Hochschule für angewandte Wissenschaften München
Alexander Edelemann, supervised by Adam Tankielun

Project Report
June 30, 2025

MIMO-OTA Antenna evaluation

Abstract **This report presents an investigation into the simulated radiation patterns of two antenna pairs, focusing on their relative performance in a 2×2 MIMO system. It details the methodologies used to derive an SNR difference metric that serves as a quantitative measure of comparative performance. In addition to the analysis, the report functions as a comprehensive guide to the underlying codebase, ensuring that future contributors can efficiently understand, reproduce, and extend the project.**

CONTENTS

1 TWO PROBE ANTENNA MATRIX EVALUATION THEORY

We consider a 2×2 MIMO system expressed as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \tag{1}$$

Or in compact matrix form:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

where $\mathbf{y}$, $\mathbf{x}$ are the received and transmitted signal vectors respectively, $\mathbf{H}$ denotes the 2×2 channel matrix, and $\mathbf{n}$ is the additive noise vector.

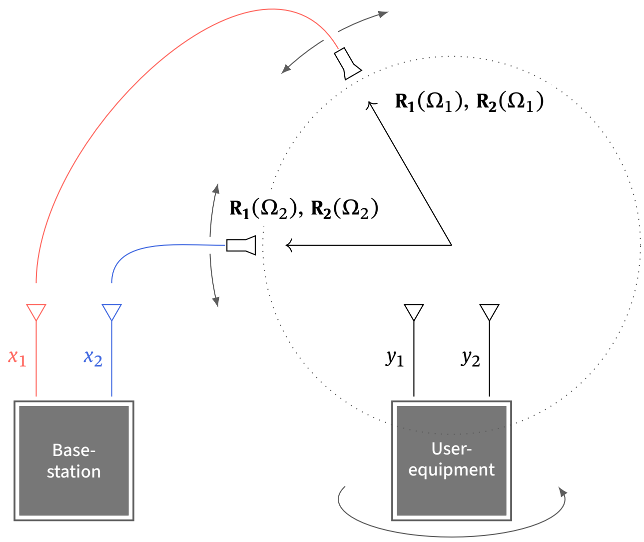
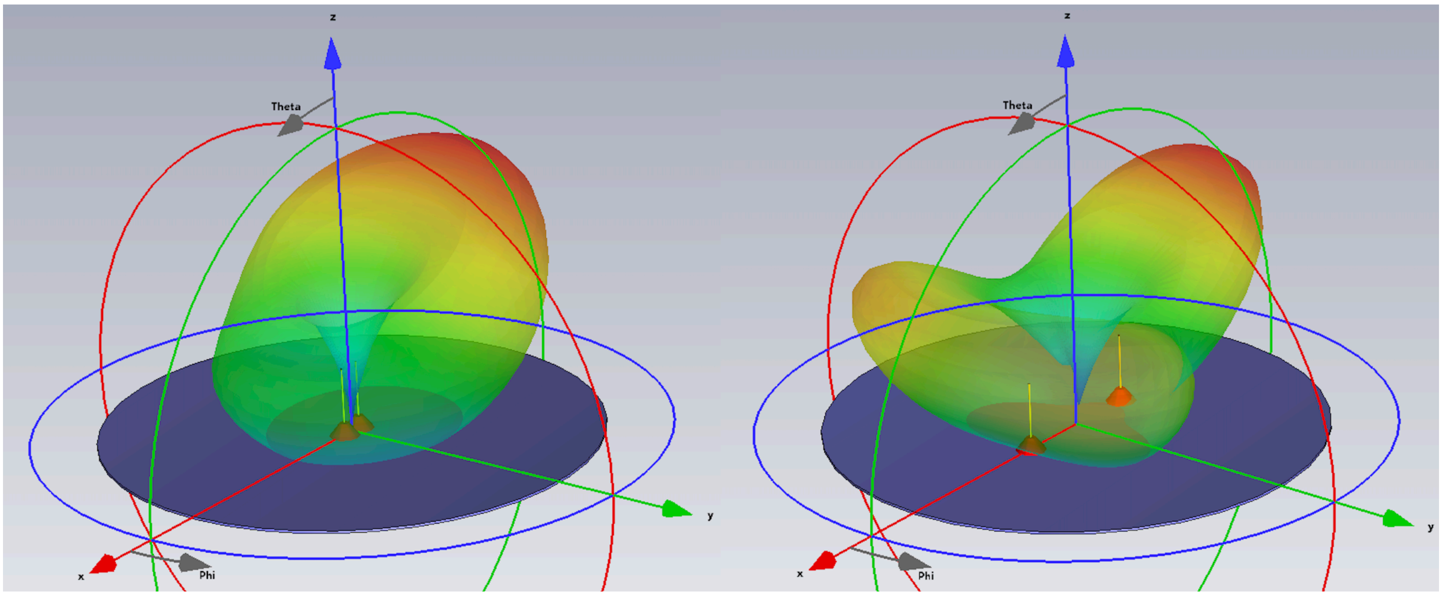


Figure 1 | **Schematic illustration of the 2-channel method.** A static MIMO 2×2 matrix can be computed for each distinct pair of Ω_1, Ω_2.

Our objective is to characterize how the user equipment (UE) antenna array influences MIMO performance. A generalized MIMO over-the-air (OTA) model based on the two-channel approach is illustrated in Figure 1, where Channel 1 and Channel 2 are represented in red and blue, respectively. Here, $\mathbf{T_1}$ and $\mathbf{T_2}$ are vector angular distributions of the electric field incident on the antennas of the UE created by radiation of base station (BS) signals x_1 and x_2 respectively. Using the Field components $T_{H,n}$ and $T_{V,n}$ of the BS antenna radiation pattern the vectors can be expressed as:



(a) "Bad" antenna.

(b) "Good" antenna.

Figure 2 | Simulated radiation pattern of the antenna pairs under investigation. The antenna pairs are distinguished by the spacing (27.5 mm and 170 mm) between the individual monopoles. Right antenna monopole pattern shown for "bad" (a) and "good" (b) antenna pairs. Patterns for the monopoles of each pair are symmetrical, hence only one of each is shown.

$$\mathbf{T_1}(\varphi,\theta) = T_{H,1}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{H} + T_{V,1}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{V} \tag{2}$$
$$\mathbf{T_2}(\varphi,\theta) = T_{H,2}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{H} + T_{V,2}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{V} \tag{3}$$

where: - $\hat{\mathbf{a}}_\mathbf{H}, \hat{\mathbf{a}}_\mathbf{V}$ are orthogonal unit vectors associated with horizontal (H) and vertical (V) polarization. Similarly, the receiving patterns of the UE antennas are defined as:

$$\mathbf{R_1}(\varphi,\theta) = R_{H,1}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{H} + R_{V,1}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{V} \tag{4}$$
$$\mathbf{R_2}(\varphi,\theta) = R_{H,2}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{H} + R_{V,2}(\varphi,\theta)\hat{\mathbf{a}}_\mathbf{V} \tag{5}$$

Using the above defined values, we can create static channel equations in accordance to (1) as:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R_1}(\Omega_1)\mathbf{T_1}(\Omega_1) & \mathbf{R_1}(\Omega_2)\mathbf{T_2}(\Omega_2) \\ \mathbf{R_2}(\Omega_1)\mathbf{T_1}(\Omega_1) & \mathbf{R_2}(\Omega_2)\mathbf{T_2}(\Omega_2) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \tag{6}$$

where a distinct set of φ and θ is denoted as Ω. If we restrict our transmission vectors $\mathbf{T_1}$ and $\mathbf{T_2}$ to be unit vectors and define their polarization state to be p and q respectively, the channel matrix H reduces to:

$$\mathbf{H} = \begin{bmatrix} R_{p1}(\Omega_1) & R_{q1}(\Omega_2) \\ R_{p2}(\Omega_1) & R_{q2}(\Omega_2) \end{bmatrix} \tag{7}$$

This matrix only contains the field components of the receive antenna pair radiation pattern. Our subsequent analysis will be based on this matrix.

[1]

2 SIMULATION DATA ANALYSIS

2.1 CST-SIMULATION DATA

The far-field radiation patterns of two antenna pairs were simulated using CST Studio Microwave. These pairs differ in the spacing between their individual monopoles. We refer to them as the "*good*" pair (170 mm separation) and the "*bad*" pair (27.5 mm separation). The radiation pattern of each monopole in the "good" pair is shown in Fig. 2. Simulations were conducted at a fixed frequency of 824 MHz with a radial resolution of 1°. The exported data is stored in `.txt` files and include the absolute value and phase of the E-field in φ and θ polarizations.

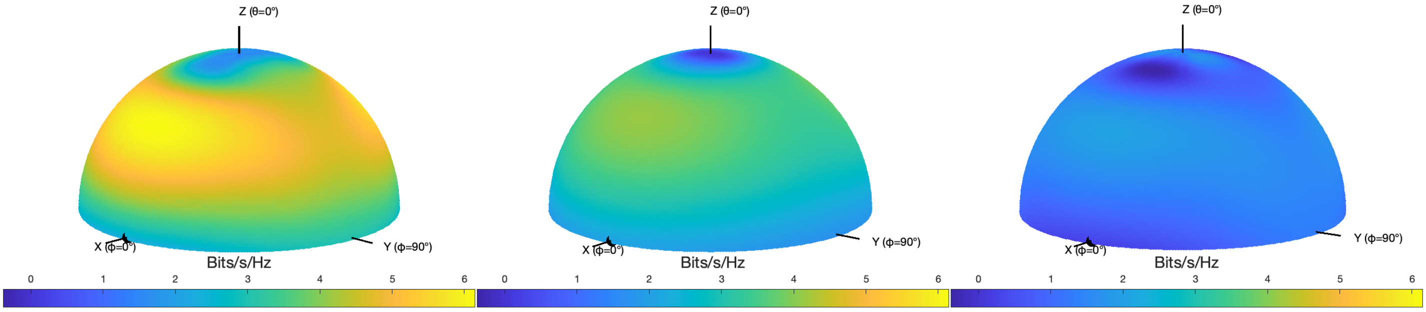
2.2 COLOCATED CAPACITY

Our first analysis will concern the complete dataset. Therefore, similar to basic anechoic chamber measurements, we regard the case of both test antennas being located at the exact same position. That is $\Omega_1 = \Omega_2$. We will refer to this case as "colocated". The metric we are interested in is a capacity value C. Assuming additive white Gaussian noise (AWGN), 2×2 MIMO capacity is given by:

$$C = \log_2\left(\det\left(\mathbf{I} + (SNR/2)\mathbf{H}\mathbf{H}^\dagger\right)\right) \tag{8}$$

For this purpose, we require a gain matrix, which we will compute using (7) at each individual Ω. The function to compute this matrix is implemented in the file `calculateH_ant.m`. It supports both H/V and $+45°/-45°$ polarization states. If we inspect any of the equations (2) through (5), using trigonometry, the $+45°/-45°$-polarization states are expressed as:

$$\hat{\mathbf{a}}_{+45} = \frac{1}{\sqrt{2}}(\hat{\mathbf{a}}_\mathbf{H} + \hat{\mathbf{a}}_\mathbf{V}) \tag{9}$$
$$\hat{\mathbf{a}}_{-45} = \frac{1}{\sqrt{2}}(\hat{\mathbf{a}}_\mathbf{H} - \hat{\mathbf{a}}_\mathbf{V}) \tag{10}$$



(a) "Good" antenna. **(b)** "Bad" antenna. **(c)** Difference.

Figure 3 | Colocated Capacity. Capacity calculated from channel matrix with both probes (see Fig. 1) colocated ($\Omega_1 = \Omega_2$).

For the case of colocated antennas, changing the polarization will be the same as applying a rotational matrix on the antenna matrix. Hence, it will not affect the capacity value, as it determinants stays unaffected. The case $\Omega_1 \neq \Omega_2$ however, yields a differently conditioned matrix resulting in a different capacity value. In MATLAB, the following function were implemented to compute our desired results:

- `computeCompleteH_ant.m`: This function takes as input arguments both files of an antenna pair and polarization-directions of the two test antennas. It discards all data located on the southern hemisphere and returns a cell array containing the **H**-matrices at all remaining points.
- `computeCompleteAntennaCap.m`: This function takes as input a cell array of **H**-matrices and a linear SNR value. It returns the computed capacity for each matrix according to (8).
- `scatterPlotCaps.m`: Among several other input arguments, this function takes the computed capacities and generates a 3d scatter plot in which the the capacity values are mapped to a color-range.

The results of the capacity calculation at each individual point for both antennas is visualized in a 3d-scatter plot depicted in Fig. 3. As expected, the "good" antenna is observed to have overall higher values.

2.3 SIMULATION OF DIFFERENT TEST ANTENNA CONSTELLATIONS

The next step of our investigation involves the extraction of a subsets of the complete data. These data-subsets will comprise of a specific number of different constellations which is a set of positions at which each of the test antennas can be located. We will refer to such a subset as constellation-set. These constellation-sets are generated as `.txt`-files using Python which are then processed in MATLAB. The files contain the angles φ, θ, and polarization state assigned to each of the two test antennas. Each line in the file represents one constellation:

```
thetaP; phiP; polP; thetaQ; phiQ; polQ
-----------------------------------------
60; 324; H; 90; 324; V
30; 180; H; 90; 270; V
...
```

The generation of these files is implemented in the Python-script `gen_constellations.py`. We define three distinct types of constellation sets. Fig. 4 illustrates an example of each distinct set. The visualization is done by the MATLAB-function `plotExtractedConstellation.m`. The properties of the constellation-sets are defined as follows:

1. Random This set consists of randomly but uniformly distributed points over the upper hemisphere. Each constellation in this set consists of two randomly chosen points. The generation of such points is implemented inside the function `uniform_hemisphere_points(n)`. The theory on which the implementation is based on is that in spherical coordinates, points uniformly distributed on a sphere require:

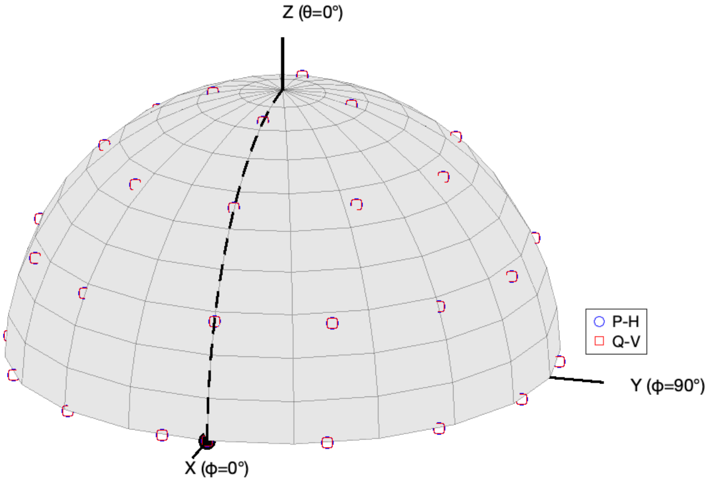
- φ to be uniform between 0 and 2π
- θ to follow a sine distribution - specifically, $\cos(\theta)$ must be uniform between -1 and 1.

The implemented algorithm generates random values $u \in [0, 1]$ and $v \in [0, 1]$ then maps $\varphi = 2\pi \cdot u$ (uniform distribution), $\theta = \arccos(2v - 1)$. The southern hemisphere points are then filtered out. This constellation-set is scalable by adjusting the number of points. Polarizations $+45$, -45, H and V are possible and are randomly assigned to constellation points. Hence, some constellations will contain points with non-orthogonal polarization states or even with the same polarizations.

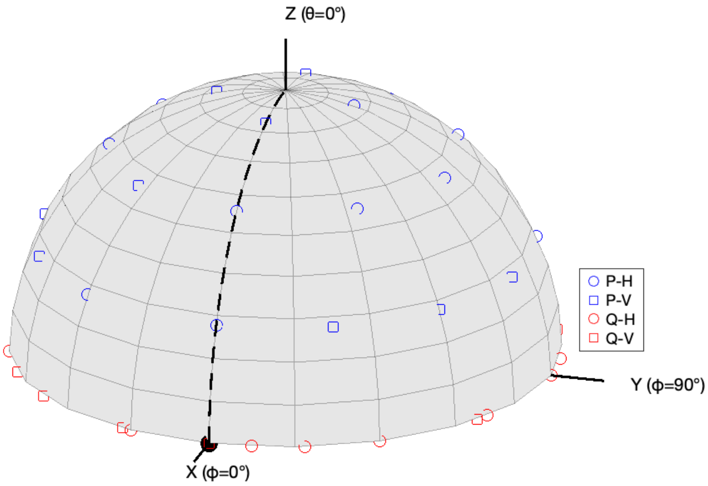
2. Gantry-Based Probing All points, are distributed over the hemisphere with nearly uniform angular density. In this constellation-set the two test antennas are colocated with orthogonal polarizations. The generation of such points is implemented inside the function `theta_dependent_phi_points(delta)`. Scaling can be done by adjusting $\Delta\varphi(\theta = 90)$. For better readability this value will be denoted as just Δ. The implemented algorithm is based on the following formula:

$$N_\varphi(\theta) = 1 + \text{int}\left(\left(N_\varphi(90) - 1\right)\sin(\theta)\right) \tag{11}$$

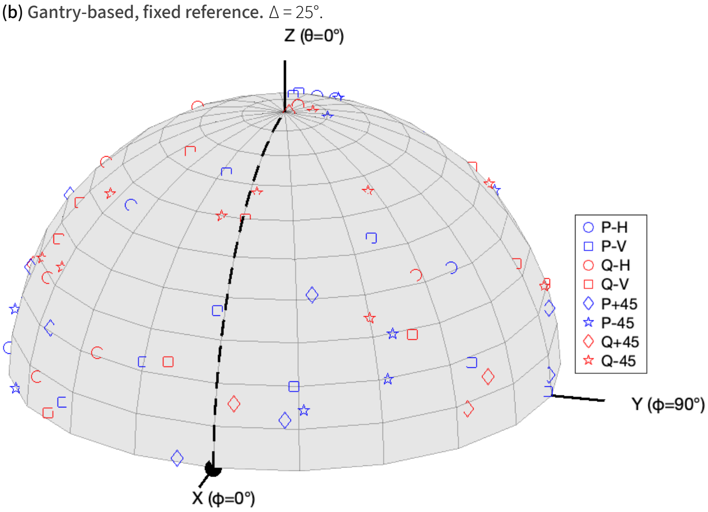
3. Gantry-Based Probing with Fixed Reference In this setup, one antenna (Probe 2) is fixed at $\theta = 90°$, while the other (Probe 1) moves across the constellation defined in Set 2 only all points at $\theta = 90°$ are discarded. Probe 2 is opposite with respect to azimuth ($\Delta\varphi = 180°$) to Probe 1. In our analysis we restrict polarizations to H and V which are randomly assigned to constellations points. As in the random constellation-set, it is possible for both points in one constellation to have the same polarization.



(a) Gantry-based. Δ = 25°.



(b) Gantry-based, fixed reference. Δ = 25°.



(c) **Random.** Number of points: 41. This is the number of points resulting from Δ = 25° in (a).

Figure 4 | Visualization of the three distinct constellation sets. Constellation points in (a), (b) are generated using a fixed value for Δ, while the generation of (c) requires the total number of points.

2.4 MAPPING CONSTELLATION POINTS TO SIMULATION DATA

The resulting angles from the constellation calculations may not always match a specific point contained in the exported simulation data. Therefore, it is necessary to define a metric of how the constellation points are mapped to the simulation data. Here, our approach is straightforward, and involves the extraction of the shortest euclidean distance between constellation and simulation point. This is implemented in the MATLAB-function `getAntDataRow.m`.

2.5 SNR-BASED ANALYSIS

Using the constellation-sets outlined in the previous section, we proceed in our antenna characterization approach by investigating the SNR-distribution over all constellations at fixed capacity. To find an appropriate capacity value, we use the *random*-constellation-set consisting of 200 points and compute the average capacity at fixed SNR of 10 dB for the “good” antenna pair yielding a value of 5.89 Bits/s/Hz. Therefore, we choose 6 Bits/s/Hz to be our reference capacity. Finding the SNR value to achieve a particular target capacity value for a given **H**-matrix can be done iteratively. The algorithm we use is implemented in the MATLAB-function `findSNRForTargetCapacity.m`.

Furthermore, we would like to know how these SNR values compare with respect to the different antenna pairs. For this, we compare the a inverse average of the linear SNR values. This inverse average aims to promote the more “sensitive” constellations yielding lower SNR values. This computation is somewhat similar to the calculation of the total resistance of parallel resistors:

$$SNR_w = \frac{N}{\sum_i^N \frac{1}{SNR}} \tag{12}$$

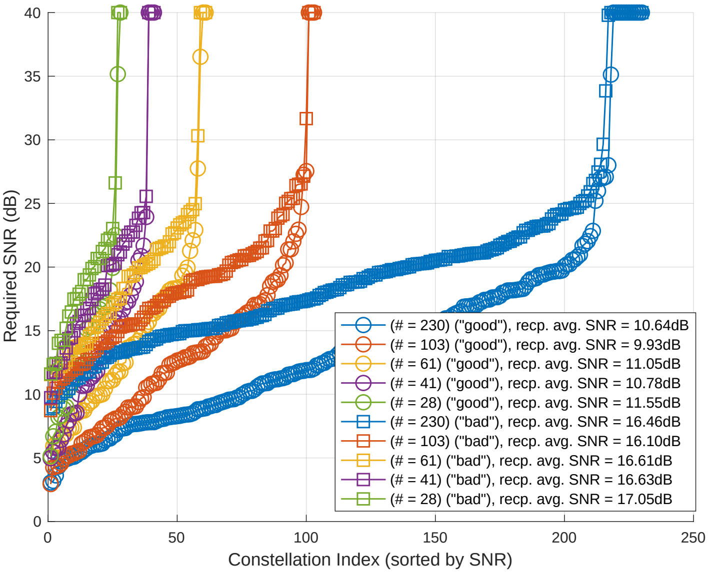
where N is the number of constellations of a given constellation-set and i is the index of the linear SNR values. In our implementation we also discard SNR-values of 40 dB and higher. The algorithm is implemented in the MATLAB-function `reciprocalAverageSNR.m`.

Moreover, we investigate how this distribution changes with the number of constellations in a given constellation-set. Fig. 5 illustrates this distribution for all of our constellation-sets. Additionally, for the random constellation set an SNR analysis has been performed (200 constellations per set) to investigate the effect of random polarizations on our average SNR result. We observed fluctuations of roughly 0.4 dB.

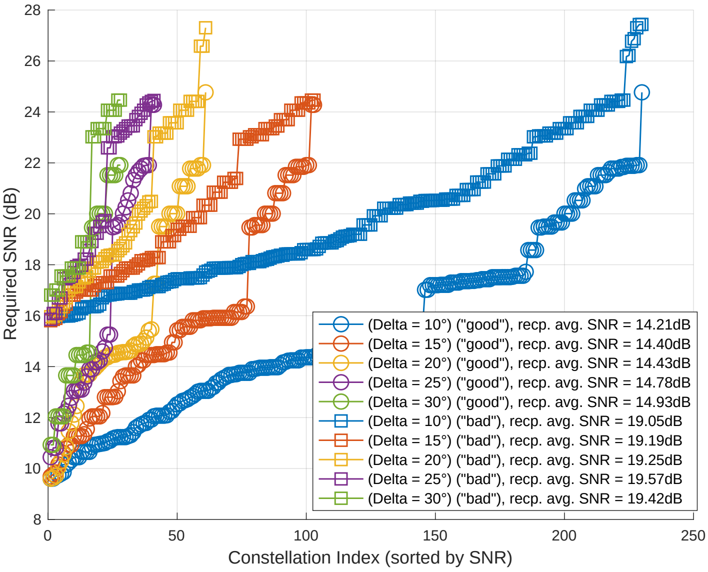
The plot is generated by the MATLAB-function `plotSNRDistributionForDifferentConstellationSets.m`. Additionally, it outputs a cell array of tables containing detailed data such as constellation angles and condition number sorted according to the plot. As stated earlier, for colocated probes, the channel capacity should not change whether the polarizations are chosen to be H/V or $\pm 45°$. To verify this, the “Gantry-Based” constellation-set was generated for both cases. The resulting data, saved in `snr_analysis_gantry_ based_pm45_delta.mat` and `snr_analysis_gantry_based_pm45_delta.mat` are in accordance with the theoretical expectations: all metrics remain identical except for the individual matrix elements, which differ due to the change in polarization basis.

2.6 3GPP-BASED CHANNEL MODELING APPROACH

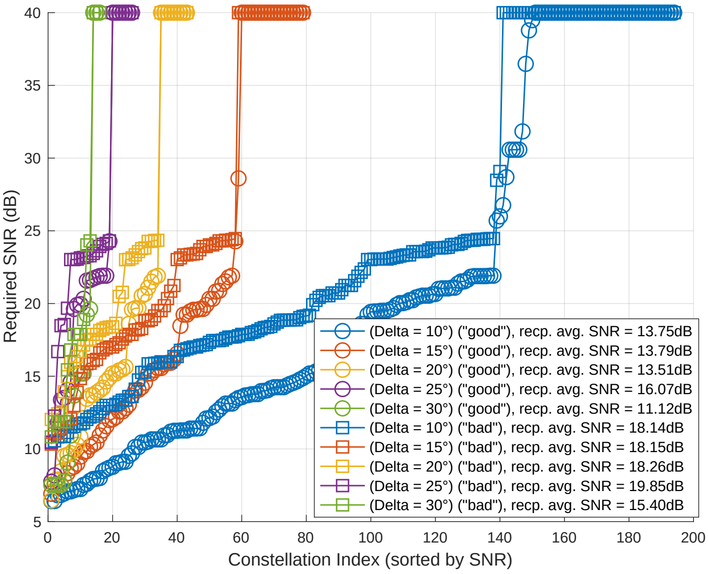
An alternative analysis of the antenna pairs was performed using an approach inspired by 3GPP channel modeling. In this method,



(a) Random: SNR difference for "good" and "bad" antenna pair: 5.28 dB (# = 230), 6.17 dB (# = 103), 5.56 dB (# = 61), 5.85 dB (# = 41), 5.50 dB (# = 28).



(b) Gantry-based: SNR difference for "good" and "bad" antenna pair: 4.84 dB ($\Delta\Phi = 10°$), 4.79 dB ($\Delta\Phi = 15°$), 4.82 dB ($\Delta\Phi = 20°$), 4.80 dB ($\Delta\Phi = 25°$), 4.50 dB ($\Delta\Phi = 30°$).



(c) Gantry-based, fixed reference: SNR difference for "good" and "bad" antenna pair: 4.39 dB ($\Delta = 10°$), 4.36 dB ($\Delta\Phi = 15°$), 4.74 dB ($\Delta\Phi = 20°$), 3.78 dB ($\Delta\Phi = 25°$), 4.28 dB ($\Delta\Phi = 30°$).

Figure 5 | SNR Required for 6.00 bits/s/Hz Capacity. SNR values equal to or higher than 40 dB are capped at 40 dB for better readability of the plot.

we construct a composite channel matrix $\mathbf{H}$ by aggregating multiple sub-matrices, each representing signal components arriving from distinct angular directions. While standard 3GPP models [2] require the knowledge of the both receiving and transmitting antenna patterns—allowing for accurate modeling of the angular departure and arrival of rays—we retain the simplifications introduced in section 1, where we defined our transmission antenna pattern vectors $\mathbf{T_1}$ and $\mathbf{T_2}$ to be orthogonal unit vectors, implying a homogeneous (isotropic) transmit antenna pattern. In standard 3GPP modeling, multipath components are grouped into clusters that follow scenario-specific angular distributions (e.g., Urban Micro, Urban Macro). Each cluster is characterized by properties such as delay, power, and a specific angular distribution of arriving and departing signal rays. Following a similar principle, our simplified model computes each component of the $\mathbf{H}$ matrix as:

$$H_{u,s,n} = \sum_{m}^{M} \sqrt{\frac{P_n}{M}} \left[R_{u,1}(\varphi_m, \theta_m) \quad R_{u,2}(\varphi_m, \theta_m) \right] \begin{bmatrix} \exp\left(\Phi_m^{j\theta\varphi}\right) & \frac{\exp\left(j\Phi_m^{j\theta\varphi}\right)}{\sqrt{\kappa}} \\ \frac{\exp\left(\Phi_m^{j\theta\varphi}\right)}{\sqrt{\kappa}} & \exp\left(\Phi_m^{j\theta\varphi}\right) \end{bmatrix} \begin{bmatrix} T_{s,1} \\ T_{s,2} \end{bmatrix} \tag{13}$$

Here, u and s denote the receive and transmit antenna indices, respectively—u can be mapped to p and q in eq. (7). While n is the cluster index, m iterates over the sub-rays within a cluster.φ_m and θ_m are the azimuth and elevation angles of arrival for each sub-ray, derived from base angles using:

$$\varphi_m(n) = \varphi_n \pm c_{ASA}\alpha_m \tag{14}$$
$$\theta_m(n) = \theta_n \pm c_{ZSA}\alpha_m \tag{15}$$

where α_m are angular offsets drawn from a normalized distribution (rms spread = 1), and c_{ASA}, c_{ZSA} define the angular spread per cluster (assumed equal across clusters). Cross-polarization effects are modeled via the parameter κ, derived from the cross-polarization ratio (XPR) X (in dB) as:

$$\kappa = 10^{\frac{X}{10}} \tag{16}$$

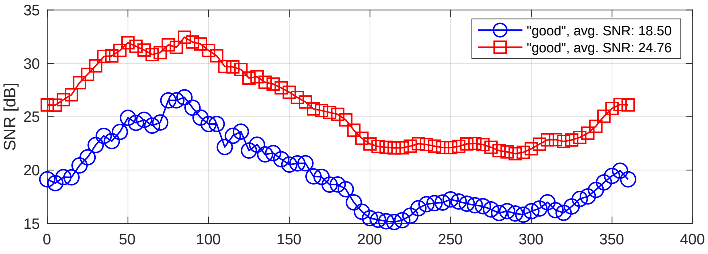
Each cluster has a total power P_n which is equally distributed over all sub-rays.

The composite $\mathbf{H}$ matrix is computed in MATLAB using the function `buildComposite3GPPSettingHMatrix.m`. It requires as input a file specifying the cluster characteristics in the following format:

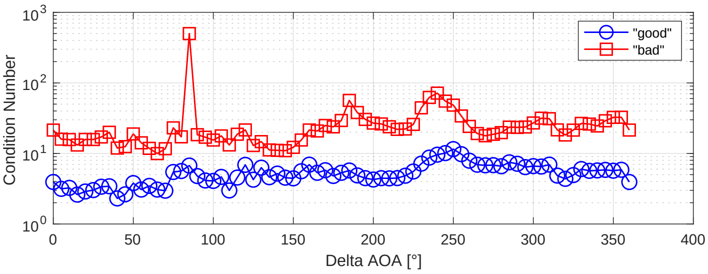
```
Cluster-No; Power [dB]; AOA(phi) [deg]; ZOA(theta) [deg]
1; 0; -175.004; 76.7603
2; -2.2185; -175.004; 76.7603
...
```

Moreover, azimuth rotation of the UE can be accounted for using the parameter `DeltaAoA`. The total power is normalized to 1 such that the results can be compared with the ones computed using (7).

An SNR analysis has been performed for one complete azimuth rotation in steps of 5° using 6 Bits/s/Hz as target capacity and



(a) SNR vs Δ_{AOA}. Average SNR difference: 6.26 dB.



(b) Condition number vs Δ_{AOA}.

Figure 6 | 3GPP-based channel modeling results. SNR values have been computed for target capacity of 6 Bits/Hz/s and for one complete rotation of the UE in 5° steps. Model parameters taken from 3GPP CDL-B.

input file `CDL_B.txt`, using the parameters of the Clustered Delay Line model CDL-B from 3GPP TR 38.901 (page 72). The model parameters are listed in Tab. 1 and the results are presented in Fig. 6. The inverse average SNR value for all rotational angles has been computed as well.

Cluster Index	Power [dB]	AOA [°]	ZOA [°]
1	0	-173.3	78.9
2	-2.2	-173.3	78.9
3	-4	-173.3	78.9
4	-3.2	125.5	63.3
5	-9.8	-88.0	59.9
6	-1.2	155.1	67.5
7	-3.4	155.1	67.5
8	-5.2	155.1	67.5
9	-7.6	-89.8	82.6
10	-3	132.1	66.3
11	-8.9	-83.6	61.6
12	-9	95.3	58.0
13	-4.8	103.7	78.2
14	-5.7	-87.8	82.0
15	-7.5	-92.5	62.4
16	-1.9	-139.1	78.0
17	-7.6	-90.6	60.9
18	-12.2	58.6	82.9
19	-9.8	-79.0	60.8
20	-11.4	65.8	57.3
21	-14.9	52.7	59.9
22	-9.2	88.7	60.1
23	-11.3	-60.4	62.3

Table 1 | 3GPP-based channel modeling input file parameters. SNR analysis results from these parameters are shown in Fig. 6.

2.7 USAGE OF MATLAB PROJECT

The complete MATLAB project is stored inside a folder called `Channel_capacity_estimation`. The folder structure is shown below:

```
🗁 Channel_capacity_estimation/
├── 🗁 Python_scripts/
│   ├── 🗋 gen_constellations.py
│   └── ...
├── 🗁 Constellations_files/
│   ├── 🗋 random_200_points.txt
│   └── ...
├── 🗁 Exports/
├── 🗁 3GPP_setting_files/
│   └── 🗋 UMi_CDL_B.txt
├── 🗁 Antenna_files/
├── 🗋 main.m
├── 🗋 calculateH_ant.m
└── ...
```

All calculations and plots presented above can be reproduced inside the `main.m`-file. It is subdivided into different sections using the two-percent-symbol-command under which a short description is given. Some results have been exported as well and can be found inside the `Exports`-folder. Python scripts are located in the `Python_scripts` folder. They can be executed from the terminal. The generation of specific constellation files for instance, can be done by properly adjusting the parameters and running the command `python Python_scripts/gen_constellations.py`. This will generate files which are stored inside the `Constellation_files`-folder.

REFERENCES

[1] "Two-Channel Method for OTA Performance Measurements of MIMO-Enabled Devices". In: Rohde & Schwarz. 2011. URL: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1sp12/1SP12_1e.pdf.

[2] "TR 38.901, Study on channel model for frequencies from 0.5 to 100 GHz". In: 3GPP (3rd Generation Partnership). 2022. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173.