number of surviving samples but meeting "Acceptance Rate Limit"
- Select the nearest sliding window index to the centroid (stable window index)
- Repeat for all channels and take average index value as final stable window index (number of indexes equal to number of stable points equal to number of clusters)
- Select the mid ofdm sample for all stable window indexes

The locations of the stable positions picked up by the above algorithm can be seen in Figure 12

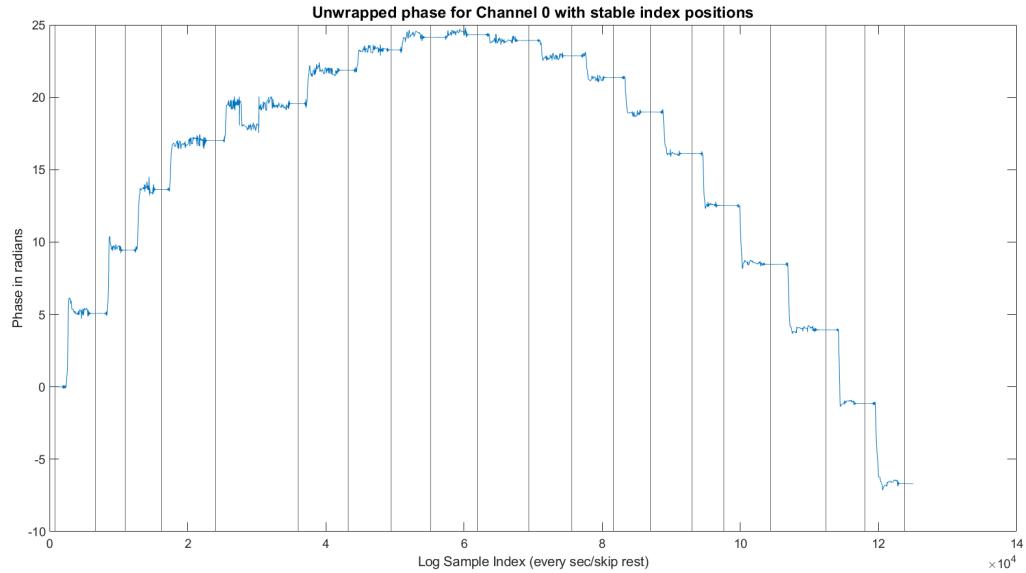


Figure 12: Vertical plot of Stable Positions

## 6 Unwrapping error removal using outlier detection

Reference Code : C3\_OutlierDetectorAndPhaseCorrector.m

This process is not necessary to complete the calibration process, but helps with obtaining the right unwrapped phase for visual analysis and basic error spotting. The more the number of receive antennas, the better is the error removal process. Figure 13 shows a good example of unwrapping error in Channel 3. The blue indicates wrapped phase and orange points indicates unwrapped phase. Carefully observing the overlapping instance to non-overlapping instance of red and blue points near Log Sample Index  $3.028 \times 10^4$ , you will see a jump of more than  $\pi$  which makes it difficult to know if there was phase wrapping or not. For some reason, a smooth transition was missing with huge jumps in phase only for Channel3. Observing other channels gives clue on what the unwrapped phase should be and detect outliers.

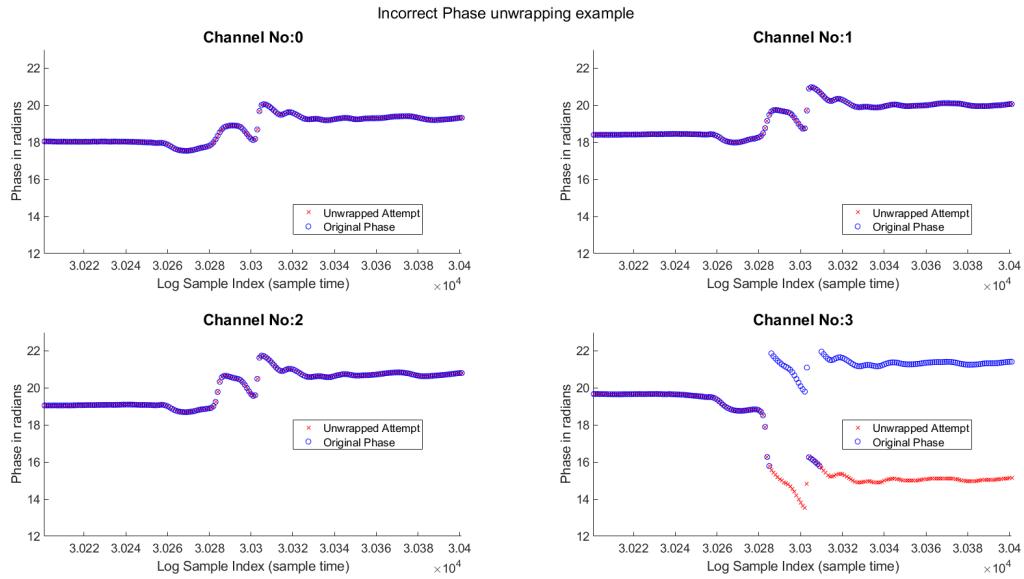


Figure 13: Unwrapping Error Example (Channel3)

The one advantage of using multiple receive antennas simultaneously; it becomes easier to detect outliers. Figure 14 shows (within green box) the error with phase unwrapping, and using information from other channels, we successfully correct the phase unwrapping error.

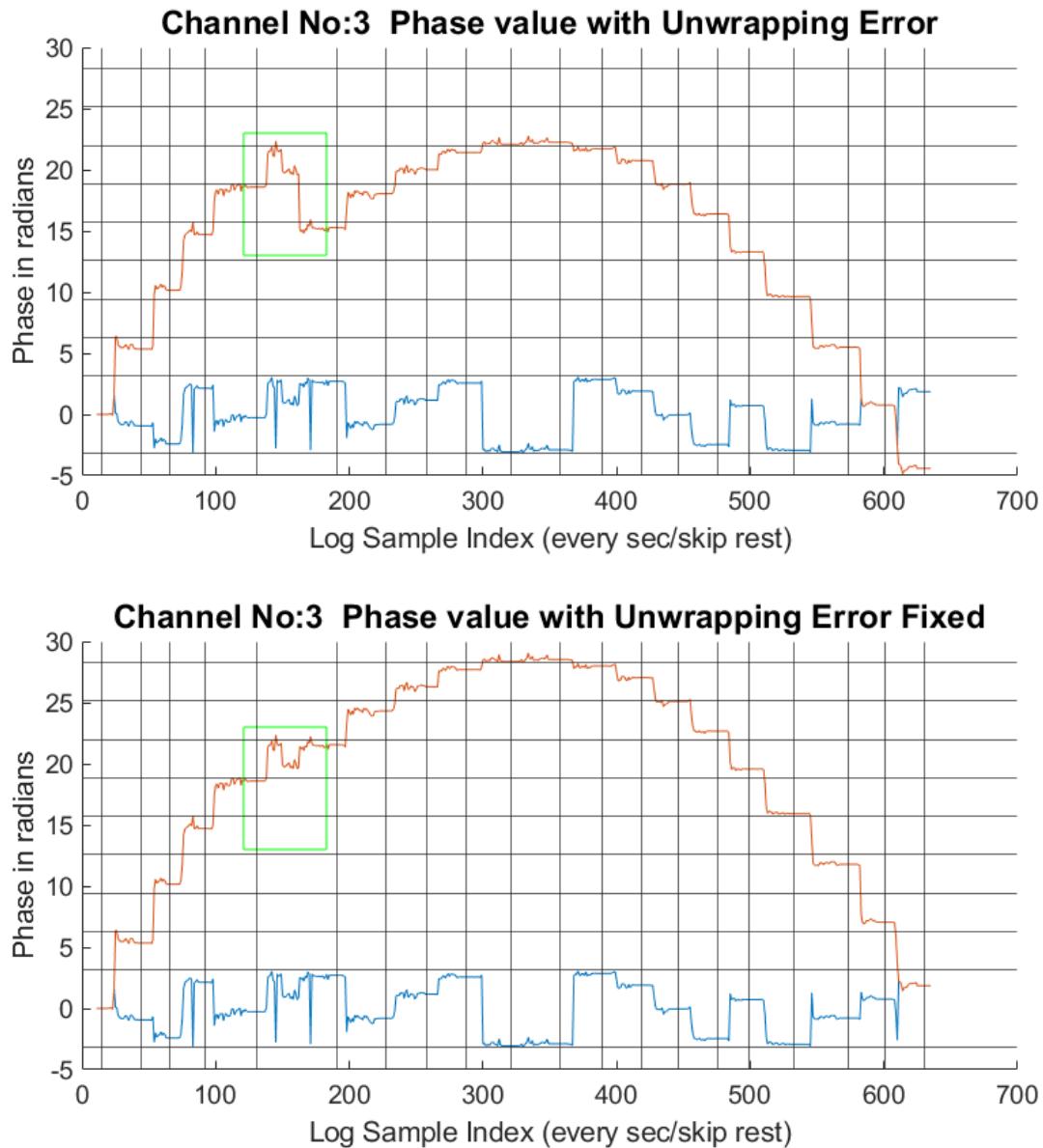


Figure 14: Unwrapping Error with focus on Channel3

To get a better overall view, provided all channels before and after correction side by side in Figure 15 and 16 respectively.

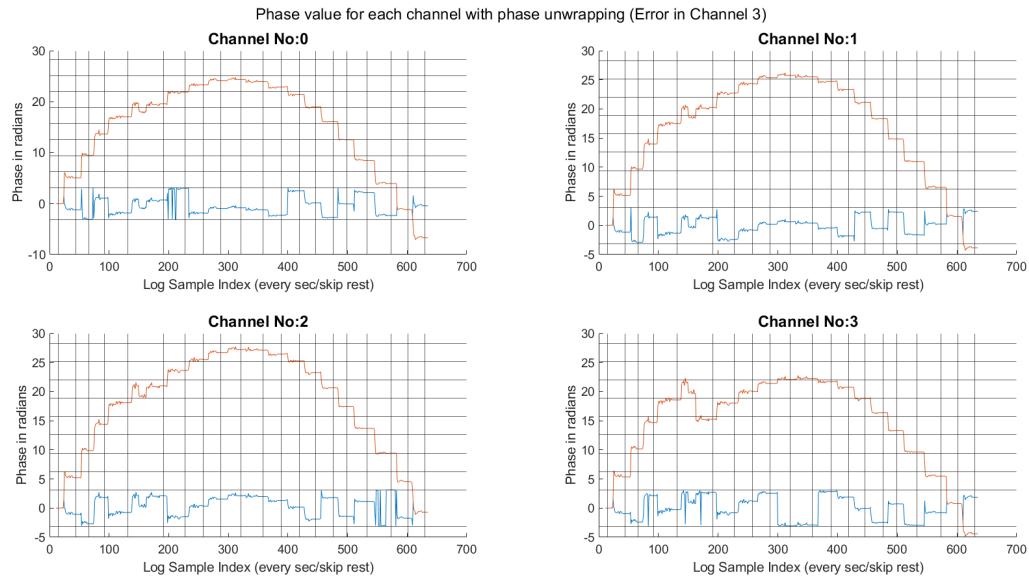


Figure 15: Unwapping Error in Channel3

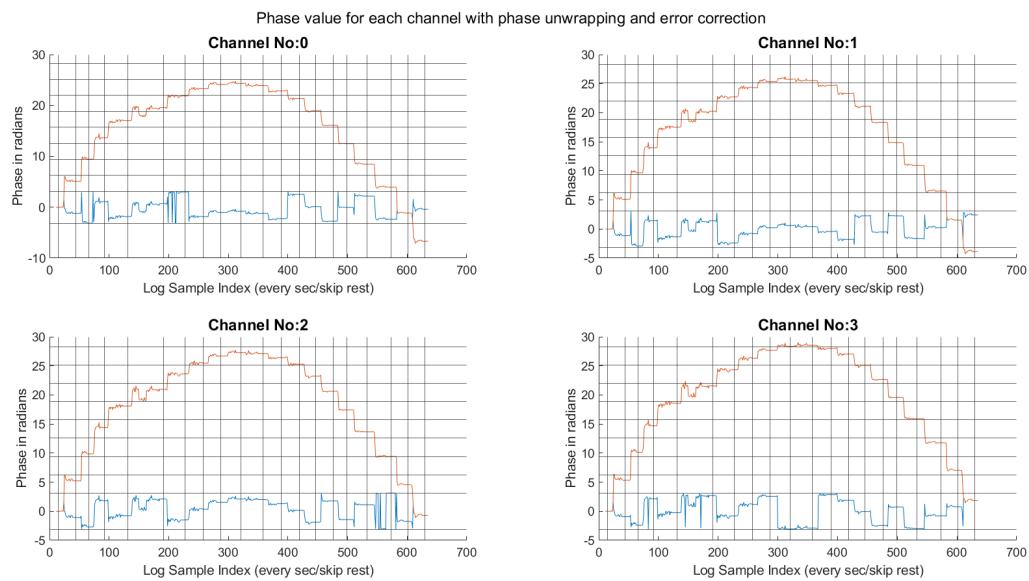


Figure 16: Unwrapping Error Fixed

## 7 LoS Separation

As indicated by Assumption 6, we need the LoS signal characteristics to perform calibration. We start with describing the signal model followed by the SAGE algorithm.

### 7.1 Signal Model

Lets denote the time-varying channel transfer function as  $h(k; t)$  where  $k \in [1, \dots, 1024]$  indicates the indexes of the subcarriers/tones.  $h$  is measured as

$$h(k, t) = \frac{y(k; t)}{x(k)} \quad (1)$$

where  $y$  and  $x$  denote the received and transmitted signals respectively in the frequency domain. The time varying channel impulse response (CIR)  $h(\tau; t)$  is then calculated by performing inverse Fourier transform to  $h(k; t)$

In an indoor environment, the receiver is expected to receive multipath components from the source emitting wireless signals. Each multipath component will consist of a delay, and a complex amplitude; and we can represent the signal model with time varying multipath components as

$$h(\tau; t) = \sum_{\ell=1}^{L(t)} \alpha_{\ell}(t) \delta(\tau - \tau_{\ell}(t)) \quad (2)$$

$h(\tau; t)$  : Time varying channel impulse response

$\alpha_{\ell}(t)$  : time varying complex amplitude

$\delta(\tau - \tau_{\ell}(t))$  : time varying delay of the  $\ell$ th multipath component.

$L(t)$  : number of MPCs observed for the snapshot at time instant  $t$

The channel parameters to be estimated are  $\boldsymbol{\theta} = [\alpha_{\ell}(t), \tau_{\ell}(t), L(t)]$  where  $\ell = 1, \dots, L(t)$  and  $t = t_1, \dots, t_M$  with  $M$  being the number of measurement shaphshots.

Since there is no relationship between the shaphshots, we can modify the parameter estimation problem to a single snapshot at a time

$$\boldsymbol{\theta} = [\alpha_{\ell}, \tau_{\ell}, L] \quad (3)$$

The equation for received signal (single shaphshot) can be written as

$$y(k) = h(k) \cdot x(k) = \sum_{\ell=1}^L \alpha_{\ell} \cdot x(k) \cdot e^{-j2\pi(k-1)\tau_{\ell}/K} + n(k) \quad (4)$$

$y(k)$  : Received signal for the  $k$ th subcarrier

$x(k)$  : Transmitted signal (calibration signal) for the  $k$ th subcarrier

$n(k)$  : AWGN

Rewriting above equation in vector form by stacking all subcarrier terms into single vector

$$\mathbf{y} = \sum_{\ell=1}^L \alpha_{\ell} \mathbf{x}(\tau_{\ell}) + \mathbf{n} \quad (5)$$

We were able to convert Equation 14 into an optimization problem described by Equation 17 (Appendix A). Unfortunately, above Equation 5 will result in an intractable <sup>1</sup> likelihood function if we attempt the similar optimization steps (steps described between Equation 16 and 17 in Appendix A). We need to employ iterative methods to separate out the multipaths which is discussed in next section.

Estimation problem :

$$\min_{\Theta} \|\mathbf{y} - \sum_{\ell=1}^L \alpha_{\ell} \mathbf{x}(\tau_{\ell})\|^2 \quad (6)$$

Comparison to Equation 17,  $\mathbf{A}_m \rightarrow \mathbf{y}$ ,  $\alpha_l \rightarrow \mathbf{M}$ ,  $\mathbf{A} \rightarrow \mathbf{x}$

---

<sup>1</sup>Intractable means unsolvable and this is due to aggregation of multipath components in a single received sample

## 7.2 SAGE implementation

Below are the steps while implementing the SAGE algorithm. For more mathematical details, please refer to Appendix G and H

$$\mathbf{x}(\tau_i) = [x(1) \cdot e^{-j2\pi(0)\tau_i/K} \quad x(2) \cdot e^{-j2\pi(1)\tau_i/K} \quad \dots \quad x(K) \cdot e^{-j2\pi(K-1)\tau_i/K}]'$$

1. **Initialize all complex amplitudes to zero.**  $\alpha_l = 0$

Reason [Step 1 in SAGE algorithm] : We need to initialize the variables to some value before applying iterative optimization. For the very first optimization process, we utilize the fact that the LOS component will be the strongest and the optimization will be able to narrow down on the right time delay and complex amplitude.

2. **Estimate the i-th multipath component, considering the other multipath as known parameters.** Hence  $\mathbf{y}_i$  becomes

$$\mathbf{y}_i = \mathbf{y} - \sum_{l \neq i} \alpha_l \mathbf{x}(\tau_l) \quad (7)$$

Reason [ Step 1 in SAGE algorithm ] : We have chosen the i-th multipath parameters  $\tau_i, \alpha_i$  to optimize on. The rest of parameters as in equation above is used to subtract out other MPCs from total signal.

Reason [ Step 2 + 3 in SAGE algorithm ] : Remember that SAGE algorithm requires “admissible hidden-data space”. By subtraction all the other multipath components, we focus only just 1 path. This leaves us with only two parameters  $\tau_i, \alpha_i$ . We perform E-step using last computed values (or initialized values for first MPC first iteration).

3. **Apply maximization to estimate the parameter**

$$\hat{\tau}_i = \max_{\tau_i} \|\mathbf{x}(\tau)^H \mathbf{y}_i\| \quad (8)$$

Reason [Step 4 in SAGE algorithm] : We apply M-step. **how to explain the derivation of this beyond intuition? Can I say that the autocorrelation property of Zadoff Chu sequences ensure maximum value for right time delay**

Note that SAGE algorithm helps determine a local minima. Since we need to determine a global minimum, we perform numerical search. We attempt all integer values and utilize the optimization function to narrow to few decimal places in accuracy. The numerical search will target the strongest signal.

#### 4. Determine remaining parameter

$$\hat{\alpha}_i = \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i / \|\mathbf{x}(\hat{\tau}_i)\|^2 \quad (9)$$

Reason : With  $\mathbf{y}_i$ , we only have two parameters to estimate  $(\tau_i, \alpha_i)$ . Since we have already performed optimization on one of the parameters  $\tau_i$ , the remaining one  $\alpha_i$  can be determined in closed form using above equation.

Proof:

$$\begin{aligned} \mathbf{y}_i &= \mathbf{y} - \sum_{l \neq i} \alpha_l \mathbf{x}(\tau_l) \approx \hat{\alpha}_i \mathbf{x}(\hat{\tau}_i) \\ \Rightarrow \mathbf{x}(\hat{\tau}_i)^H \hat{\alpha}_i \mathbf{x}(\hat{\tau}_i) &= \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i \quad \text{premultiplying both sides by } \mathbf{x}(\hat{\tau}_i)^H \\ \Rightarrow \hat{\alpha}_i \mathbf{x}(\hat{\tau}_i)^H \mathbf{x}(\hat{\tau}_i) &= \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i \quad \hat{\alpha}_i \text{ is a scalar hence re-ordering terms} \\ \Rightarrow \hat{\alpha}_i \|\mathbf{x}(\hat{\tau}_i)\|^2 &= \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i \\ \Rightarrow \hat{\alpha}_i &= \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i / \|\mathbf{x}(\hat{\tau}_i)\|^2 \end{aligned}$$

#### 5. Repeat step 2 for all multipath

Reason : For every iteration, we choose a different index set (all parameters pertaining to one multipath component), estimate the parameters, subtract out the path and continue to next path. In this fashion, we remove the strongest path (LoS) in the first iteration followed by next strongest path, and so forth. As and when we remove the stronger paths, we can focus on the next weaker paths.

#### 6. Repeat step 3 a few times until you deem it accurate enough

Reason : There is a possibility of residual energy even if we attempted to remove a significant portion of energy from each multipath. More iterations helps reduce this residual energy

### 7.3 Matlab Simulation Results

Reference Code : C4\_Sage\_Implementation.m

The successive peak removal as part of SAGE algorithm can be clearly seen in Figure 17, 18, 19, 20

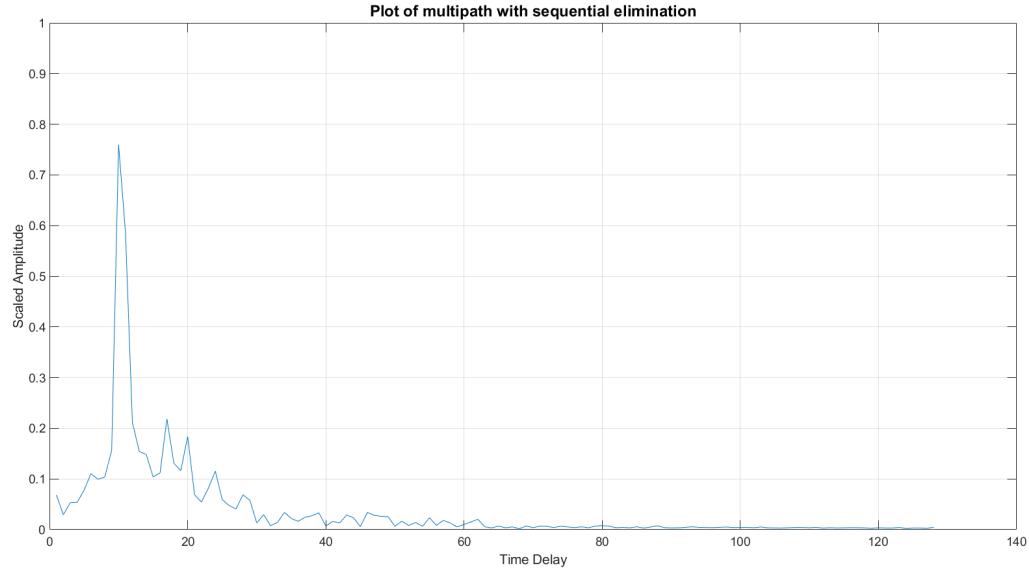


Figure 17: Original Impulse response output

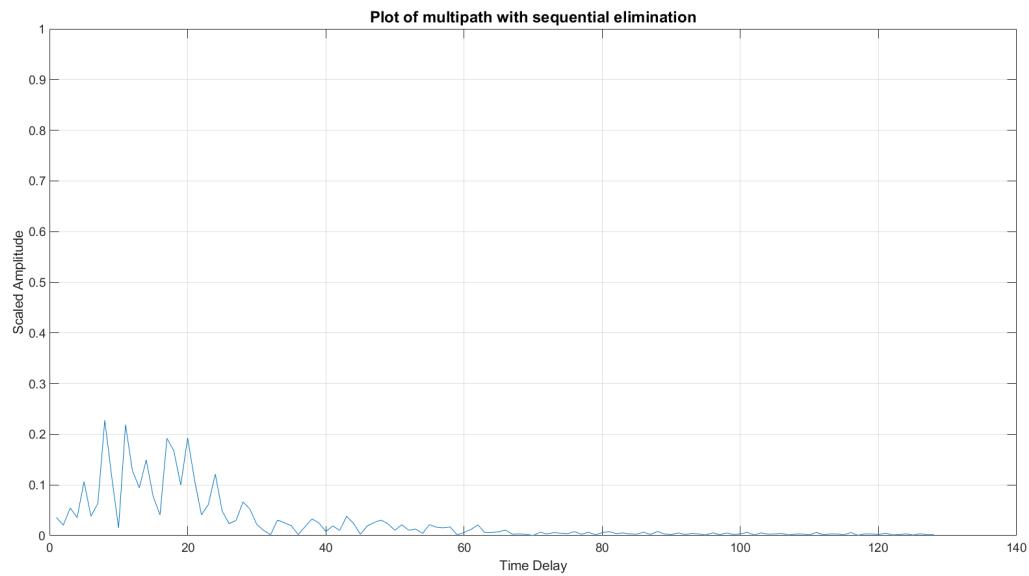


Figure 18: After removing 1st peak or LoS

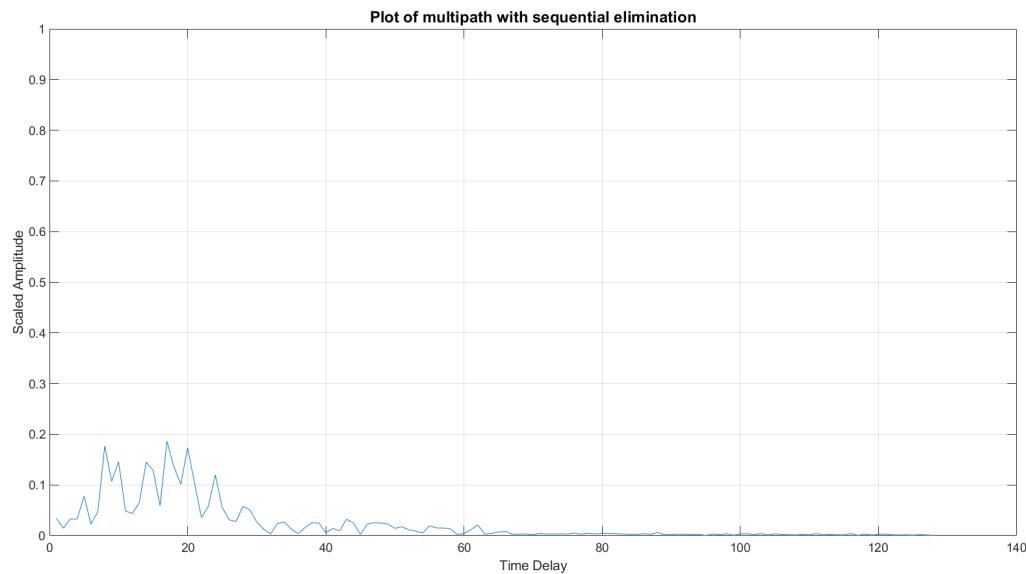


Figure 19: After removing 1st and 2nd peak

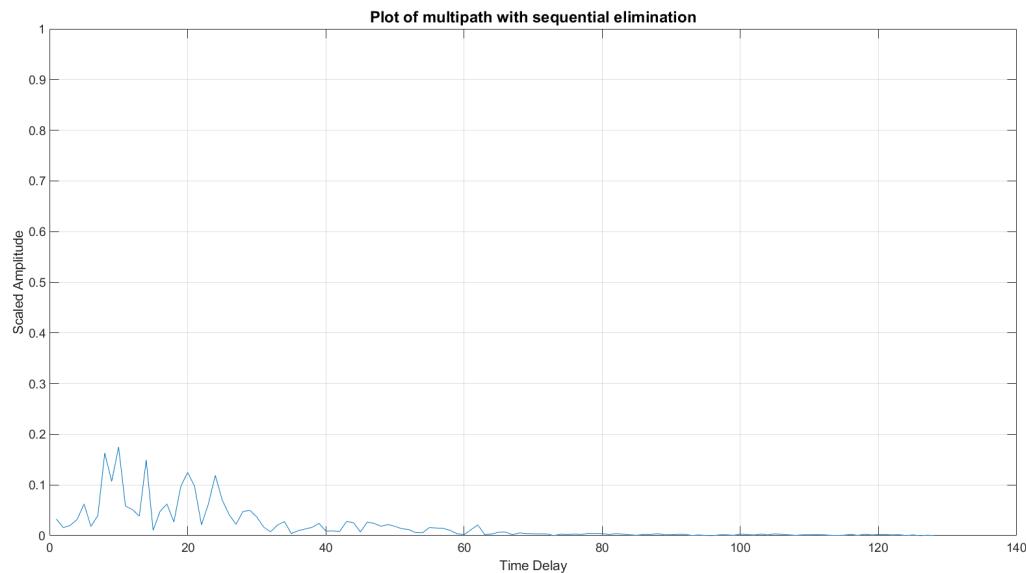


Figure 20: After removing top three peaks

After sufficient iterations, we have sufficient confidence in LoS signal estimation, which can then be used later for calibration.

The retrieved LoS Components from SAGE is shown in Figure 21. Each point corresponds to a phase value of LoS signal, collected for different positions of the Tx Antenna.

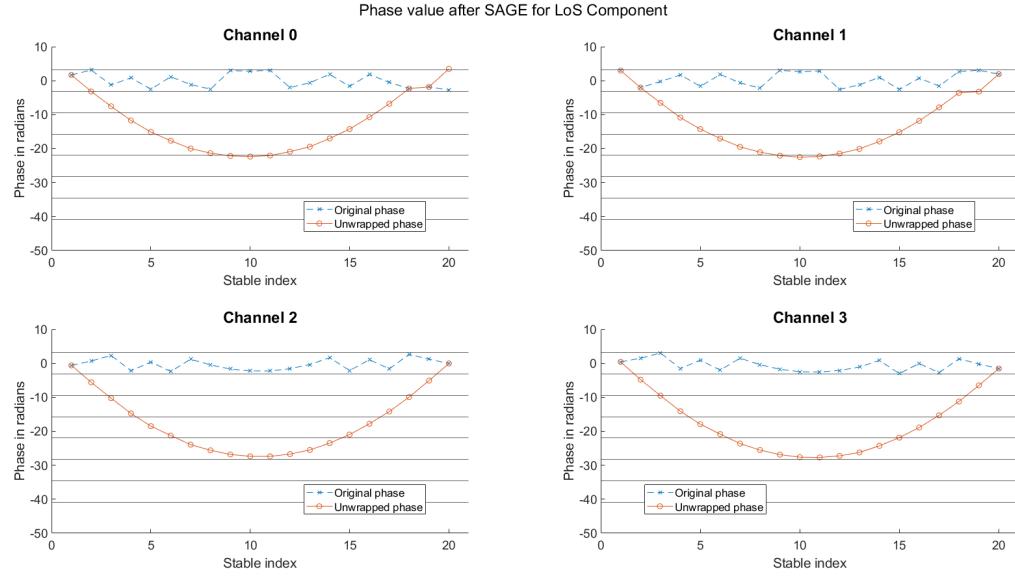


Figure 21: LoS Componenets after SAGE

The phase unwrapping is opposite to what was done earlier, is there sign problem somewhere in the computation

## 7.4 Verifying Measurement Data Reliability

Unwrapping phase is very good way to measure reliability of the measurement data from Table 1. We should expect the start and end phase to align (or almost align) with unwrapping phase. Below Figure 22 is an example of a reliable measurement case. Channel0, the very last phase value seems to have a problem for both measurement data and SAGE output, which I suspect to be issue with phase shift greater than  $2\pi$ . This problem was discussed in more detail as part of [Room for Improvement 1](#) in section 2.

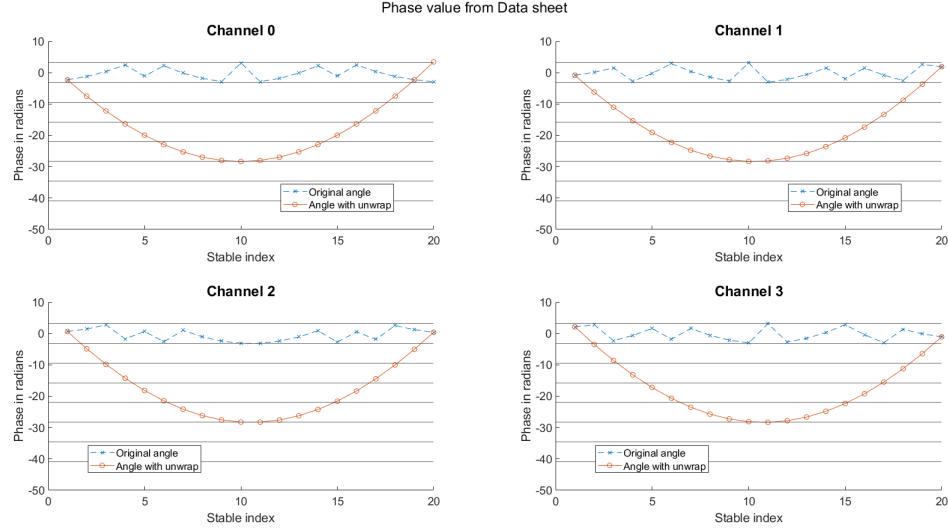


Figure 22: Phase from measurement data

To provide perspectives, the case where the provided distance between the antenna patches was incorrect (7cm instead of 2.6cm) is shown in Figure 23. You can clearly see the unwrapping phase problem, and is a very good error spot check method.

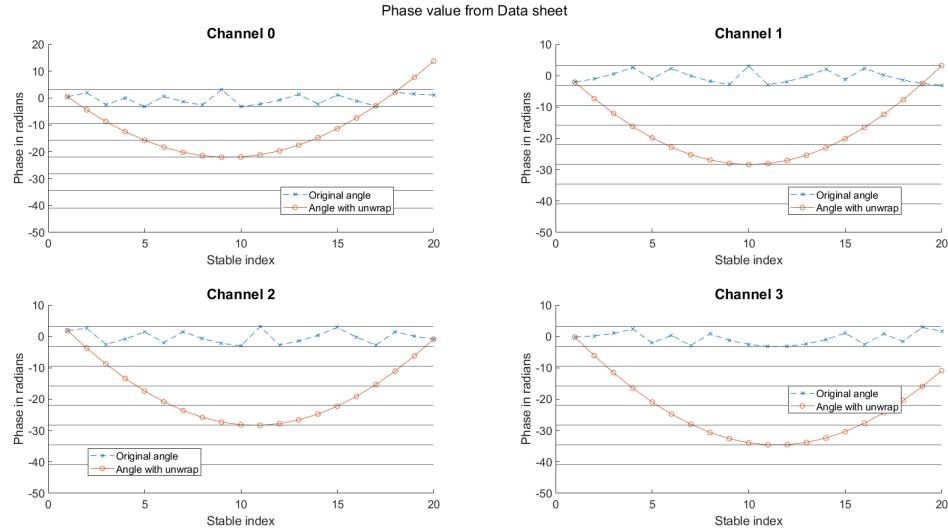


Figure 23: Phase from measurement data with measurement inaccuracy

## 8 Calibration

Reference Code : C5\_FinalCalibration.m

Given that we performed SAGE to extract LoS component and we have reliable measurement data, we can now turn our attention to calibration process. We utilize the approach provided in [2].

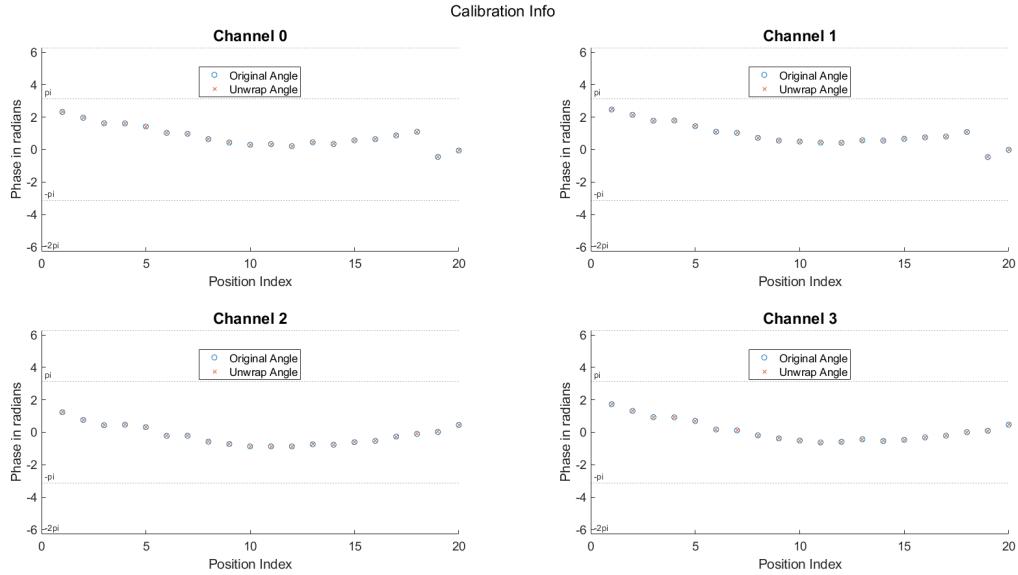


Figure 24: Final Calibration Results

A near straight line is best case for calibration results, but it appears there is a slight non-linear phase behavior as the Tx moves towards the mid and then away from the mid. We need a way to verify the accuracy of the results against the ground truth, which is covered in the next section.

## 9 MUSIC Algorithm for verifying calibration data reliability

Reference Code : C6\_MusicImplementation.m

MUSIC (MULTiple SIgnal Classification) is an algorithm useful in radio direction finding [3]. We use the *musicdoa* function available in Matlab to determine the Direction of Arrival (DoA) and compare the results against the ground truth.

The output from SAGE algorithm contains phase contributions due to propagation as well as the Tx/Rx components. Taking the calibration results from section 8 and subtracting this out from SAGE results from section 7.3, we get the phase contribution from channel. We can then feed this information into MUSIC algorithm to verify DoA estimation accuracy.

The algorithm expects a semi-positive definite matrix to perform decomposition and operates on the eigen vectors. This matrix also called a Covariance Matrix is the product of the steering vector with its hermitian. A steering vector is just the phase value of the received LoS signal at each of the patch antenna written in vector form. We have 4 patch antennas, hence vector of size 4.

We deal with complex phase vectors giving us unsymmetric complex covariance matrix, and directly using the matrix as input to the music function results in failure for some position indexes. We perform few operations before feeding the covariance matrix to the music function. To get better understanding of what we need to do, lets look at the eigen value and eigen vector for one sample of the Covariance Matrix.

Eigen Value Matrix ( $4 \times 4$ )

$$\begin{bmatrix} -6.66786213940145e-16 & 0 & 0 & 0 \\ 0 & -3.87977807963586e-16 & 0 & 0 \\ 0 & 0 & 5.55632997410900e-17 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

Eigen Vector Matrix ( $4 \times 4$ )

$$\begin{bmatrix} +0.153528580240315 - 0.458507933915201i & +0.151031093319364 + 0.137858318176127i & \dots \\ +0.467280405884219 + 0.205066339202291i & +0.181743253205344 - 0.244153006866384i & \dots \\ +0.217914399258811 - 0.464971134494054i & -0.161224357677090 + 0.623675472096618i & \dots \\ +0.492048195883331 + 0.0000000000000000i & +0.6712518701089 + 0.0000000000000000i & \dots \\ \dots & +0.509921879945141 + 0.462994404825617i & -0.109702063159603 + 0.487817032644953i \\ \dots & +0.442354921515545 - 0.448639767927753i & -0.492045560963922 - 0.088832234778255i \\ \dots & +0.105299536258310 - 0.245484782305216i & +0.052411702845505 - 0.497245425725400i \\ \dots & -0.239394170393599 + 0.0000000000000000i & +0.5000000000000000 + 0.0000000000000000i \end{bmatrix}$$

Looking at Eigen Vector and Eigen Value, there are two properties that we should be concerned with, specifically Very small decimal values and Non-terminating decimal numbers

Because of these properties, there are approximation/rounding errors which makes the matrix **not** semi-positive definite even though they should be. There are few things we can do to get through this problem, described below...

- Perform rounding to 4 decimal places on Eigen Values.
- Perform iterative reduction in decimal places for Eigen Vectors until music function works. Start with considering all 15 places, and keep reducing to 14, 13, ... until music function usage doesn't complain about matrix.

Note : In the custom function *F.RecResolReductForMusic* we cap at minimum of 5 decimal places, to avoid high rounding errors and to utilize alternative approximation approaches.

After performing the necessary value adjustments, we get good DoA Estimates as shown in Figure 25

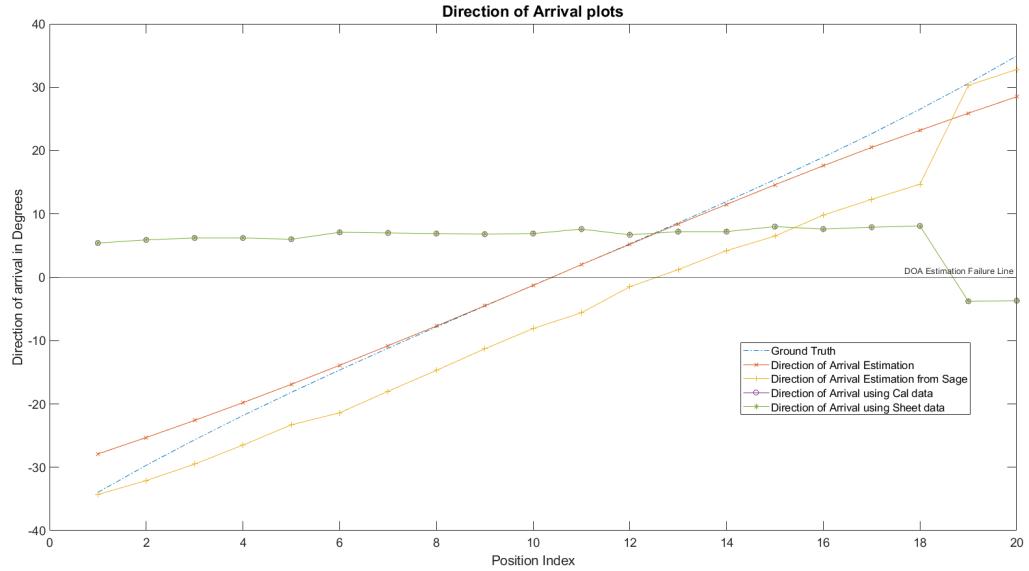


Figure 25: DoA Plots for various signals

We see good results near the mid positions, but it deviates towards the ends. Note sure how to explain the anomaly at extreme points

## 10 Overall Summary

To wrap our heads around the entire list of steps described earlier, I will summarize the various steps, the files used and conclusion

Step1 : Layout Requirement of Tx and Rx antennas (section 2 )

Purpose : Determine the Tx/Rx layout to ensure we get good LoS signal and low multipaths

Source Code : Most of the effort is manual calculations with one useful code ArrangementCalculator.m

Step2 : Data Collection ( section 3 )

Purpose : Collect both physical measurements as well as OTA and B2B data required for calibration.  
Measurement details are saved in DATA SHEET.

Source Code : **NEED GIT LOCATION FOR FW/SW TO BE LOADED ON USRP AND SOURCE CODE TO COLLECT SAMPLES FROM USRP**

Step3 : Frequency Error Detection and Correction ( section 4 )

Purpose : Remove frequency errors prior to calibration process.

Source Code : C1\_FrequencyErrorAndCorrection.m

Step4 : Stable stationary point selection for Channel Estimation ( section : 5 )

Purpose : Select stable OTA samples to avoid phase jitters due to physical antenna movements

Source Code : C2\_KNN\_Clustering.m

Step5 : Unwrapping error removal using outlier detection ( section : 6 )

Purpose : Catch errors while performing phase unwrapping

Source Code : C3\_OutlierDetectorAndPhaseCorrector.m

Step6 : LoS Separation ( section : 7 )

Purpose : We need to use LoS signal phase with measurement data to perform calibration.

Source Code : C4\_Sage\_Implementation.m

Step7 : Calibration ( section : 8 )

Purpose : Finally, we perform calibration to determine the phase contribution from the hardware

Source Code : C5\_FinalCalibration.m

Step8 : MUSIC Algorithm for verifying calibration data reliability ( section : 9 )

Purpose : We need to verify the calibration results accuracy, for which MUSIC algorithm is very useful.

Source Code : C6\_MusicImplementation.m

As shown in section 8 and 9, we were able to perform calibration reliably using wide band signal which is very useful to perform accurate channel estimations.

Last but not the least, we should be aware of the improvement areas in case we intend to perform the listed processes again on a different setup or for a different receiver.

- Ensure that you have a middle position point at dead center (in relation to patch antenna center) with equal position points on both sides.
- Select number of points on both sides of mid such that the number is mathematically friendly.
- Ensure the separation between the points doesn't fall below  $0.2\pi$  difference in propagation path, and doesn't exceed  $1.8\pi$  difference in propagation path.

For setup used above (values picked from ArrangementCalculator.m), we should set the separation between Tx/Rx to around 1.8m, set 16cm as separation between sliding positions and 64cm as maximum length (which corresponds to 4 positions) on either sides of mid. This ensures we have total number of positions as 9 which is mathematically friendly (also 16/64 are mathematically friendly).

## A Summary of Calibration algorithm and simplification

The sensor array calibration paper[4] covers in detail calibration algorithm, starting with the most generic problem statement of using  $n$  sources and  $m$  sensors. This section will identify the assumptions/approximations made by the paper followed by simplifying the problem statement to suit our needs and the relevant equations.

In order to obtain the phase difference w.r.t reference sensor, we use the below equation

$$a(\psi_i, \theta_j) = e^{j\omega_o \tau_i(\theta_j)} \quad (10)$$

$\psi_i$  : sensor coordinate position such as value of x, y and z-axis in 3D space

$\theta_i$  : direction of arrival such as value of azimuth and elevation angle in 3D space

$\tau_i$  : time delay which is a function of  $\theta_i$

Solving the system of equations in 3D space would be quite complex and require lots of accurate/high precision sensor data. Instead we can opt to simplify the equation by assuming the source and sensors in a single plane and far field approximation.

**Assumption 1 :** Sensor and source in a single plane

Reason : 3D space can be reduced to 2D space

**Assumption 2 :** Wavefront follows far field approximation

Reason : Easier to compute the relative distance from source to different sensors

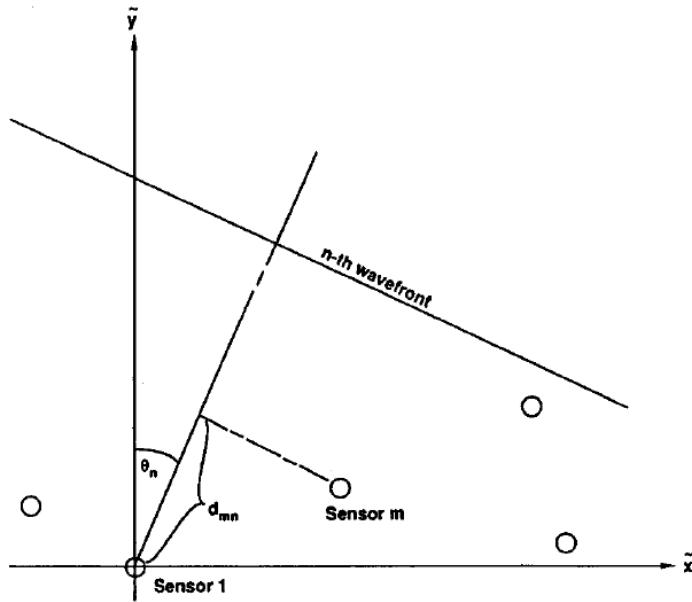


Figure 26: Problem geometry [5]

Assumption 1 and 2 simplifies the calculations for relative time delays of received signal across sensors. The Figure 26 will help understand how the time delays are derived. Assuming Sensor 1 as reference, the delay of the signal from source  $n$  on the sensor  $m$  can be expressed by below equation

$$\tau_m(\theta_n) = -d_{mn}/c = -\frac{1}{c}(x_m \sin \theta_n + y_m \cos \theta_n) \quad (11)$$

$n$  : Index for source

$m$  : Index for sensor

$\tau_m(\cdot)$  : Time delay of received signal on Sensor  $m$  w.r.t reference sensor (Sensor 1)

$d_{mn}$  : Distance from Sensor  $m$  to reference sensor in the direction of source  $n$

$(x_m, y_m)$  : Coordinates of sensor  $m$  also represented as  $\psi_m$

$\theta_n$  : Direction of Arrival (DOA) of source  $n$  relative to y-axis

$c$  : Propagation velocity or the speed of light

Rewriting sensor coordination positions as an array

$$\boldsymbol{\psi} = [\psi_1, \psi_2, \dots, \psi_m]^T = [(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)]^T$$

Rewriting in vector form to represent phase vector,

$$\mathbf{a}(\boldsymbol{\psi}, \theta_j) = [e^{j\omega_o \tau_1(\theta_j)}, e^{j\omega_o \tau_2(\theta_j)}, \dots, e^{j\omega_o \tau_m(\theta_j)}] \quad (12)$$

Note, we have only represented the phase contributions based on physical location of sources. We still need to factor in phase and amplitude contributions by each sensor that may or may not be related to other sensors. We will introduce a matrix to describe exactly these characteristics.

$\mathbf{M} \in \mathbb{C}^{m \times m}$  : Calibration matrix comprising of mutual coupling coefficients between sensor elements and the gain and phase terms of each sensor (or channel)

For a single signal arriving from  $\theta$ , the complex envelope of the received signal vector,

$$\mathbf{r}(t) = \mathbf{M} \cdot \mathbf{a}(\boldsymbol{\psi}, \theta_j) \cdot s(t) + \mathbf{e}(t) \quad (13)$$

While performing experiments, what is known to us is the received signal  $\mathbf{r}(t)$  and the complex envelop of the source signal  $s(t)$ . What we need to determine is calibration matrix  $\mathbf{M}$  and phase vector  $\mathbf{a}(\boldsymbol{\psi}, \theta_j)$  or to be more specific, we need to determine  $\boldsymbol{\psi}$ , but  $\theta$  is known. We will come back to how  $\theta$  can be calculated in our case. The zero mean additive Gaussian noise vector  $\mathbf{e}(t)$  is not known and cannot be measured.

Performing summation on Equation 13 for N samples over time, we get

$$\begin{aligned} \sum_{t=1}^N \mathbf{r}(t) &= \sum_{t=1}^N \left( \mathbf{M} \cdot \mathbf{a}(\boldsymbol{\psi}, \theta_j) \cdot s(t) + \mathbf{e}(t) \right) \\ &= \mathbf{M} \cdot \mathbf{a}(\boldsymbol{\psi}, \theta_j) \cdot \sum_{t=1}^N s(t) + \sum_{t=1}^N \mathbf{e}(t) \\ &= \mathbf{M} \cdot \mathbf{a}(\boldsymbol{\psi}, \theta_j) \cdot \alpha N + \sum_{t=1}^N \mathbf{e}(t) \end{aligned}$$

$$\frac{1}{\alpha N} \sum_{t=1}^N \mathbf{r}(t) = \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j) + \frac{1}{\alpha N} \sum_{t=1}^N \mathbf{e}(t) \quad (14)$$

$$\mathbf{a}_m(\theta_j) = \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j) + \mathbf{n}_j \quad (15)$$

$\mathbf{a}_m(\theta_j)$  : Steering vector for a calibration source at location  $\theta_j$

$\mathbf{n}_j$  : zero-mean Gaussian distribution but with different covariance  $\zeta^2 \mathbb{I}$  as  $\mathbf{e}(t)$  where  $\mathbb{I}$  is identity matrix of size  $n \times n$

Equation 15 is a linear problem, but because  $\mathbf{n}_j$  is not a measurable quantity, we have to rely on stochastic approaches such as utilizing probability density function. We can easily formulate the optimization equation (because noise is zero-mean Gaussian) but we need to ensure that signal from different sources do not overlap when received by a sensor. We can achieve this by collecting the received signal at every sensor after transmitting from all  $n$  sources but in temporally disjoint fashion. In other words, transmit only one source at a time, and measure the signal at the sensor, before transmitting from the next source.

$$\max_{\mathbf{M}, \psi} p(\mathbf{a}_m(\theta_1), \mathbf{a}_m(\theta_2), \dots, \mathbf{a}_m(\theta_n) | \mathbf{M}, \psi, \zeta^2) = \max_{\mathbf{M}, \psi} \frac{1}{(\pi \zeta^2)^{nm}} e^{-\frac{1}{\zeta^2} \sum_{j=1}^n \|\mathbf{a}_m(\theta_j) - \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j)\|^2} \quad (16)$$

We can simplify the above equation by taking log but also convert a maximizing problem to a minimizing problem by taking the negative log. The constant terms can be ignored as they dont contribute to optimization.

$$\min_{\mathbf{M}, \psi} \left( -\log(p(\cdot)) \right) = \min_{\mathbf{M}, \psi} \left( - \left( \log \left( \frac{1}{(\pi \zeta^2)^{nm}} \right) - \frac{1}{\zeta^2} \sum_{j=1}^n \|\mathbf{a}_m(\theta_j) - \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j)\|^2 \right) \right)$$

getting rid of constant additive terms

$$= \min_{\mathbf{M}, \psi} \left( \frac{1}{\zeta^2} \sum_{j=1}^n \|\mathbf{a}_m(\theta_j) - \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j)\|^2 \right)$$

getting rid of constant multiplicative terms

$$= \min_{\mathbf{M}, \psi} \left( \sum_{j=1}^n \|\mathbf{a}_m(\theta_j) - \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j)\|^2 \right)$$

expanding summation term and swapping order within modulus square

$$= \min_{\mathbf{M}, \psi} \|\mathbf{M} \cdot \mathbf{a}(\psi, \theta_1) - \mathbf{a}_m(\theta_1)\|^2 + \dots + \min_{\mathbf{M}, \psi} \|\mathbf{M} \cdot \mathbf{a}(\psi, \theta_n) - \mathbf{a}_m(\theta_n)\|^2$$

converting additive vector terms into matrix multiplication + Frobenius Norm

$$= \min_{\mathbf{M}, \psi} \|\mathbf{M} \mathbf{A}(\psi) - \mathbf{A}_m\|_{\textcolor{red}{F}}^2$$

$$\mathbf{A}_m = [\mathbf{a}_m(\theta_1), \mathbf{a}_m(\theta_2), \dots, \mathbf{a}_m(\theta_n)]$$

$$\mathbf{A}(\psi) = [\mathbf{a}(\psi, \theta_1), \mathbf{a}(\psi, \theta_2), \dots, \mathbf{a}(\psi, \theta_n)]$$

Optimization problem (changed order of terms, outcome doesnt change) :

$$\min_{\mathbf{M}, \psi} \|\mathbf{A}_m - \mathbf{M} \mathbf{A}(\psi)\|_{\textcolor{red}{F}}^2 \quad (17)$$

We cannot solve the optimization problem unless the number of measurements is equal to or exceeds the unknowns, or in other words, the number of equations must be equal to or exceed the number of unknowns

**M** : This is a  $m \times m$  matrix implying  $m \times m = m^2$  terms and hence  $m^2$  unknowns. But we also have to keep in mind this is complex matrix, resulting in two unknowns (real + imaginary) for every element in the matrix. This gives us a total of  $2m^2$  (real) unknowns.

**$\psi$**  : The coordinates of the  $m$  sensors are not known, and hence  $2m$  (real) unknowns.

**$\mathbf{A}_m$**  : This is known and will give us  $m \times n$  independent measurements, but if we factor in both real and complex, we get  $2mn$  (real) known measurements.

To ensure a possible solution, we can represent the condition as an inequality expression

$$2mn \geq 2m^2 + 2m \quad (18)$$

and also  **$\mathbf{A}_m$**  needs to be full rank!

**Assumption 3** : No mutual coupling between sensor elements

Benefit : Reduces the number of unknowns for system of equations

The sensors or in our case the USRP receivers are assumed to have good shielding and hence negligible coupling. This will convert **M** into diagonal matrix instead of full matrix thereby reducing the number of (real) unknowns from  $2m^2$  to  $2m$ . Rewriting the inequality expression18 as

$$2mn \geq 2m + 2m \quad (19)$$

**Assumption 4** : Sensor antennas are in single array

Reason : Reduces the number of unknowns for system of equations

Note this is different from phase array antennas where all the antennas feed to a single receiver; here we have antenna patches on a single board and each patch is connected to a different receiver. This simplifies the problem statement by reducing the number of unknowns from 2 coordinate values to single distance between antenna patches reducing the  **$\psi$**  unknown contribution from  $2m$  to  $m$ . Rewriting the inequality expression19 as

$$2mn \geq m + m \Rightarrow n \geq 1 \quad (20)$$

This gives us the ability to perform calibration using just 1 source! But do notice that we had implicitly assumed that the direction of arrival  $\theta$  is known, and hence must be clearly listed as an assumption.

**Assumption 5** : Direction of Arrival is known

Reason : Reduces the number of unknowns for system of equations

The derivations so far has assumed a single signal from source to sensor, but unfortunately we cannot assume a pure LoS signal from source to the sensor under most environment conditions, and especially not true in indoor environment that has high likelihood of reflective surfaces for carrier frequency we are operating at.

**Assumption 6** : Only LoS signal from source to sensor.

Reason : Derivation described earlier doesn't consider multipath case

Like any wireless signal reception, timing synchronization is very critical to know the delay between the transmit and receive signal for our calibration data.

**Assumption 7 :** Timing information can be accurately determine

Reason : Receiving multiple samples for a fix setup requires accurate timing synchronization

## B Far field approximation

Far field sources results in planar wavefronts making it easier to compute the distance from the source in relation to other sensors. The minimum distance between the source and sensors to approximate the wavefront to be planar instead of spherical can be obtained using Rayleigh distance formula,

$$d_R = \frac{2L_a^2}{\lambda} \quad (21)$$

$d_R$  : Separation between the source and sensor

$L_a$  : Largest separation between the sensors

$\lambda$  : wavelength of transmitted signal

## C Fresnel Zone Clearance

Fresnel zones are regions within which if we have a reflecting surface or an obstruction, it will impact the quality of LoS signal. There are two properties that we should be aware of

1. Reflecting surface : If we have a reflecting surface within the first fresnel zone, the multipath reflected off the surface might distractively combine with LoS signal
2. Obstruction near LoS : Obstruction of free path around LoS reduces the strength of the LoS signal (**HOW TO EXPLAIN THIS USING TEXT REFERENCES**) due to Huygens–Fresnel principle of propagating waves.

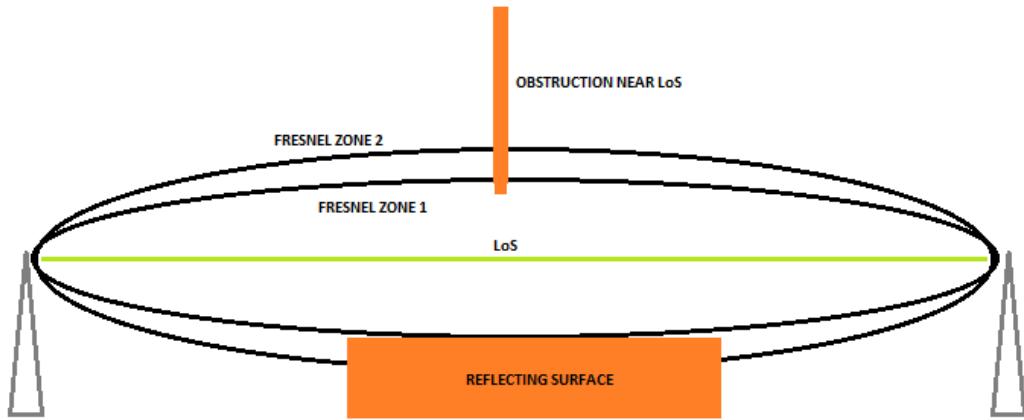


Figure 27: Fresnel Zone

Radius of the first Fresnel Zone is given by

$$F_1 = 8.656 \sqrt{\frac{d_R \times 10^6}{f_c}} \quad (22)$$

$d_R$  : Separation between the source and sensor

$f_c$  : Source signal center frequency

## D Constant amplitude continuous wave calibration signal source

We will use Zadoff Chu sequence to generate the desired source signal that meets the requirement of constant amplitude continuous wave signal [6] and has desirable properties. The sequence needed to generate the calibration signal is described by below equation

$$s_q[n] = e^{-j\pi q \frac{n(n+1)}{\mathcal{N}_{zc}}} \quad (23)$$

$\mathcal{N}_{zc}$  : Length of the sequence (odd number)

$q$  : Root index whose value  $\in (1, 2, \dots, \mathcal{N}_{zc} - 1)$

$n = (0, 1, 2, \dots, \mathcal{N}_{zc})$  : sequence index

The desirable property of zero cyclic autocorrelation makes it very useful in time synchronization, or to time the exact start of the sequence in the received signal.

## E Signalling Mode

As we are working with UWB signal, we will use OFDM scheme to transmit multiple narrow band signals but aggregate them to form a 400Mhz wide band signal. The Figure 28 29 below gives a summary of the transmitter and receiver processes, except the constellation mapping is replaced with zadoff chu sequence at transmitter, with match filter in the receiver. This is because we are not sending data but pure reference signal for proper synchronization and channel estimation. You can also refer to [7], [8] for better understanding.

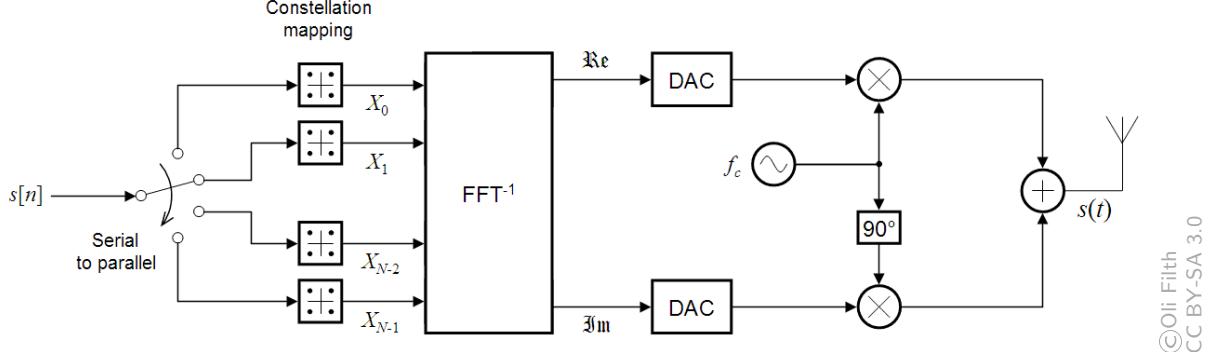


Figure 28: OFDM Transmitter

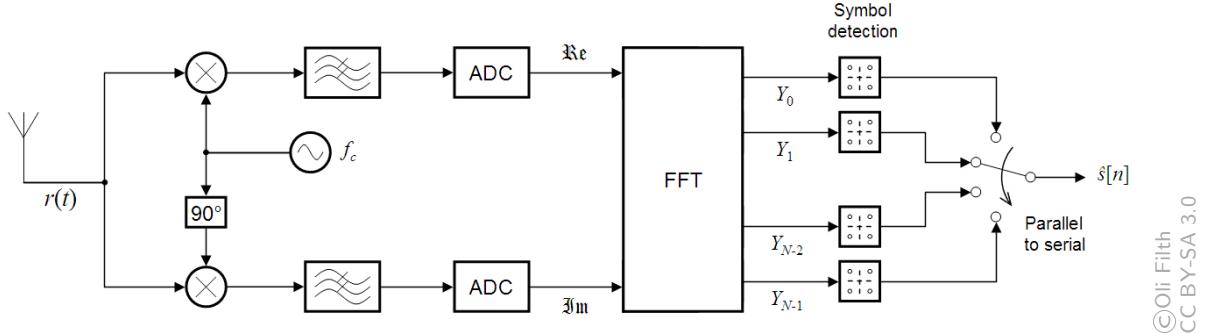


Figure 29: OFDM Receiver

Since we are dealing with multiple subcarriers to implement a UWB signal, we need to determine how many subcarriers should we select. Keep in mind that we need to send a known reference or calibration signal, and since we are going to use Zadoff Chu sequences to generate the desired source signal, we need to follow the requirements listed as part of Equation 23.  $N_{zc}$  needs to be an odd number but an even number of subcarriers is desired to perform FFT optimally. We have 1024 subcarriers with sequence length of 813 (the rest are padded with zeros)

## F KNN Clustering Algorithm for Stable Point Detection

We use a sliding window whose width increases from 1 sec to 13 secs with increment of 2 (selecting only odd secs window size). For each window width we compute the mean and variance of the window samples as we slide from start to the end, moving 1 sec at a time.

If all the windows were considered, it would mean acceptance rate of 100%, but this shouldnt be greater than RatioStableVsMovement. We need a way to penalize/deselect the windows with high jitter. What we have from KNN clustering function is the centroid location, which centroid each data sample belongs to and the sum of points to centroid distances (sumD).

For a given target of sumD (using SumDLimit variable), we can attempt to reject data samples (window sample) whose variance is max, one window at a time and cross check if we go below SumDLimit. The surviving window samples are considered as accepted, but if higher than RatioStableVsMovement percentage, we will need to tighten requirement by lowering down SumDLimit value (we reduce by factor of 10).

Once we are below RatioStableVsMovement, we note down the number of surviving samples. We then repeat this for higher window width ( $1 \rightarrow 3 \rightarrow 5 \rightarrow \dots$ ). While attempting higher window width, if there is any instance where SumDLimit had to be lowered from previous value, we restart from window width 1 and repeat again. This is to ensure equal strictness (SumDLimit value) to all window sizes.

We then note the number of surviving window samples for each window width. The number of surviving window samples will follow a concave shape, meaning there is one window size that will give us max value. We store the window samples and centroid values for this window size, and we pick the window index that is closest to the centroid. This is basically the best stable point!

We repeat the above steps for each channel, take the average of index value across all channels and round it to the nearest integer. This gives us the best stable point across all channels.

NEEDS MORE WORK, TBC

## G Expectation-Maximization Method

The Expectation-Maximization (EM) method[9],[10],[11] is one of the early iterative methods that works by iteratively maximizing the conditional log-likelihood of a single Unobservable Complete <sup>1</sup> data space rather than maximizing the intractable likelihood function for the measured data. EM method in simplest form can be described as iteration between E-step and M-step described below

1. Expectation step (E step):

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}_{\mathbf{X}|\mathbf{Y},\boldsymbol{\theta}}[\log L(\boldsymbol{\theta}; \mathbf{Y}, \mathbf{X})] = \sum_{\mathbf{X}} P(\mathbf{X}|\mathbf{Y}, \boldsymbol{\theta}^{(t)}) \log(P(\mathbf{Y}, \mathbf{X}|\boldsymbol{\theta}))$$

2. Maximization step (M step):

$$\boldsymbol{\theta}^{(t+1)} = \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$$

**Y** : Observed incomplete data (sometimes referred to output)

**X** : Unobserved complete data

**$\theta$**  : Unknown parameters

$t$  : iteration count

The EM method suffers from a drawback of slow convergence. The optimization step above needs to be performed on entire observed data space **Y** and across all parameters  **$\theta$**  simultaneously.

The Space-Alternating Generalized Expectation-Maximization (SAGE) Algorithm [12] [13] solves the drawback via updating the parameters sequentially by alternating between several hidden spaces.

---

<sup>1</sup> “Unobserved Complete” implies (“Unobserved”) not all variables that describes the model are observable (or measurable) and (“Complete”) for those variables that are observable, we have complete information.

## H Space-Alternating Generalized Expectation-Maximization Algorithm

The below steps are directly picked up from [12] with slight change in notations to compare with EM method.

For  $t = (0, 1, \dots)$ , iterate through the below steps...

1. Choose an index set  $\mathbf{S} = \mathbf{S}^t$
2. Choose an admissible hidden-data space  $X^{\mathbf{S}^t}$  for  $\theta_{\mathbf{S}^t}$
3. E-step : Compute  $Q(\theta_{\mathbf{S}^t}; \theta^t)$  using step 4
4. M-step :

$$\theta_{\mathbf{S}^t}^{t+1} = \max_{\theta_{\mathbf{S}^t}} Q(\theta_{\mathbf{S}^t}; \theta^t) \quad (24)$$

$$\theta_{\tilde{\mathbf{S}}^t}^{t+1} = \theta_{\tilde{\mathbf{S}}^t}^t \quad (25)$$

To represent subspace, we need to introduce indexing, or assign one index to each parameter  
 $\Phi(\cdot)$  : Original function to maximize

$\theta = [\theta_1, \theta_2, \dots, \theta_p]'$  : Unknown parameters (same as EM but enumerating for index set)

$\mathbf{S}$  : Index set which is a subset of  $\theta$ , represents selected parameters for optimization

$\tilde{\mathbf{S}}$  : Index set which is a subset of  $\theta$ , represents parameters not selected for optimization (value is fixed)

$\theta = [\theta_{\mathbf{S}}, \theta_{\tilde{\mathbf{S}}}]'$

$\mathbf{X}^{\mathbf{S}}$  : Random vector considering the available parameter  $\theta_{\mathbf{S}}$  **CHECK DEFINITION**

Identifying the admissible hidden-data space is important for SAGE algorithm to work.

Lets consider a probability density function  $f(y, x; \theta)^1$

$$\begin{aligned} f(y, x; \theta) &= f(y|x; \theta)f(x; \theta) && \dots \text{Product rule} \\ &= f(y|x; \theta_{\mathbf{S}}, \theta_{\tilde{\mathbf{S}}})f(x; \theta) && \dots \text{Separating sets} \\ &= f(y|x; \theta_{\tilde{\mathbf{S}}})f(x; \theta) && \dots \text{True only if } ^2 f(y|x; \theta_{\tilde{\mathbf{S}}}) \perp\!\!\!\perp \theta_{\mathbf{S}} \end{aligned}$$

Definition for “Admissible hidden-data space” : A random vector  $\mathbf{X}^{\mathbf{S}}$  with probability density function  $f(x, \theta)$  is an admissible hidden-data space with respect to  $\theta_{\mathbf{S}}$  for  $f(y; \theta)$  if the joint density of  $\mathbf{X}^{\mathbf{S}}$  and  $\mathbf{Y}$  satisfies

$$f(y, x; \theta) = f(y|x; \theta_{\tilde{\mathbf{S}}})f(x; \theta) \quad (26)$$

In other words,  $\mathbf{X}^{\mathbf{S}}$  must be a complete-data space for  $\theta_{\mathbf{S}}$  given that  $\theta_{\tilde{\mathbf{S}}}$  is known.

---

<sup>1</sup>Semicolon Notation :  $f(\mathbf{a}; \mathbf{b})$  : A function  $f$  that depends on variables  $\mathbf{a}$  and parameters  $\mathbf{b}$ . Parameter is similar to variable but stays fixed when we use the function.

<sup>2</sup> $\perp\!\!\!\perp$  : Independent of

## References

- [1] NI, “Ettus usrp x410 specifications.” <https://www.ni.com/docs/en-US/bundle/ettus-usrp-x410-specs/page/specs.html>, 2022. Accessed: 2022-12-15.
- [2] J. Pierre and M. Kaveh, “Experimental performance of calibration and direction-finding algorithms,” in [Proceedings] ICASSP 91: 1991 International Conference on Acoustics, Speech, and Signal Processing, pp. 1365–1368 vol.2, 1991.
- [3] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [4] B. C. Ng and C. M. S. See, “Sensor-array calibration using a maximum-likelihood approach,” *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 6, pp. 827–835, 1996.
- [5] B. Friedlander and A. Weiss, “Direction finding in the presence of mutual coupling,” *IEEE Transactions on Antennas and Propagation*, vol. 39, no. 3, pp. 273–284, 1991.
- [6] J. G. Andrews, “A primer on zadoff chu sequences,” Nov. 2022.
- [7] *MIMO-OFDM Wireless Communications with MATLAB®*. John Wiley & Sons, Ltd, 2010.
- [8] *Wireless OFDM Systems*. Springer New York, NY, 2002.
- [9] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.
- [10] Wikipedia contributors, “Expectation–maximization algorithm — Wikipedia, the free encyclopedia,” 2022. [Online; accessed 29-December-2022].
- [11] M. Bonakdarpour, “Introduction to em: Gaussian mixture models.” [https://github.com/stephens999/fiveMinuteStats/blob/master/docs/intro\\_to\\_em.html](https://github.com/stephens999/fiveMinuteStats/blob/master/docs/intro_to_em.html), 2019.
- [12] J. Fessler and A. Hero, “Space-alternating generalized expectation-maximization algorithm,” *IEEE Transactions on Signal Processing*, vol. 42, no. 10, pp. 2664–2677, 1994.
- [13] K. Hausmair, K. Witrisal, P. Meissner, C. Steiner, and G. Kail, “SAGE algorithm for UWB channel parameter estimation,” tech. rep., EURO-COST 2100, Athens, Greece, Feb. 2010.