

Wide Band Receiver Calibration using USRP X410

Advanced Course in Electrical and Information Technology - EITN35

Nikhil Challa
Email : ni7666ch-s@student.lu.se
Personal Number : 19870209-5855

Supervisor : Anders J Johansson
Mentor : Michiel Sandra
Examiner : Liang Liu

June 8th 2023

Contents

1	Introduction	3
2	Methodology	5
2.1	Introduction to calibration processes	5
2.2	Signal Model	5
2.3	SAGE implementation	7
2.4	Summarized sequence of steps to complete the calibration process	8
3	Processes	9
3.1	Setup Requirement of Tx and Rx antennas	9
3.2	Data Collection	11
3.2.1	Tx-Rx Antenna setup	11
3.2.2	Cabling details	13
3.2.3	USRP Configuration	14
3.2.4	Table of positional data	15
3.3	Frequency Error Detection and Correction	17
3.4	Stable stationary point selection for Channel Estimation	19
3.5	Unwrapping error removal using outlier detection	21
3.6	LoS Separation	23
3.7	Verifying B2B Measurement Data Reliability	24
3.8	Calibration	25
3.9	MUSIC Algorithm for verifying calibration data reliability	26
4	Conclusion	28
4.1	Results	28
4.2	Room for Improvements	29
4.3	Recommended Setup Requirement for better calibration data estimates	30
A	Summary of Calibration algorithm and simplification	31
B	Far field approximation	36
C	Fresnel Zone Clearance	36
D	Constant amplitude continuous wave calibration signal source	37
E	Signalling Mode	38
F	KNN Clustering Algorithm for Stationary Point Detection	39
G	Expectation-Maximization Method	40
H	Space-Alternating Generalized Expectation-Maximization Algorithm	41
I	Processes summary and source code details	42

1 Introduction

While describing the channel in mathematical form, we utilize the below expression (Example 1 Tx and 4 Rx) represented in the frequency domain

$$Y(f) = H(f) \cdot X(f) \quad (1)$$

$Y(f)$: Received Signal Matrix of size (4×1)

$H(f)$: Frequency response of the channel or Channel Matrix of size (4×1)

$X(f)$: Transmitted Signal Matrix of size (1×1)

We can utilize $H(f)$ for a variety of purposes which can be broadly classified into three groupings, but there may be more applications beyond those listed below

- Direction of Arrival (DoA) estimation for a variety of applications such as radar, radio astronomy, remote sensing, and many more.
- Matrix decomposition of channel matrix to perform Maximum Radio Combining (MRC), Water pouring, or other methods to improve channel capacity
- Reusing Precoding Matrix Information (PMI) for different User Equipments (UEs) in modern cellular systems.

But in reality, the received signal also contains contributions from the hardware, and can be written as

$$Y(f) = \alpha \cdot Rx(f) \cdot H(f) \cdot Tx(f) \cdot X(f) = G(f) \cdot X(f) \quad (2)$$

$Tx(f)$: Transmitter hardware contributions

$Rx(f)$: Receiver hardware contributions

α : Phase of incoming signal the PLL locks to

We can club all the new terms into “hardware response” of the system. The α doesn’t change until the receiver system reboot, but $Tx(f)$ and $Rx(f)$ are expected to remain the same for a given under normal operating conditions. For the applications listed earlier, we will need to separate out the hardware response of the system to extract the channel response. Using well established calibration methods further discussed in the next section, we can estimate the hardware contributions, and thereby estimate $H(f)$.

Thus the objective of this document is to identify methods for calibration; perform calibration for multiple receiver system using an open source hardware (E.g. USRP X410) and verify the estimation accuracy. By calibration we mean, estimating the phase changes to reference signal due RF signal conditioning units such as low-noise amplifiers, filters and other components within the hardware at transmitter and receiver for received signal. Note that strictly speaking, amplitude changes should also be determined, but is out of scope of this document.

We will be utilizing the below components to attempt the calibration process.

- One Ettus USRP X410 unit ([1])
- One omnidirectional antenna (Transmitter Antenna)
- Patch antenna with atleast 4 patches (Receiver Antenna)
- RF cables to connect the USRP to the antennas
- 20dB attenuators (for Back-to-Back measurements)
- CPU for data sample collection
- Ethernet cable to connect USRP to CPU

Since we are restricted to using patch antennas that are not omnidirectional, we will focus only on phase estimation and not amplitude estimation for the hardware response. The reason for selecting a patch antenna as a receiver is to perform DoA estimation.

The following terms will be used interchangably :

source : transmitter

sensor : receiver

Tx : Transmitter

Rx : Receiver

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

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

2 Methodology

2.1 Introduction to calibration processes

The Appendix A gradually simplifies a more complex problem statement, the most complex being a random arrangement of sources and sensors to a more simplified problem statement, arriving at the conclusion that only one measurement is necessary to perform calibration for a single pair of Tx-Rx.

Starting with Equation 2 and rewriting the full channel response $G(f)$ in terms of OTA channel response and hardware response. We can also describe $G(f)$ in terms of available terms namely $X(f)$ and $Y(f)$.

$$G(f) = H(f) \cdot (\alpha Rx(f) \cdot Tx(f)) = \frac{Y(f)}{X(f)} \quad (3)$$

$G(f)$ can be determined using the reference signal $X(f)$ and the received signal $Y(f)$. Through some means, if we can accurately estimate $H(f)$, it will be possible to estimate the hardware response. One trick is to arrange the receiver and transmitter in such a way that we can perform near accurate physical measurements. Using simple geometry, we determine the LoS path distance and calculate phase shift of the transmitted signal. This arrangement is described in more detail in section 3.1.

Using the recommended arrangement, we can collect both OTA and B2B data for the calibration process along with the physical measurements, described in more detail in section 3.2.

After determining the layout and collecting both the OTA and B2B signal, it is crucial to check for frequency errors due to the receiver. This frequency error results in a linearly increasing or decreasing phase error, which is covered in more detail in section 3.3. It is important to rectify the frequency error, before we move to the next stage in the calibration process.

Wirelessly transmitted signals consist of multipath components, and hence the separation of the LoS component from the other multipath is necessary. To achieve this, we adopt a well-researched SAGE algorithm, summarized in Appendix H. We first describe the “Signal Model” followed by steps for the SAGE algorithm in the subsequent subsections.

2.2 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

$$h(k, t) = \frac{y(k; t)}{x(k)} \quad (4)$$

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 Equation 5

$$h(\tau; t) = \sum_{\ell=1}^{L(t)} \alpha_\ell(t) \delta(\tau - \tau_\ell(t)) \quad (5)$$

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

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

$\delta(\tau - \tau_\ell(t))$: time varying delay of the ℓ 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 snapshots.

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

$$\boldsymbol{\theta} = [\alpha_\ell, \tau_\ell, L] \quad (6)$$

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

$$y(k) = h(k) \cdot x(k) = \sum_{\ell=1}^L \alpha_\ell \cdot x(k) \cdot e^{-j2\pi(k-1)\tau_\ell/K} + n(k) \quad (7)$$

$y(k)$: 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

$$\mathbf{y} = \sum_{\ell=1}^L \alpha_\ell \mathbf{x}(\tau_\ell) + \mathbf{n} \quad (8)$$

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

Estimation problem :

$$\min_{\Theta} \|\mathbf{y} - \sum_{\ell=1}^L \alpha_\ell \mathbf{x}(\tau_\ell)\|^2 \quad (9)$$

(Comparing to Equation 20, $\mathbf{A}_m \rightarrow \mathbf{y}$, $\alpha_l \rightarrow \mathbf{M}$, $\mathbf{A} \rightarrow \mathbf{x}$)

Unfortunately, solving for Equation 9 will result in an intractable ¹ likelihood function if we attempt the similar optimization steps (steps described between Equation 19 and 20 in Appendix A). We need to employ iterative methods to separate out the multipath which is where the SAGE algorithm will come to our rescue.

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

2.3 SAGE implementation

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

$$\mathbf{y}_i = \mathbf{y} - \sum_{l \neq i} \alpha_l \mathbf{x}(\tau_l) \quad (10)$$

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

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

3. Apply maximization to estimate the parameter

$$\hat{\tau}_i = \max_{\tau_i} \|\mathbf{x}(\tau)^H \mathbf{y}_i\| \quad (11)$$

Reason [Step 4 in SAGE algorithm] : We apply M-step. The SAGE algorithm helps determine a local minima. Since we need to determine a global minimum, we perform a numerical search and target the strongest signal. We attempt all integer values and utilize the optimization function to narrow to a few decimal places in accuracy.

4. Determine remaining parameter

$$\hat{\alpha}_i = \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i / \|\mathbf{x}(\hat{\tau}_i)\|^2 \quad (12)$$

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

Proof:

$$\mathbf{y}_i = \mathbf{y} - \sum_{l \neq i} \alpha_l \mathbf{x}(\tau_l) \approx \hat{\alpha}_i \mathbf{x}(\hat{\tau}_i)$$

$$\Rightarrow \mathbf{x}(\hat{\tau}_i)^H \hat{\alpha}_i \mathbf{x}(\hat{\tau}_i) = \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i \quad \text{premultiplying both sides by } \mathbf{x}(\hat{\tau}_i)^H$$

$$\Rightarrow \hat{\alpha}_i \mathbf{x}(\hat{\tau}_i)^H \mathbf{x}(\hat{\tau}_i) = \mathbf{x}(\hat{\tau}_i)^H \mathbf{y}_i \quad \hat{\alpha}_i \text{ is a scalar hence re-ordering terms}$$

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

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

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 the next path. In this fashion, we remove the strongest path (LoS) in the first iteration followed by the next strongest path, and so forth. As and when we remove the stronger paths, we can focus on subsequent 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 help reduce this residual energy

The results of the SAGE algorithm are covered in more detail in section 3.6

Once we have the LoS signal information, we can then perform calibration to retrieve the hardware contribution to the phase rotation. This is covered in more detail in section 3.8

Finally, with estimating the hardware contributions we complete the calibration process. To ensure the accuracy of our estimation, we utilize the MUSIC algorithm, which is covered in more detail in section 3.9.

There are additional intermediate steps, not explicitly covered here but will be cover in section 3

2.4 Summarized sequence of steps to complete the calibration process

Below is a very brief list of important steps to complete the calibration process, explained in more detail in the next section.

- Planning of Measurements
- Perform OTA Channel Measurements and B2B Measurements
- Perform frequency error estimation and frequency correction
- Determine stationary points in the OTA data after phase unwrapping
- Fix errors in unwrapping using outlier detection
- Perform SAGE algorithm to extract LoS component (time delay and complex signal)
- Verify B2B measurement data reliability
- Perform calibration using the LoS signal phase (determine phase contributions from the hardware)
- Perform correctness of the calibration data using DoA estimation

3 Processes

3.1 Setup Requirement of Tx and Rx antennas

Reference Code : ArrangementCalculator.m

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 adjacent patch antennas RxS : 2.6 cm
- Number of Rx Patch Antennas $Nrx = 4$

We also need to measure the Tx antenna base-plate dimensions, detailed in the next subsection. We now need to determine the desired height placement of the antenna, Tx-Rx separation, and other such requirements

We consider the requirements listed in Appendix A, that will be factored in for layout requirements. The Appendix contains details to help understand the reason for these requirements

- Requirement 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 a single Rx) gives us the possibility of using the MUSIC algorithm to verify estimated calibration data accuracy, covered in section 3.9.
- Requirement 1 requires us to ensure the source and sensor are 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.
- Requirement 2 requires us to ensure the separation between the source and sensor is 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 0.23 meters.

Largest Separation between Rx antennas $L_a = RxS * (Nrx - 1) = 0.026 * 3 = 0.078$

$$\text{Rayleigh distance } d_R = \frac{2L_a^2}{\lambda} = \frac{2(0.078)^2}{0.052} \approx 0.23 \text{ meters}$$

- Requirement 3 requires no-mutual coupling between the receivers. The USRP X410 unit is assumed to have decent shielding (negligent coupling) between the different receivers.
- Requirement 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 3.6



(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 the bench separation, and the LoS height is greater than the bench height. This ensures the requirements for both the Rayleigh distance ($> 0.23\text{m}$) and the first Fresnel Zero radius ($> 0.15\text{m}$) are met.

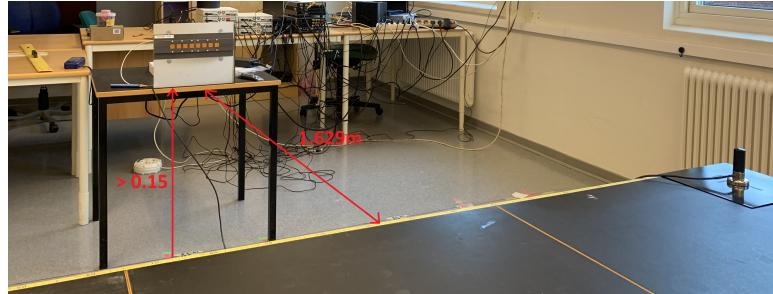


Figure 2: Measurements to check layout requirements

3.2 Data Collection

3.2.1 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 the Tx and Rx antenna won't give us confidence in the accuracy of calibration results. 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 3. Ensure that the center of the row of benches is in the same position as the center of the patch antenna; to obtain equal measurements on either side. Note down the measurement from one end to the center of the bench row setup as shown in Figure 3 (106cm)

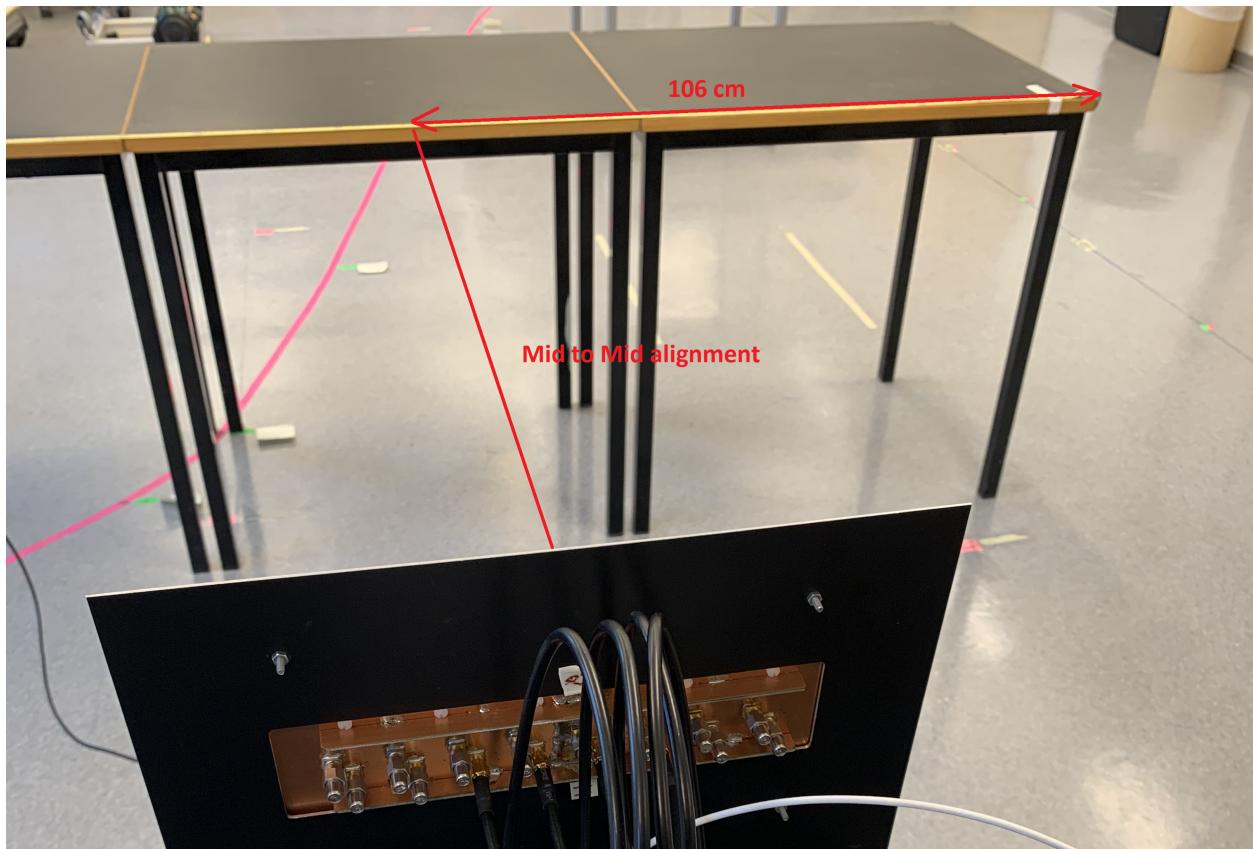


Figure 3: Mid point measurement

Once you have the bench row arrangement completed, we measure the Tx antenna bottom plate's physical dimensions. The measurement value is shown in Figure 4. We need these measurements as the propagation path distance is from Tx to Rx, not from bench to bench.

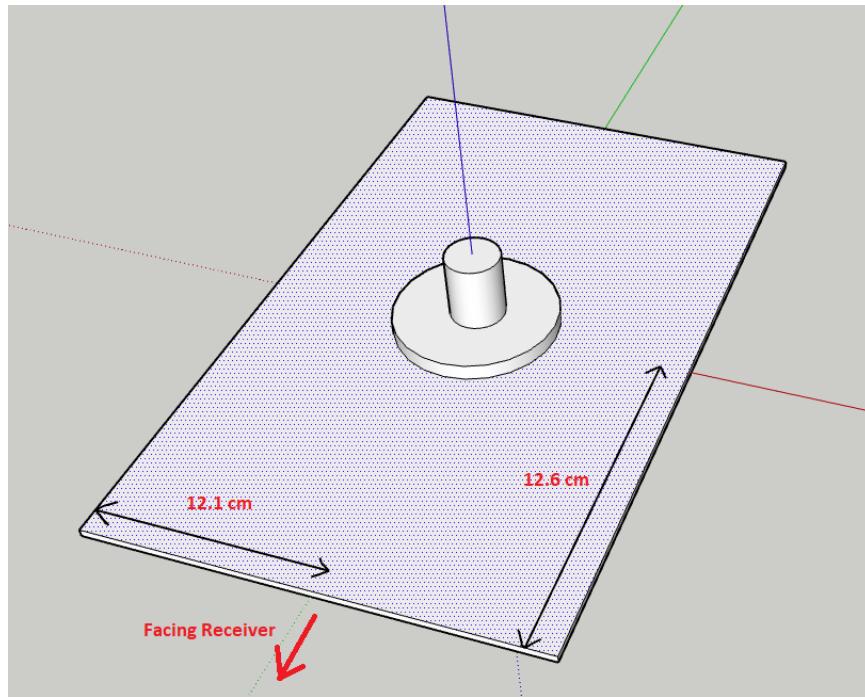


Figure 4: Tx antenna on metal plate

We will utilize the sliding approach to collect multiple data points, as shown in Figure 5. For this, we mark from left to right, a 10cm interval and attempt to collect a set of measurements for every 10 cm shift in the Tx antenna. The more measurements, the more data points we have, and the more is our confidence in the calibration data.

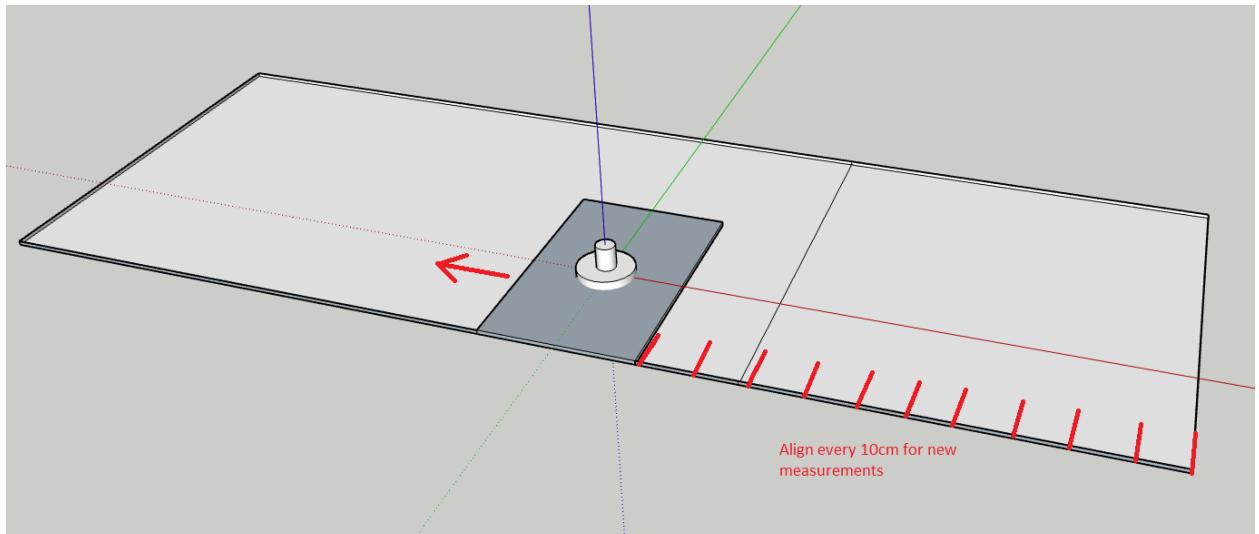


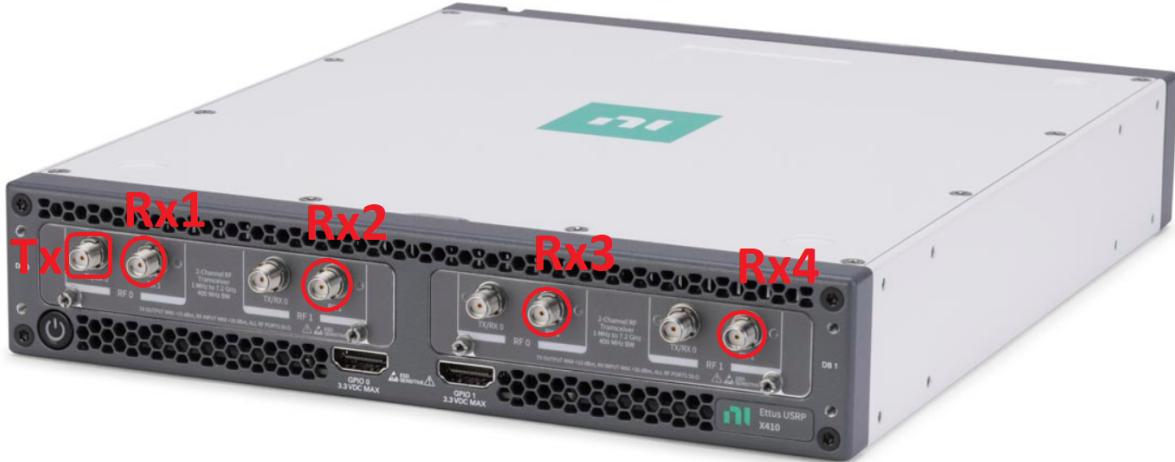
Figure 5: Sliding Tx Antenna

3.2.2 Cabling details

Now we turn our attention to RF cable connections between USRP and Tx-Rx antennas. Refer to Figure 6 for the details. The Tx port will be connected directly to the omnidirectional antenna.



(a) Selecting patch antenna



(b) USRP Connections

Figure 6: Patch antenna to USRP (Port numbers marked in red)

3.2.3 USRP Configuration

Center frequency : 5.725 Ghz

Waveform Type : OFDM (Appendix E)

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

Subcarrier spacing : 488Khz

Number of subcarriers : 813

Sampling rate : 500 Msps

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 7 as an example of Tx to Rx3 case. We need to collect the same for Tx to Rx0, Tx to Rx1, and Tx to Rx2.

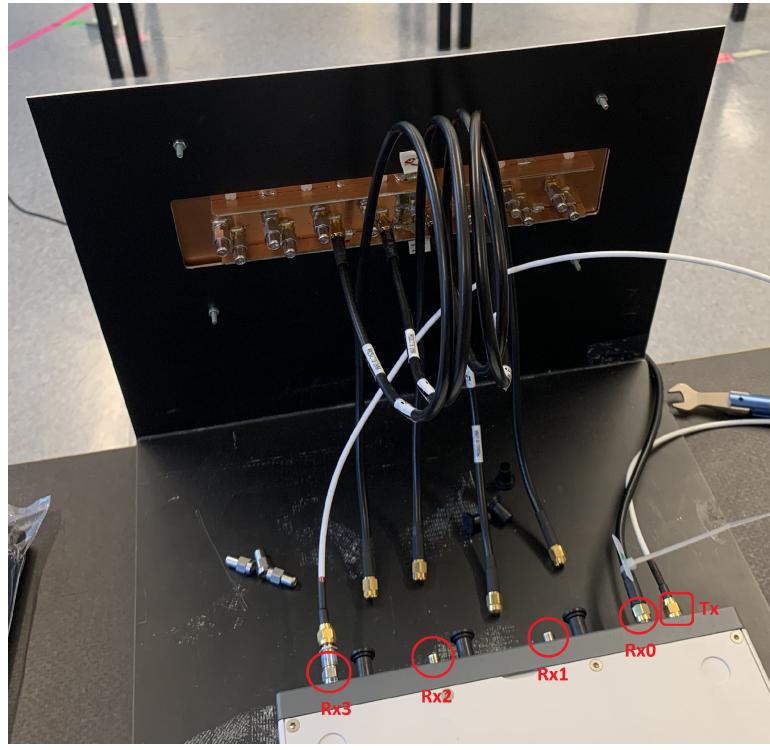


Figure 7: B2B Data collection (Tx to Rx3 connection)

3.2.4 Table of positional data

Using the sliding approach (Figure 5), 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 calculation results are rounded to 4 decimal places
- The Excel sheet in Git has more columns, only the first 11 columns are posted above due to size restrictions
- The 35th column as configurable variables, do not modify any other contents (after column 11) as they auto-populate based on configurable variables.

3.3 Frequency Error Detection and Correction

Reference Code : C1_FrequencyErrorAndCorrection.m

It is always good to check if we have frequency errors prior to the calibration process. Frequency error can be easily identified by checking the phase offset from the first sample. At each sampling instance, the phase should be the same if the frequency error is zero. We utilize B2B data for this verification, as there is no OTA-related degradation. If we observe a linear increment or decrement in phase difference, then the contribution to this offset in phase is the frequency error. The phase offset for each channel using B2B Data is shown in Figure 8

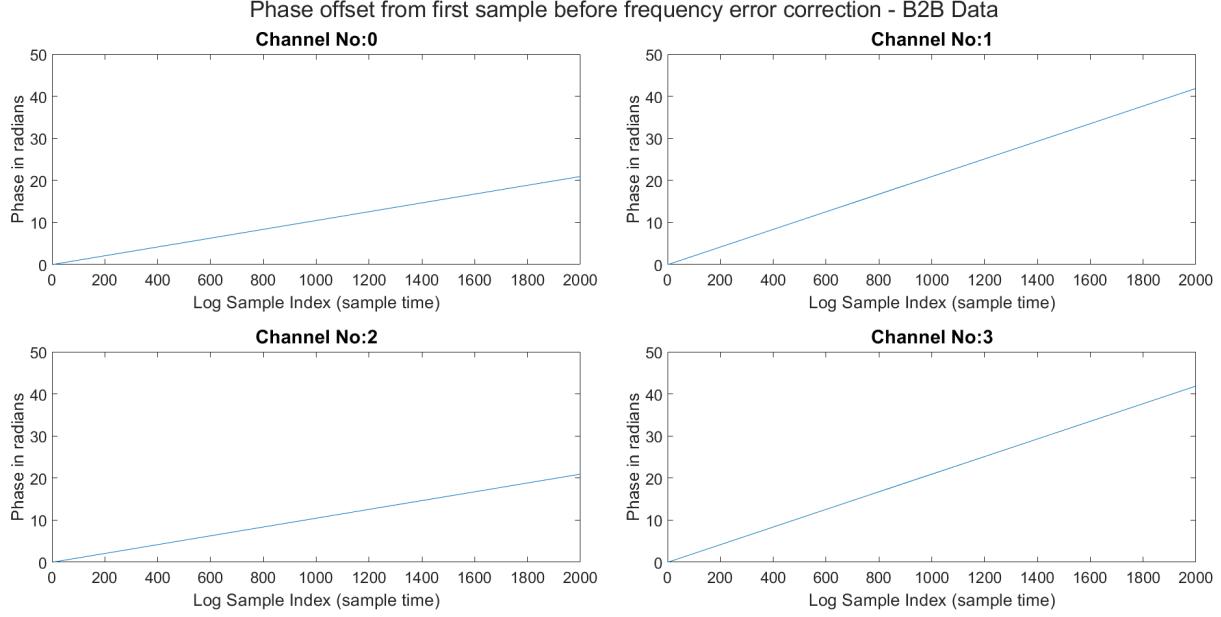


Figure 8: Phase error before frequency error removal

Notice that phase is usually measured within the range of $[-\pi \ \pi]$ or $[0 \ 2\pi]$. We used the concept of phase unwrapping to see a clear increment and plot beyond the bounds if the frequency error is high. Phase unwrapping makes it easier to calculate a single slope value for each Channel. The more the number of data samples, the more accurate the slope estimate via 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 the frequency error is shown in Figure 9

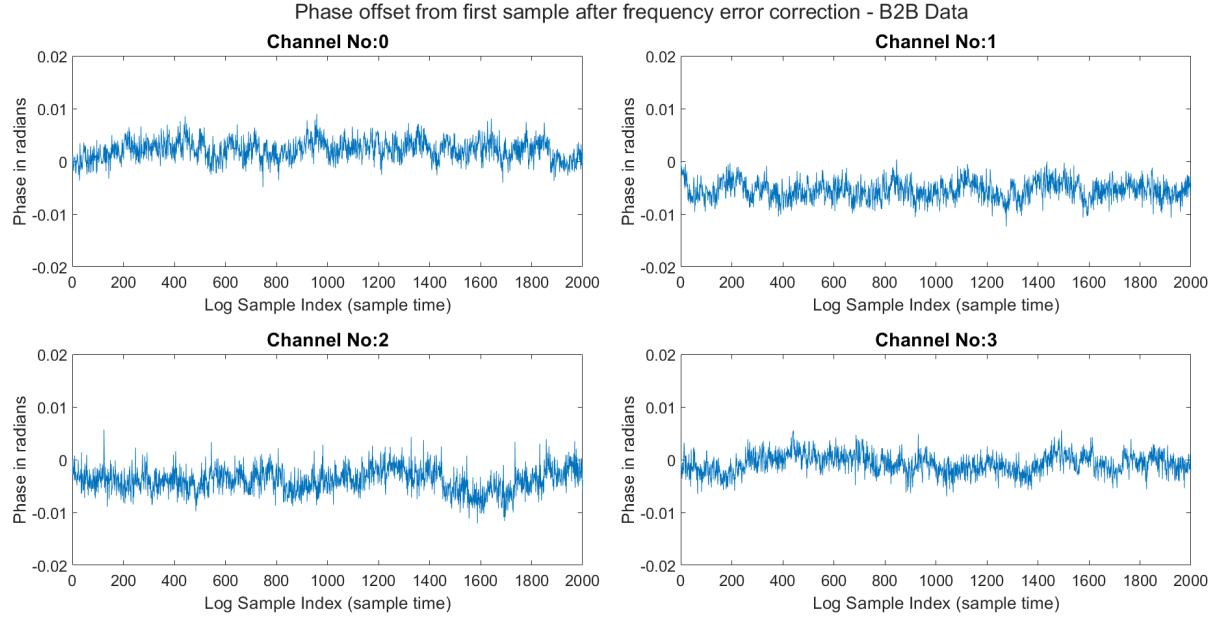


Figure 9: Phase error after frequency error removal

For the provided data, the frequency error is approximately [0.3334, 0.6667, 0.3333, 0.6667] corresponding to channel 0,1,2,3 respectively.

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

3.4 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 variations either due to physical obstruction, or actual antenna movement. It takes a few secs to stabilize after completing the antenna placement hence selecting the stationary points for calibration at every step is necessary. Figure 10 gives a good view of phase jitters.

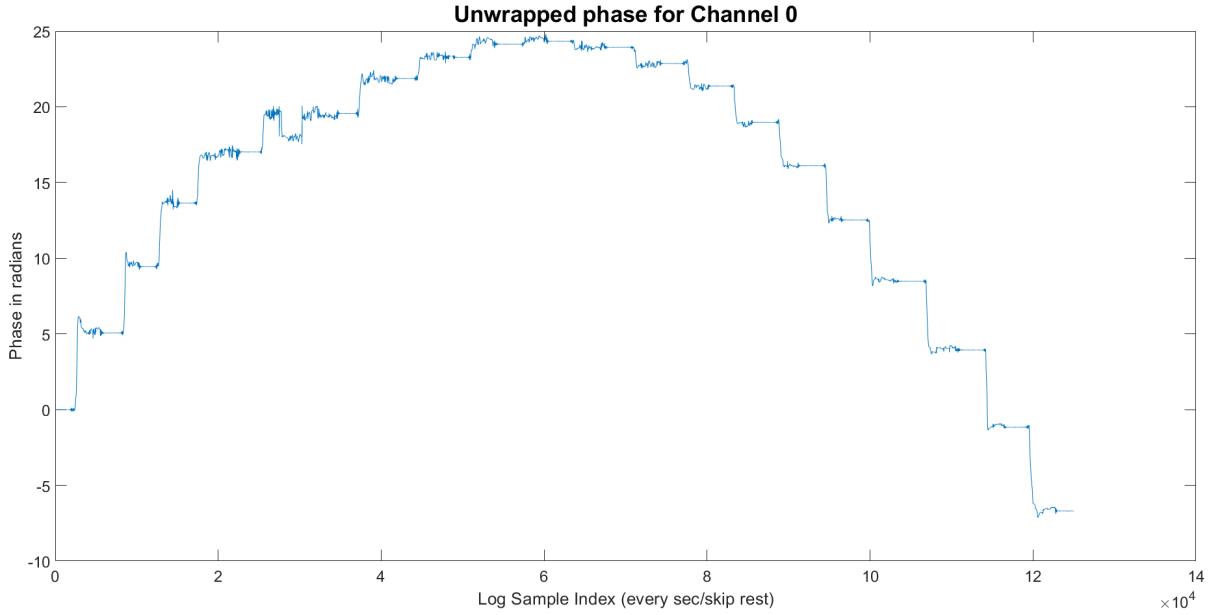


Figure 10: Sample of Phase Jitter

The stationary 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 stationary points. This corresponds to the 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 the Algorithm :

- Provide two values, number of stationary points (eg 20) and percentage of time in stationary state (eg 0.5). The “Acceptance Rate Limit” should be less than percentage of time in stationary state.
- Introduce the concept of a window, which is 1 or more seconds of consecutive OFDM samples. Window width is in multiples of 1 sec. Each second has 200 sample points in our dataset.
- Start with a window width, utilize sliding window (minimum slide duration = 1sec) to pick a different sets 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 the number of centroids equal to the number of stationary points. (This needs to be modified for symmetric positioning)

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

The locations of the stable positions picked up by the above algorithm can be seen in Figure 11

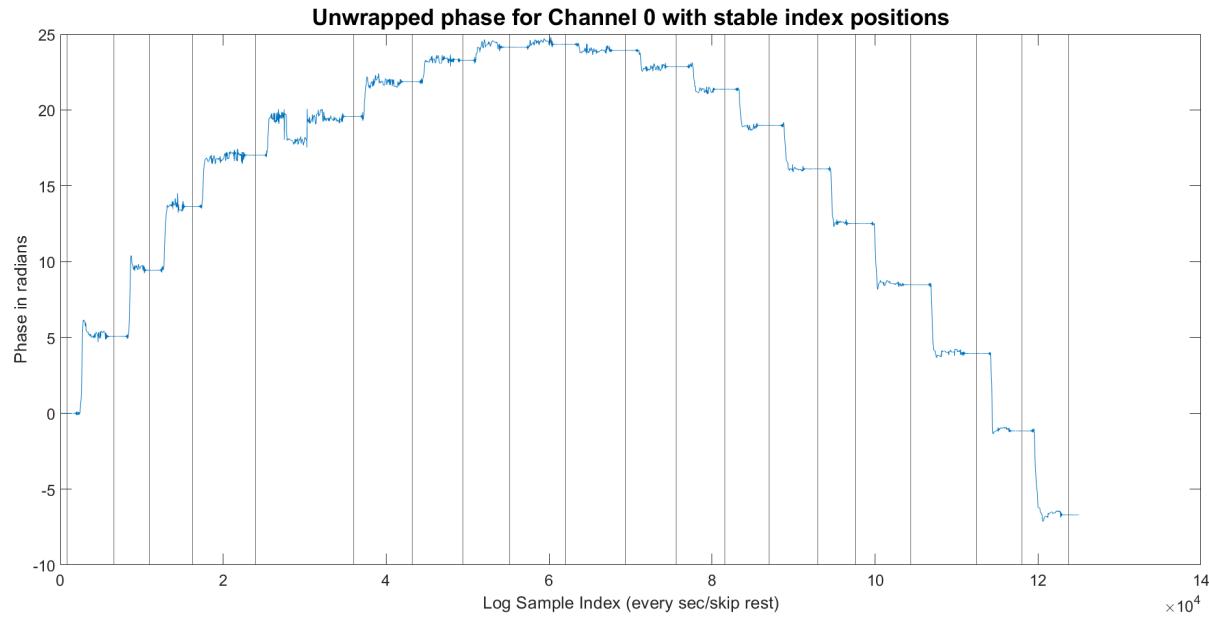


Figure 11: Vertical plot of Stable Positions

3.5 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 the error removal process. Figure 12 shows a good example of an unwrapping error in Channel 3. The blue symbols indicates the wrapped phase and the orange symbols indicate the unwrapped phase. Carefully observing the overlapping instance to non-overlapping instance of the red and blue symbols near Log Sample Index 3.028×10^4 , you will see a jump of more than π 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 a clue on what the unwrapped phase should be and detect outliers.

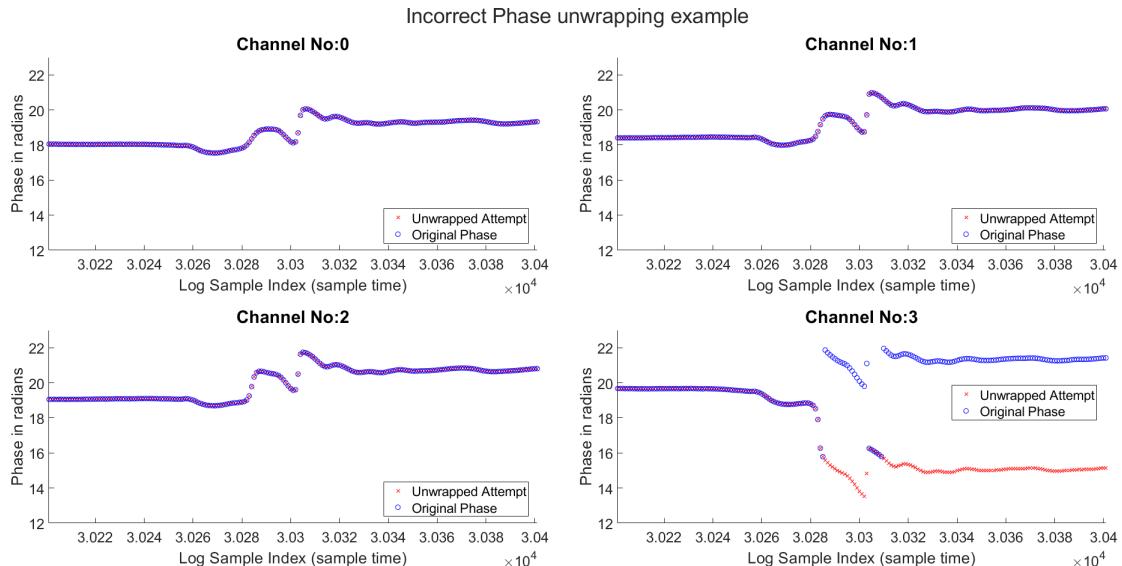


Figure 12: Unwrapping Error Example (Channel3)

Figure 13 shows (within the green box) the error with phase unwrapping, and using information from other channels, we successfully correct the phase unwrapping error.

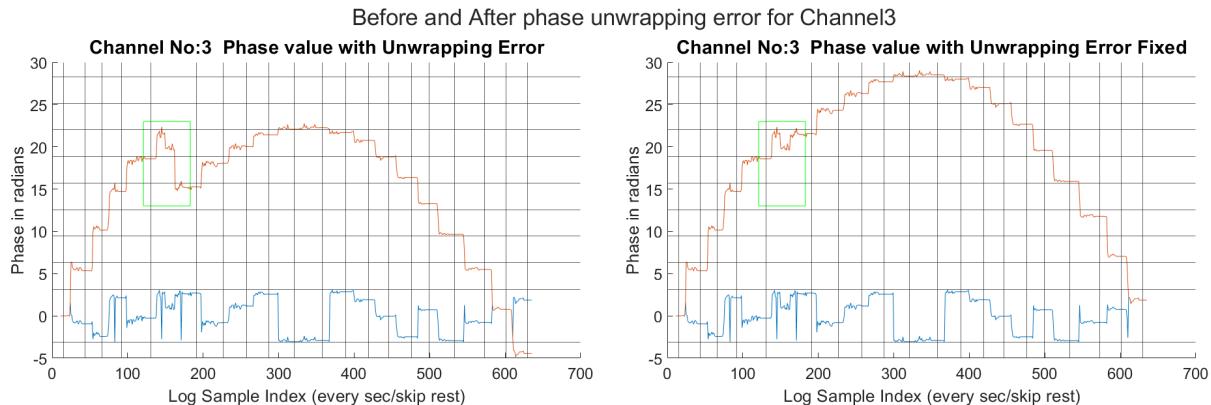


Figure 13: Unwrapping Error with focus on Channel3

To get a better overall view, provided all channels before and after correction side by side in Figure 14 and 15 respectively.

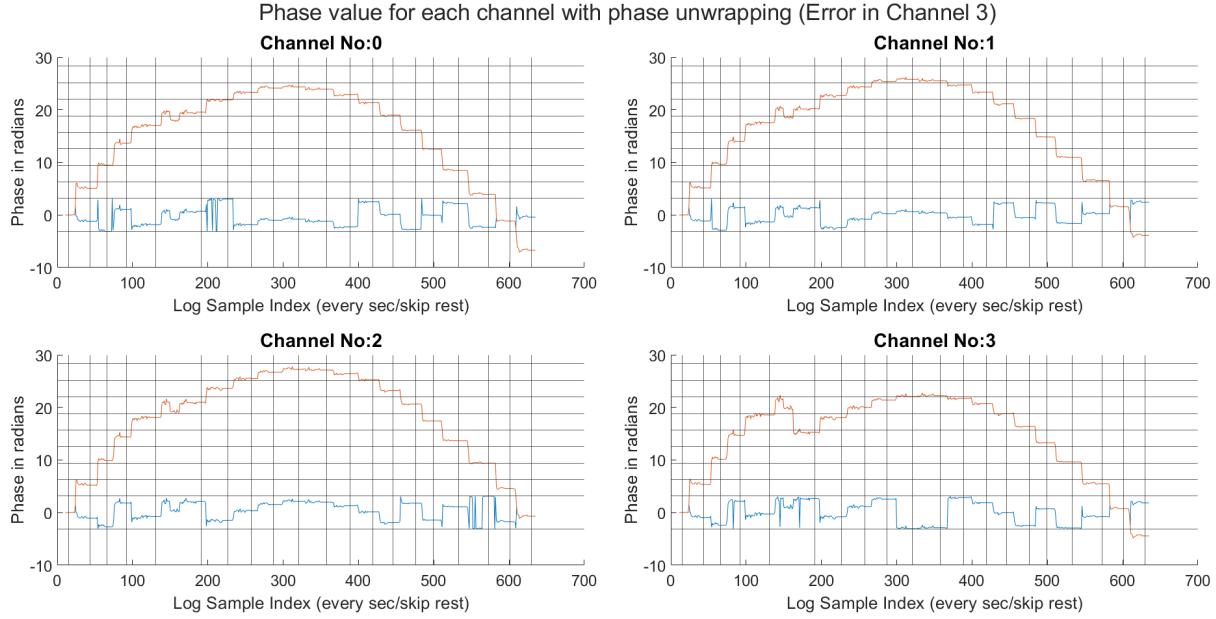


Figure 14: Unwapping Error in Channel3

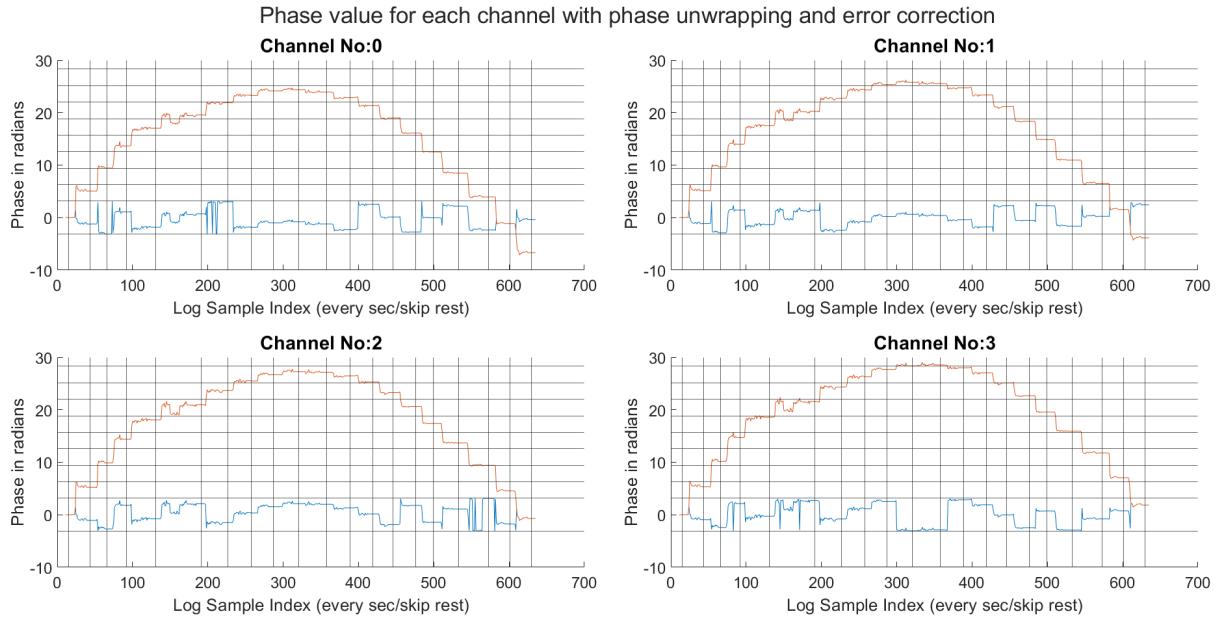


Figure 15: Unwrapping Error Fixed

3.6 LoS Separation

Reference Code : C4_Sage_Implementation.m

The successive peak removal as part of SAGE algorithm can be clearly seen in Figure 16

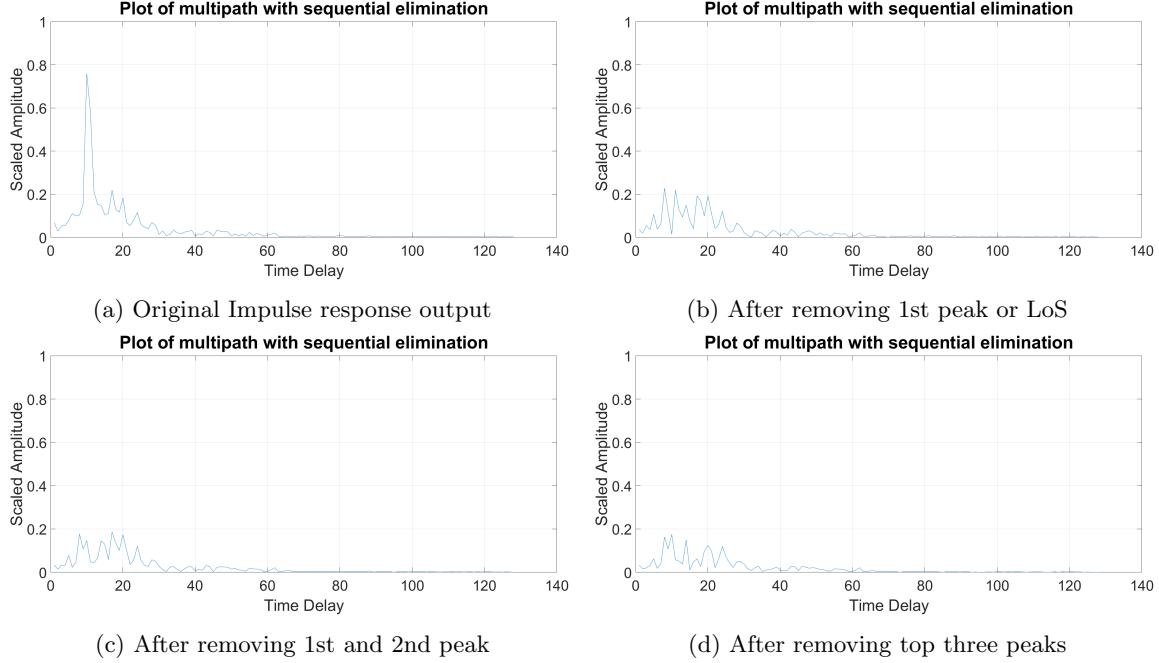


Figure 16: Sequential peak removal plots

After sufficient iterations, the retrieved LoS components from SAGE are shown in Figure 17. Each point corresponds to a phase value of the LoS signal, collected for different positions of the Tx Antenna.

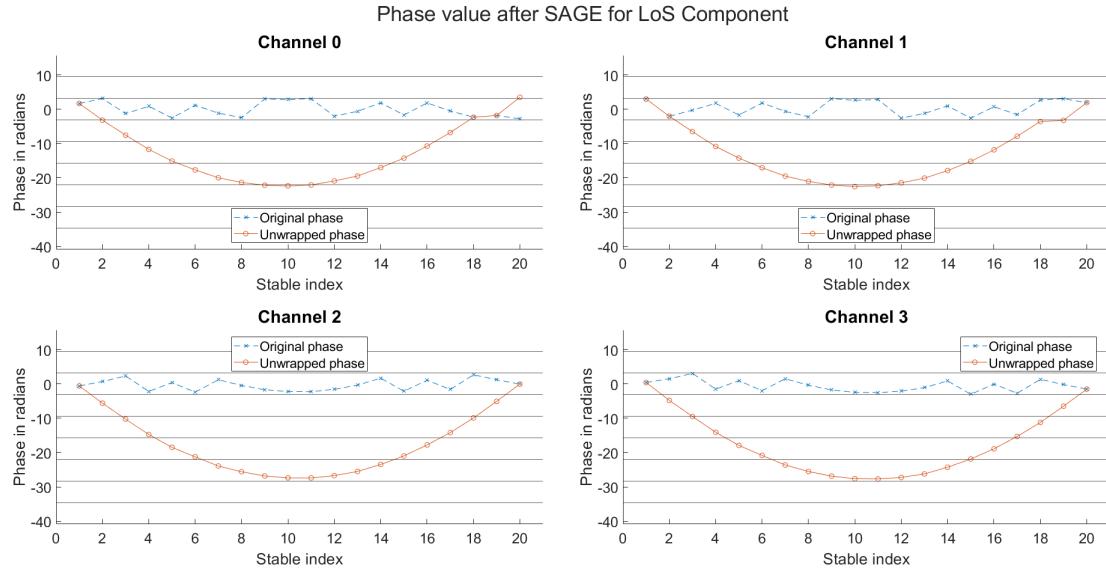


Figure 17: LoS Componenets after SAGE

3.7 Verifying B2B Measurement Data Reliability

The unwrapped phase is a very good way to measure the reliability of the measurement data from Table 1. We should expect the start and end phases to align (or almost align) with the unwrapping phase. Below Figure 18 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 the issue with phase difference too close to 2π .

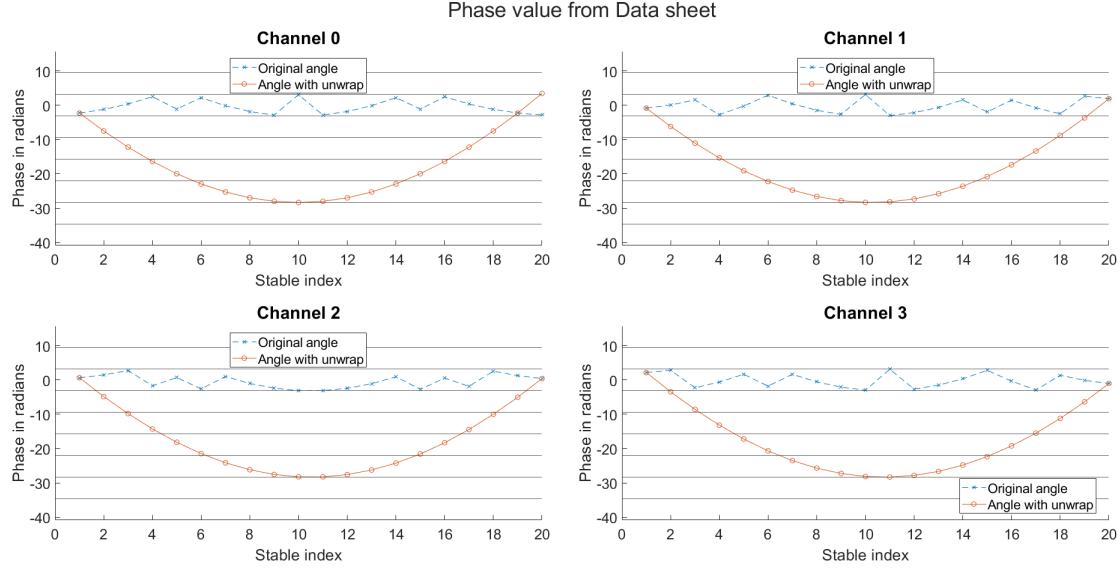


Figure 18: Phase from measurement data

To provide perspectives, the case where the provided distance between the antenna patches was incorrect (7cm instead of 2.6cm) is shown in Figure 19. You can clearly see the unwrapping phase problem and is a very good error spot-check method.

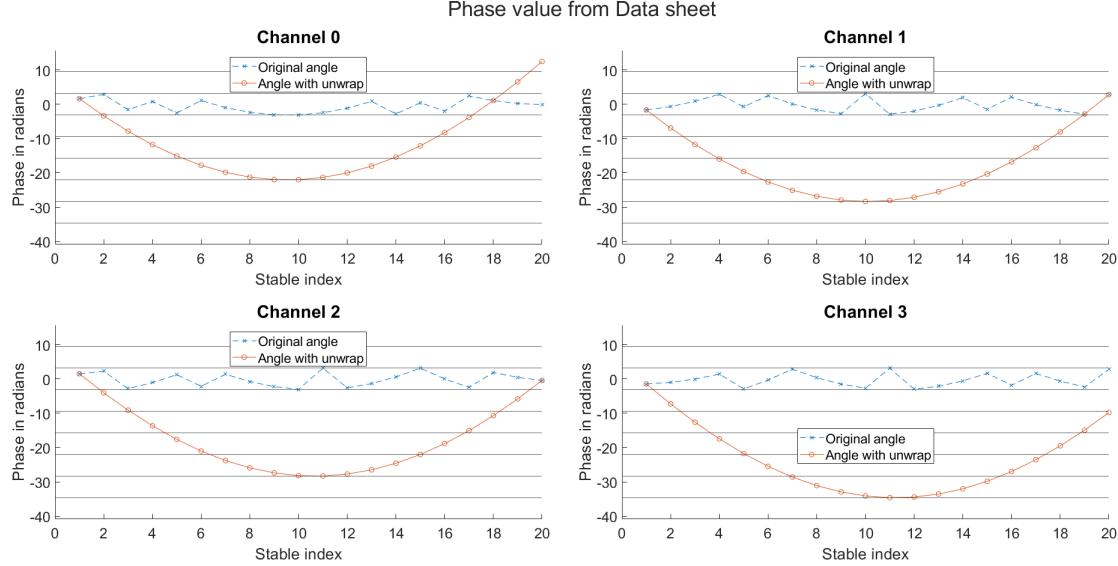


Figure 19: Phase from measurement data with measurement inaccuracy

3.8 Calibration

Reference Code : C5_FinalCalibration.m

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

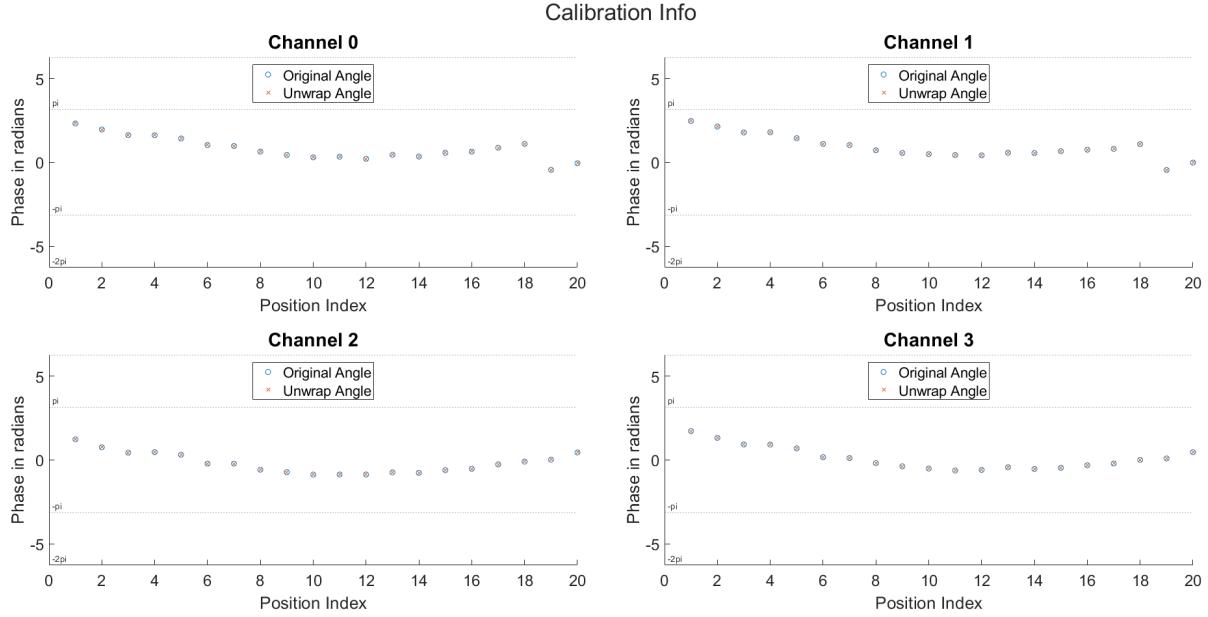


Figure 20: Final Calibration Results

A near-straight horizontal line is the best case for calibration results, but it appears there is a slight non-linear phase behavior as the Tx moves toward 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.

3.9 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 (physical measurement data).

The output from the SAGE algorithm contains phase contributions due to propagation as well as the Tx/Rx components. Taking the calibration results (section 3.8) and extracting this out from SAGE results (section 3.6), we get the phase contribution from the channel or the channel response. We can then feed this information into the MUSIC algorithm to verify estimated DoA accuracy.

The algorithm expects a semi-positive definite matrix to perform decomposition and operates on the eigenvectors. 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 a vector of size 4.

We deal with complex steering vectors giving us a complex covariance matrix, and directly using the matrix as input to the music function results in failure for some position indexes. We perform a few operations before feeding the covariance matrix to the music function. To get a better understanding of what we need to do, let's look at the eigenvalue and eigenvector for one sample of the Covariance Matrix.

$$\text{Eigen Value Matrix } (4 \times 4)$$

$$\begin{bmatrix} -6.66786213940145 \times 10^{-16} & 0 & 0 & 0 \\ 0 & -3.87977807963586 \times 10^{-16} & 0 & 0 \\ 0 & 0 & 5.55632997410900 \times 10^{-17} & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$\text{Eigen Vector Matrix } (4 \times 4)$$

$$\begin{bmatrix} +0.153528580240315 - 0.458507933915201i & +0.151031093319364 + 0.137858318176127i & \dots \\ +0.467280405884219 + 0.205066339202291i & +0.181743253205344 - 0.244153006866384i & \dots \\ +0.217914399258811 - 0.464971134494054i & -0.161224357677090 + 0.623675472096618i & \dots \\ +0.492048195883331 + 0.0000000000000000i & +0.671251818701089 + 0.0000000000000000i & \dots \\ \dots + 0.509921879945141 + 0.462994404825617i & -0.109702063159603 + 0.487817032644953i & \dots \\ \dots + 0.442354921515545 - 0.448639767927753i & -0.492045560963922 - 0.088832234778255i & \dots \\ \dots + 0.105299536258310 - 0.245484782305216i & +0.052411702845505 - 0.497245425725400i & \dots \\ \dots - 0.239394170393599 + 0.0000000000000000i & +0.5000000000000000 + 0.0000000000000000i & \dots \end{bmatrix}$$

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 that make the matrix **not** semi-positive definite even though they should be. There are a 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 the music function works. Start with considering all 15 places, and keep reducing to 14, 13, ... until music function usage doesn't complain about the matrix.

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

After performing the necessary value adjustments, the DoA estimates are shown in Figure 21

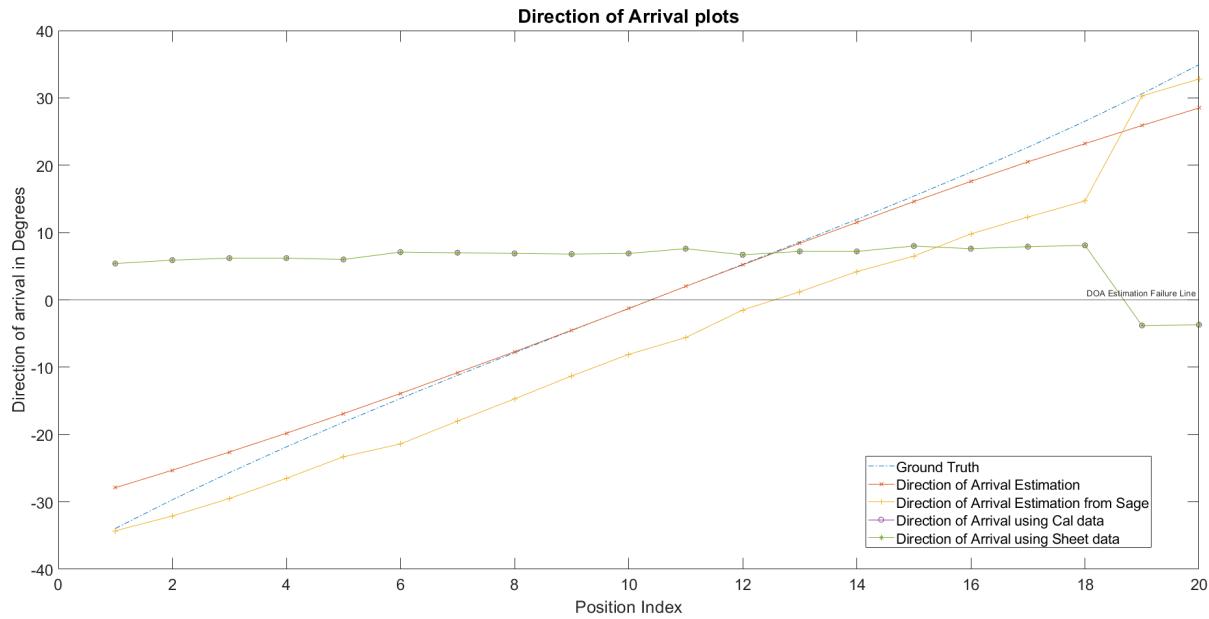


Figure 21: DoA Plots for various signals

4 Conclusion

4.1 Results

As shown in section 3.8 and 3.9, we were able to perform calibration reliably using wide band signal which is very useful to perform reliable channel estimations. Referring to Figure 21, we see good results near the mid positions, but it deviates towards the ends. I suspect that the reason for this deviation is the bench alignment is not parallel to the patch antenna surface plane (let's call it “Rx plane”).

While collecting ground truth data for verifying DoA estimation results, we rely on simple trigonometry. The shortest distance from the bench row to the Rx plane is assumed to be fixed and we use Pythagoras theorem to calculate the LoS distance. If the bench row and Rx plane are not exactly parallel you will observe the error as seen in both Figure 21 and Figure 20. The results in both plots rely on measurement data explaining the bend around the extremes. In short, we have an error in the ground truth, stemming from error in measurement data due to error in table alignment.

The 2nd observation is the jump in the phase for Channel 0 and 1 for the two right most points as seen in 20, and I suspect it is the ambiguity phase change when its very close to 2π .

4.2 Room for Improvements

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.

Limit minimum and maximum phase difference for adjacent Tx positions

Apart from the critical aspects of the setup requirements discussed in section 3.1, there are some non-critical considerations that will help with the 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 wavelengths (or 1.8π 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 at least 0.1 wavelengths (or 0.2π in radians) to reduce the impact of human error or noise on calibration data. The wavelength is approximately 5.24cm, and we expect the error to not exceed 0.5 cm, hence the 10% margins.

Including exact mid Tx placement

One important consideration (to ensure lower error in the calibration data around mid) is while placing the Tx antenna in various positions, we have an absolute mid position also. Having symmetry in placement gives us pair of observations/measurement data on either side of the mid position; to perform more corrective action if necessary. Also, the phase at both extreme positions (either side of the mid) will be the same, making it easy to perform quick spot checks. 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 an odd number of positions in total (mid + equal positions on either side). The other reason for having a mid position is to ensure the largest possible phase difference from mid to any adjacent position.

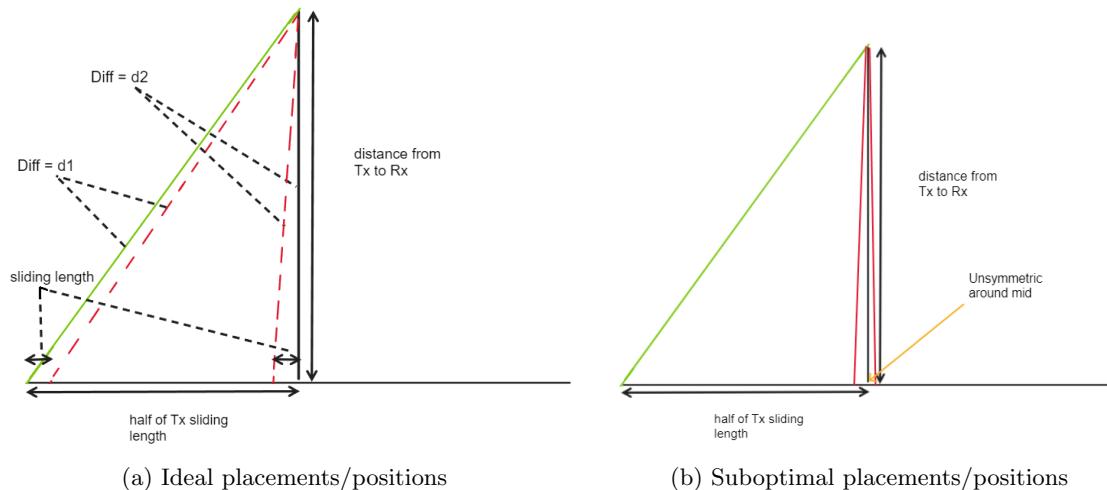


Figure 22: Visual understanding of min/max difference d_2/d_1

Mathematically friendly numbers

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 a number of stationary points as 11 (5 on each side and 1 midpoint) 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 that can result in non-terminating decimals. I will call

these "mathematically friendly" numbers, for future reference.

Selecting Rx antennas with larger separation

To obtain a better DoA estimate and ground truth, it might help to have a larger Rx antenna separation to have better field view, one suggestion provided in Figure 23. This will require adjustment of the mid position, as shown in the same figure but the benefits are significant.

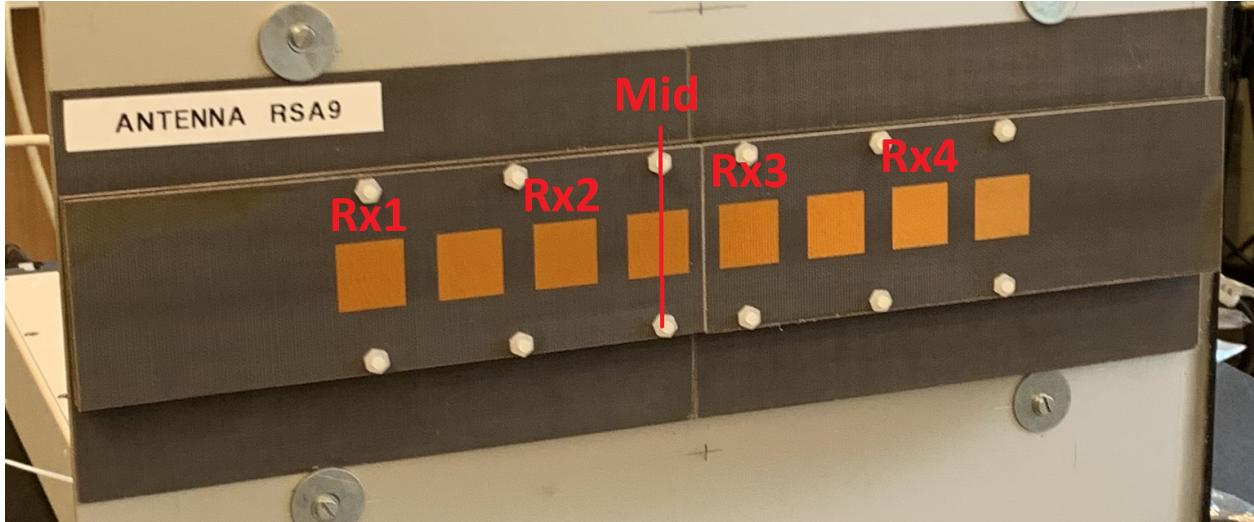


Figure 23: Recommended Rx Antenna selection

Near perfect bench alignment

As indicated in the previous subsection, the reason for deviation from ground truth at the extremes was due to the misalignment of benches with respect to the patch antenna plane. A more precise method is recommended to ensure no errors in the ground truth.

4.3 Recommended Setup Requirement for better calibration data estimates

If utilizing the same USRP configuration as earlier (ArrangementCalculator.m), we should utilize a wider Rx antenna separation (Figure 23f), set the separation between Tx/Rx to around 2.0m, set 16cm as the separation between sliding positions, and 64cm as the maximum length (which corresponds to 4 positions) on either side of mid. This ensures we have a 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 the 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

$$a(\psi_i, \theta_j) = e^{j\omega_o \tau_i(\theta_j)} \quad (13)$$

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

θ_i : direction of arrival such as value of azimuth and elevation angle in 3D space

τ_i : time delay which is a function of θ_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.

Requirement 1 : Sensor and source in a single plane

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

Requirement 2 : Wavefront follows far field approximation

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

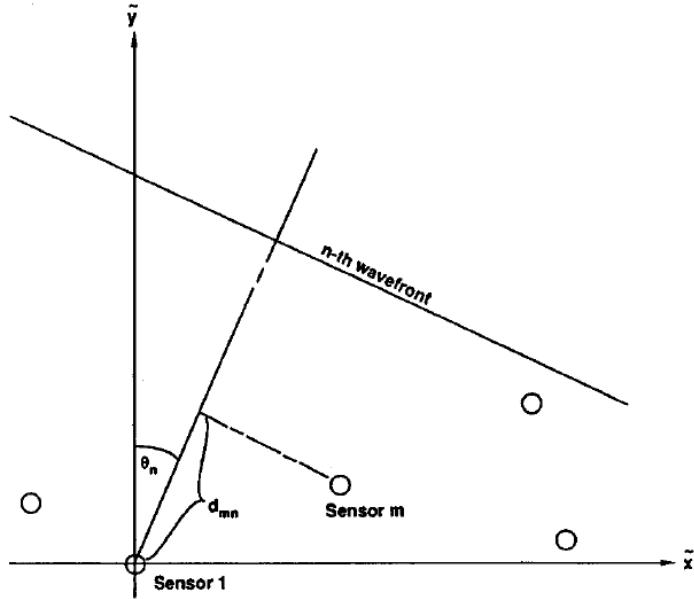


Figure 24: Problem geometry [5]

Requirement 1 and 2 simplify the calculations for relative time delays of received signal across sensors. Figure 24 will help understand how the time delays are derived. Assuming Sensor 1 as a reference, the delay of the signal from source n on the sensor m can be expressed by the below equation

$$\tau_m(\theta_n) = -d_{mn}/c = -\frac{1}{c}(x_m \sin \theta_n + y_m \cos \theta_n) \quad (14)$$

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 ψ_m

θ_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,

$$\mathbf{a}(\boldsymbol{\psi}, \theta_j) = [e^{j\omega_o \tau_1(\theta_j)}, e^{j\omega_o \tau_2(\theta_j)}, \dots, e^{j\omega_o \tau_m(\theta_j)}] \quad (15)$$

Note, we have only represented the phase contributions based on the 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 θ , the complex envelope of the received signal vector,

$$\mathbf{r}(t) = \mathbf{M} \cdot \mathbf{a}(\boldsymbol{\psi}, \theta_j) \cdot s(t) + \mathbf{e}(t) \quad (16)$$

While performing experiments, what is known to us is the received signal $\mathbf{r}(t)$ and the complex envelope 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 θ is known. We will come back to how θ 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 16 for N samples over time, we get

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

$$\frac{1}{\alpha N} \sum_{t=1}^N \mathbf{r}(t) = \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j) + \frac{1}{\alpha N} \sum_{t=1}^N \mathbf{e}(t) \quad (17)$$

$$\mathbf{a}_m(\theta_j) = \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j) + \mathbf{n}_j \quad (18)$$

$\mathbf{a}_m(\theta_j)$: Steering vector for a calibration source at location θ_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 18 is a linear problem, but because \mathbf{n}_j is not a measurable quantity, we have to rely on stochastic approaches such as utilizing the probability density function. We can easily formulate the optimization equation (because noise is zero-mean Gaussian) but we need to ensure that signals 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 a 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.

$$\max_{\mathbf{M}, \psi} p(\mathbf{a}_m(\theta_1), \mathbf{a}_m(\theta_2), \dots, \mathbf{a}_m(\theta_n) | \mathbf{M}, \psi, \zeta^2) = \max_{\mathbf{M}, \psi} \frac{1}{(\pi \zeta^2)^{nm}} e^{-\frac{1}{\zeta^2} \sum_{j=1}^n \|\mathbf{a}_m(\theta_j) - \mathbf{M} \cdot \mathbf{a}(\psi, \theta_j)\|^2} \quad (19)$$

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 don't 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

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

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

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

The optimization problem (changed order of terms, outcome doesn't change) :

$$\min_{\mathbf{M}, \psi} \|\mathbf{A}_m - \mathbf{M} \mathbf{A}(\psi)\|_{\textcolor{red}{F}}^2 \quad (20)$$

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 a complex matrix, resulting in two unknowns (real + imaginary) for every element in the matrix. This gives us a total of $2m^2$ (real) unknowns.

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

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

$$2mn \geq 2m^2 + 2m \quad (21)$$

and also **A_m** needs to be full rank!

Requirement 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 a diagonal matrix instead of a full matrix thereby reducing the number of (real) unknowns from $2m^2$ to $2m$. Rewriting the inequality expression21 as

$$2mn \geq 2m + 2m \quad (22)$$

Requirement 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 are fed 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 a single distance between antenna patches reducing the ψ unknown contribution from $2m$ to m . Rewriting the inequality expression22 as

$$2mn \geq m + m \Rightarrow n \geq 1 \quad (23)$$

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 θ is known, and hence must be clearly listed as an assumption.

Requirement 5 : Direction of Arrival is known

Reason : Reduces the number of unknowns for a system of equations

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

Requirement 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.

Requirement 7 : Timing information can be accurately determine

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

B Far field approximation

Far-field sources result 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 the Rayleigh distance formula,

$$d_R = \frac{2L_a^2}{\lambda} \quad (24)$$

d_R : Separation between the source and sensor

L_a : Largest separation between the sensors

λ : 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 the 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 destructively combine with the LoS signal

2. Obstruction near LoS : Obstruction of the free path around LoS reduces the strength of the LoS signal due to Huygens–Fresnel principle of propagating waves.

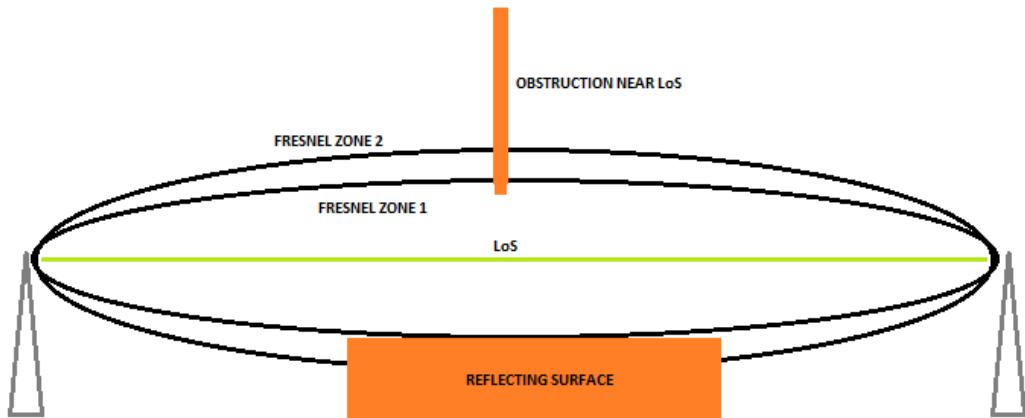


Figure 25: Fresnel Zone

Radius of the first Fresnel Zone is given by

$$F_1 = 8.656 \sqrt{\frac{d_R \times 10^6}{f_c}} \quad (25)$$

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 the 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 the below equation

$$s_q[n] = e^{-j\pi q \frac{n(n+1)}{\mathcal{N}_{zc}}} \quad (26)$$

\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. Another property; the constant amplitude signal makes it easy on the power amplifiers.

E Signalling Mode

As we are working with the UWB signal, we will use the OFDM scheme to transmit multiple narrow band signals but aggregate them to form a 400Mhz wide-band signal. Figure 26 27 below gives a summary of the transmitter and receiver processes, except the constellation mapping is replaced with zadoff chu sequence at the transmitter, with match filter in the receiver. This is because we are not sending data but a pure reference signal for proper synchronization, frequency error correction and estimation of channel/hardware responses. You can also refer to [7], [8] for better understanding.

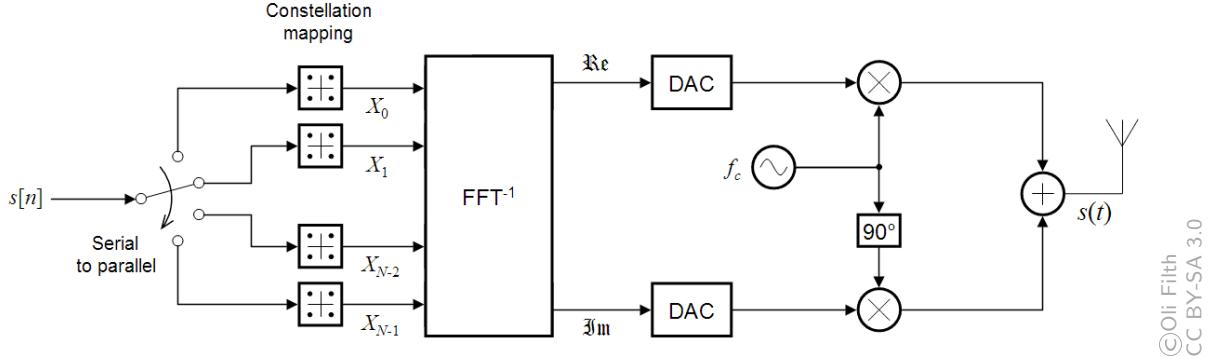
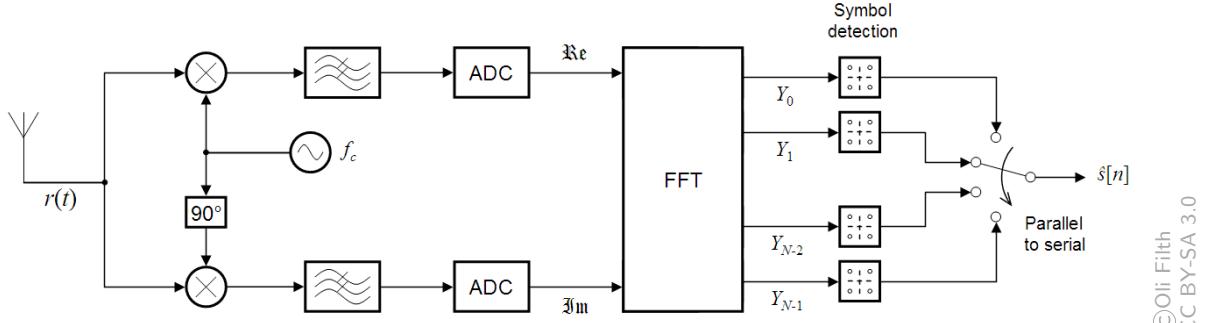


Figure 26: OFDM Transmitter

©Oli Filth
CC BY-SA 3.0



©Oli Filth
CC BY-SA 3.0

Figure 27: 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 26. 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 a zadoff chu's sequence length of 813 (the rest are padded with zeros)

F KNN Clustering Algorithm for Stationary Point Detection

We use a sliding window whose width increases from 1 sec to 13 secs with increments 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 end, moving 1 sec at a time.

If all the windows were considered, it would mean an acceptance rate of 100%, but this shouldn't be greater than RatioStableVsMovement. We need a way to penalize/deselect the windows with high jitter. What we have from the 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 the 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 acceptable, but if higher than the RatioStableVsMovement percentage, we will need to tighten the requirement by lowering down SumDLimit value (we reduce by a 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 a higher window width, if there is any instance where SumDLimit had to be lowered from the 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 a 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 stationary 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 stationary point across all channels.

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 ¹ data space rather than maximizing the intractable likelihood function for the measured data. EM method in the 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

θ : Unknown parameters

t : iteration count

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

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

¹ “Unobserved Complete” implies (“Unobserved”) not all variables that describe 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 changes in notations to compare with EM method. It would be good to review the EM Method covered in the previous Appendix G

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 :

$$\theta_{\mathbf{S}^t}^{t+1} = \max_{\theta_{\mathbf{S}^t}} Q(\theta_{\mathbf{S}^t}; \theta^t) \quad (27)$$

$$\theta_{\tilde{\mathbf{S}}^t}^{t+1} = \theta_{\tilde{\mathbf{S}}^t}^t \quad (28)$$

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 θ , represents selected parameters for optimization

$\tilde{\mathbf{S}}$: Index set which is a subset of θ , 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 $\theta_{\mathbf{S}}$

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

$$f(y, x; \theta) = f(y|x; \theta_{\tilde{\mathbf{S}}})f(x; \theta) \quad (29)$$

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.

¹Semicolon 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.

² $\perp\!\!\!\perp$: Independent of

I Processes summary and source code details

To wrap our heads around the entire list of steps described earlier, below is the summary of various steps, the purpose, and source code location

Step1 : Setup Requirement of Tx and Rx antennas (section 3.1)

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

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

Step2 : Data Collection (section 3.2)

Purpose : Collect both physical measurements as well as OTA and B2B data required for calibration.

Source Code : USRP Channel Sounder (Measurement details are saved in DATA SHEET)

Step3 : Frequency Error Detection and Correction (section 3.3)

Purpose : Resolve frequency errors prior to the calibration process.

Source Code : C1_FrequencyErrorAndCorrection.m

Step4 : Stable stationary point selection for Channel Estimation (section : 3.4)

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 : 3.5)

Purpose : Catch and rectify phase unwrapping errors

Source Code : C3_OutlierDetectorAndPhaseCorrector.m

Step6 : LoS Separation (section : 3.6)

Purpose : Use SAGE algorithm to separate out LoS

Source Code : C4_Sage_Implementation.m

Step7 : Calibration (section : 3.8)

Purpose : Perform calibration to determine the hardware response

Source Code : C5_FinalCalibration.m

Step8 : MUSIC Algorithm for verifying calibration data reliability (section : 3.9)

Purpose : Estimate DoA on estimated channel response using calibration data; use ground truth

Source Code : C6_MusicImplementation.m

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, 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.