Channel Charting: Model-Based Approaches

Amr Aly, *Student Member, IEEE*
CPCC, Department of EECS
University of California, Irvine, CA, USA
alyas@uci.edu

Ender Ayanoglu, *Fellow, IEEE*
CPCC, Department of EECS
University of California, Irvine, CA, USA
ayanoglu@uci.edu

***Abstract*—We present new ways of producing a channel chart employing model-based approaches. We estimate the angle of arrival θ and the distance between the base station and the user equipment ρ by employing our algorithms, inverse of the root sum squares of channel coefficients (ISQ) algorithm, linear regression (LR) algorithm, and the MUSIC/MUSIC (MM) algorithm. We compare these methods with the channel charting algorithms principal component analysis (PCA), Sammon's method (SM), and autoencoder (AE) from [1]. We show that ISQ, LR, and MM outperform all three in performance. ISQ and LR have similar performance with ISQ having less complexity than LR. The performance of MM is better than ISQ and LR but it is more complex. Finally, we introduce the rotate-and-sum (RS) algorithm which has about the same performance as the MM algorithm but is less complex due to the avoidance of the eigenvector and eigenvalue analysis and a potential register transfer logic (RTL) implementation.**

***Index Terms*—Channel charting, user equipment (UE), channel state information (CSI), MUSIC, PCA, SM, AE.**

I. Introduction

A channel chart is a chart created from channel state information (CSI) that preserves the relative geometry of the radio environment consisting of a base station (BS) and user equipments (UEs) [1]. This chart helps the BS locate the UEs (relatively), which can help in many applications such as handover, cell search, user localization, and more. In this paper we calculate the channel chart directly employing model-based approaches.

A. Background

1) Channel Models: Throughout this paper, we employ three channel models, namely Vanilla line-of-sight (LOS), Quadriga LOS (QLOS), and Quadriga non-LOS (QNLOS) [2], [3] as in the paper that introduced channel charting [1] so that we can compare our results with the approaches it introduced. We refer the reader to [4, Table 1] for system parameters used in this paper.

2) Angle of Arrival and Steering Vector : Assuming θ is the angle of arrival (AOA) to a BS array, we have the steering vector

$$\mathbf{A}(\theta) = (1, e^{j\pi\cos(\theta)}, e^{j\pi 2\cos(\theta)}, \ldots, e^{j\pi(N_R-1)\cos(\theta)})^T \quad (1)$$

where N_R is the number of receive antennas at the BS. This vector is essential in beamforming applications and in determining the AOA, as we will see later when we use the MUSIC algorithm.

This work is partially supported by NSF grant 2030029.

3) Measures for Channel Charting: Continuity and Trustworthiness: As in [1], we use continuity (CT) and trustworthiness (TW) as performance measures. CT measures if neighbors in the original space are close in the representation space. TW measures how well the feature mapping avoids introducing new neighbor relations that were absent in the original space. CT and TW are between 0 and 1, with larger values being better, see [1] for their formal definitions.

II. Estimating the Coordinates θ and ρ

We will use the symbol θ for the AOA and ρ for the distance between the BS and the UE. Estimating θ and ρ can happen concurrently as they do not depend on each other. In this section, we will first discuss how to estimate θ by using the MUSIC algorithm and then we will discuss our first three algorithms to estimate ρ.

A. Estimating θ Using MUSIC

To estimate the AOA, we employ the MUSIC algorithm [5]. The MUSIC algorithm is well-known and widely used for the estimation of the AOA. We wish to emphasize that in addition to the AOA, we need to estimate ρ, which we discuss next.

B. Estimating ρ

1) Estimating ρ Using ISQ: Our first proposal is a rather direct and simple approach. We calculate the square root inverse of the sum of CSI magnitudes for all antennas as

$$\rho = \frac{1}{\sqrt{\sum_{n=0}^{N-1} \mathrm{abs}(h_n)}}.$$

We refer to this algorithm as ISQ (inverse square root sum).

2) Estimating ρ Using LR: This is actually a learning-based, supervised approach where we assume we know the location of 256 (out of 2048) UEs and do a linear regression with the logarithm of the sum of CSI magnitudes for all antennas to find a and b in

$$\rho = aX + b \text{ , where } X = \log \sum_{n=0}^{N-1} \mathrm{abs}(h_n).$$

We call this algorithm the LR algorithm.

Algorithm 1 MUSIC Procedure for Estimating ρ

Calculate the CSI across antennas and subcarriers covariance matrix $\mathbf{R} = \mathbb{E}[\mathbf{h}\mathbf{h}^H]$
Get the eigenvectors and eigenvalues of $\mathbf{R}$
Separate system subspace $\mathcal{S}$ and noise subspace $\mathcal{N}$ by defining a threshold
Calculate $\mathbf{N}$ by concatenating the eigenvectors of $\mathcal{N}$
for $\rho = 0 : 1000$ in increments of 1 **do**
  Calculate vector $\mathbf{B}(\rho)$
  Calculate the PMF$(\rho) = \frac{1}{\mathrm{Norm}_2(\mathbf{N}^H\mathbf{B}(\rho))}$
end for
Search the PMF for a peak and find the corresponding ρ

3) Estimating ρ Using MUSIC: Here we use the same concept for estimating ρ as in estimating θ. The only difference is that we use MUSIC to leverage the phase difference between subsequent subcarriers. As the ray travels, the phases of the subcarriers keep changing each with rate according to their frequencies. If the subcarriers have a spacing of Δf and we have N_s subcarriers, their phase relation with distance is given as

$$\mathbf{B}(\rho) = (1, e^{-j2\pi\rho\Delta f/c}, e^{-j2\pi\rho 2\Delta f/c}, \ldots, e^{-j2\pi\rho(N_s-1)\Delta f/c})^T \quad (2)$$

where ρ is the distance and c is the speed of light. The vector $\mathbf{B}(\rho)$ will be used exactly as we used the steering vector $\mathbf{A}(\theta)$ in estimating θ. The procedure is explained in Algorithm 1. We call the combination of using MUSIC to estimate θ and using MUSIC to estimate ρ the MUSIC/MUSIC (MM) algorithm. Note that in Algorithm 1, PMF(ρ) is a PMF within a scale of constant. In Algorithm 1, the increments for ρ and their final value can be altered.

III. Simulation Environment

In this paper we reused and integrated our algorithms into the simulation environment in [1] so that we can compare the performance improvement in a fair fashion. We adopted the simulation parameters in [4, Table 1] at SNR = 0 dB. We used a three-dimensional environment exactly as in paper [1, Fig. 1(a)], where the antenna is 8.5 meters above the plane of the UEs. We call this three-dimensional scenario as 3D. The simulation environment is 1000m $\times$ 500m. As in [1], the 2048 UEs are placed randomly except 234 of the UEs are selected to make the word "VIP," so we can see if in the channel chart we preserve the shape.

IV. Performance Comparison and Complexity Analysis

We now compare the performance of our algorithms against the results under the three channels used in [1].

A. LR and ISQ Performance Comparison

It can be observed from Table 2, Fig. 2, and Fig. 3 that our algorithms LR and ISQ outperform PCA, SM, and AE from [1] in terms of TW and CT, as well as channel charts. The performance of the training-based LR and the model-based ISQ are very close. We refer the reader to [4] for a discussion of these results.

B. MM Performance Comparison

We will now discuss the performance of the MM algorithm at 2, 8, 20, and 32 subcarriers. Table 3 presents the TW and CT results for k-nearest neighbors equal to 102. Then, Fig. 4 presents the channel charts. Finally, Fig. 5, presents TW and CT performance against k-nearest neighbors. First, we note that, in all cases, increasing the number of subcarriers increases TW and CT performance as well as the visual quality of the channel charts, which is to be expected as more information is available with more subcarriers. One can also carry out a comparison for the same column in Table 3, for all columns, to note that the performance of the MM algorithm with the same number of subcarriers is always the best for the LOS channel, second best for the QLOS channel, and the worst for the QNLOS channel. Note that comparisons with algorithms PCA, SM, AE, LR, and AE are not present in Table 3, Fig. 4, and Fig. 5. For that purpose, one needs to compare Table 3, Fig. 4, and Fig. 5 one-by-one with Table 2, Fig. 2 and Fig 3, respectively.

We will not provide a very detailed comparison. But, it can be seen from the tables and figures above that the MM algorithm provides significantly better TW and CT results, as well as visually more appealing channel charts than those were produced by the PCA, SM, AE, LR, and ISQ algorithms on LOS, QLOS, and QNLOS channels.

C. Runtime and Complexity Comparison

When we compare the complexity of our algorithms ISQ and MM against the three algorithms used in [1], i.e., PCA, SM, and AE, the most important advantage is that our algorithm does not require training or an abundant number of CSI to be able to reduce dimensionality efficiently. We can calculate the channel chart even for one UE data. This can make us calculate the channel chart sequentially in real-time as the data are received. The alternative in [1] is to store the data of all UEs (2048 in our simulations as well as in [1]) and use it all at once as in the case of PCA, SM, or AE, which consumes a very large amount of memory and complexity. Please note that our algorithm LR requires a modest amount of training.

The other advantage is the latency. PCA, SM, and AE algorithms need to collect the data of all UEs, which can take some time. Furthermore, if the system is mobile, the geometry might have already changed by the time the channel chart is calculated. In our case, we can calculate each UE channel chart as we receive it, which makes our algorithms much more efficient.

As an indication of the complexity, we will compare the runtime (simulation time) for producing channel chart for 2048 UEs using different algorithms. This is provided in Table 1. The simulation time is not dependent on the channel, the scale, or the geometry of the environment. The simulation time only

Algorithm	Simulation time (seconds)
PCA	0.817
SM	12.2
AE	53.9
LR	7.15
ISQ	7.09
MM	20.4

Table 1: Simulation times.

relies on the number of UEs, the number of antennas, and the number of subcarriers. In Table 1, the number of UEs is 2048, the number of BS antennas is 32, and the number of subcarriers is 32.

We note that PCA has very small simulation time, much lower than all of the other five algorithms. However, we know from earlier sections that LR, ISQ, and especially MM beats it in terms of performance. SM and AE not only are beaten by LR and ISQ in terms of performance, but also, in terms of simulation time. MM has the best performance by far but its simulation time is about 2.5 times those of LR and ISQ and about 1.5 times that of SM. It has less simulation time than AE. Although its simulation time is at a disadvantage as compared to PCA, SM, LR, and ISQ, the performance gains with it are substantial. We note that the simulation time for PCA is very small but this is due to the fact that PCA is an optimized routine in Matlab. An Internet search would show that PCA is actually considered to be computationally complicated.

The complexity of the MUSIC algorithm is given as $O(M^2P + M^2N)$ where M is the number of antennas, N is the number of snapshots or multiple measurement vectors, and P is the number of potential AOAs searched [6]. In the case of ISQ, we have $M = N_R$, $N = 1$, and P is the number of potential AOAs for the MUSIC algorithm to determine θ, with $P \gg 1$. As a result, we have $O(N_R^2 P)$ to determine θ. With ISQ, to determine ρ, we need N_R instantiations of two squares, a sum, and a square root. These need to be summed, a square root and a reciprocal operation needs to take place. As a result, there are $2N_R$ multiplies, $N_R + 1$ square roots, $2N_R - 1$ adds, and a reciprocal needed to determine ρ. We note that it is stated that a square root operation can be performed with the same complexity as a multiplication [7]. In any case, the complexity of the ISQ algorithm will be dominated by the MUSIC algorithm for θ, which has $O(N_R^2 P)$, where we ignore a second term $N_R^2 \cdot 1$ since $P \gg 1$. The LR algorithm employs MUSIC to determine θ, thus it has the same complexity as ISQ for θ. After LR parameters a and b are determined, it needs $2N_R + 1$ multiplies, N_R square roots, N_R adds, and a $\log$ operation, similar to ISQ. In addition, there is the computational complexity of calculating the linear regression, based on real ρ values of a number of UEs (in our case 256), shared by all UEs (in our case, 2048). As a result, LR is more complicated than ISQ, but with only a small gain over it. We consider its complexity to be dominated by the MUSIC algorithm for estimating θ, or $O(N_R^2 P)$, as in the ISQ algorithm. For the MM algorithm, we use MUSIC to estimate

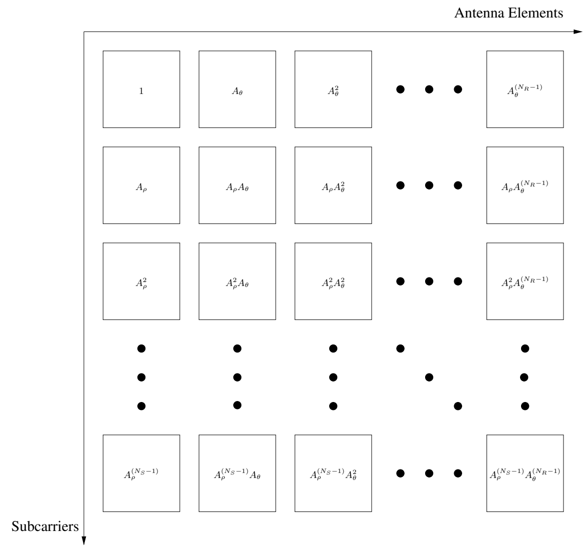

Fig. 1: Ideal CSI matrix $(N_S \times N_R)$ in the absence of fading and noise.

both θ and ρ. For θ, the number of snapshots is equal to N_S, unlike 1 in the case of ISQ and LR. Therefore, the complexity of MUSIC for θ in the MM algorithm is $O(N_R^2 P + N_R^2 N_S)$. For ρ, the roles of N_R and N_S are reversed and P is replaced by Q where Q is the number of potential distances for the MUSIC algorithm to determine ρ. Thus the complexity of the MUSIC algorithm for ρ is $O(N_S^2 Q + N_S^2 N_R)$.

As a result, we can state that the ISQ and LR algorithms are dominated by the MUSIC algorithm for θ, at $O(N_R^2 P)$. On the other hand, MM employs two MUSIC algorithms, for θ and ρ, and has complexity $O(N_R^2 P + N_R^2 N_S + N_S^2 Q + N_S^2 N_R)$.

D. RS Algorithm

In order to reduce complexity substantially, we propose a new model-based algorithm we call rotate and sum (RS). As shown in Fig. 1, the CSI matrix has N_R column vectors where N_R corresponds to the number of receive antennas at the BS, and N_S row vectors where N_S corresponds to the number of subcarriers. In the absence of fading and noise, for each column, we have a rotation factor A_θ. This factor is the phase shift between two vertical neighboring elements or antenna elements. It is equal to $A_\theta = e^{j\pi\cos(\theta)}$, see (1). For each row, we have a rotation factor A_ρ. This factor is the phase shift between two horizontal neighboring elements or subcarriers. It is equal to $A_\rho = e^{-j2\pi\rho\Delta f/c}$ where Δf is the difference in frequency between the subcarriers and c is the speed of light, see (2).

We will discuss two rotate-and-sum procedures in Algorithm 2 and Algorithm 3, corresponding to the estimation of θ and ρ respectively. The algorithms can be understood with the help of Fig. 1. In the case of Algorithm 2, consider the ideal CSI matrix shown in Fig. 1. Recall that this matrix does not include the effects of fading and noise. Let $\mathbf{c}$ be any row

Algorithm 2 Rotate-and-Sum Procedure for Estimating θ

Calculate the CSI matrix across antennas and subcarriers with size $N_S \times N_R$
Set all entries of $N_R \times N_R$ matrix $\mathbf{S}$ equal to 0
for $\texttt{subcarrier\_num} = 1 : N_S$ **do**
 Let $\mathbf{c}$ be the row vector with index $\texttt{subcarrier\_num}$ of the CSI matrix
 Calculate $\mathbf{S} = \mathbf{S} + \mathbf{c}^H \mathbf{c}$
end for
Divide all entries of $\mathbf{S}$ by N_S
for $\phi = 0 : 180$ in increments of 1 **do**
 Let $A_\phi = e^{j\pi \cos\phi}$ where ϕ is in degrees
 Form matrix $\mathbf{B}$ as follows
 for do $i = 1 : N_R$ in increments of 1
  for do $j = i : N_R$ in increments of 1
   $\mathbf{B}_{i,j} = A_\phi^{j-i}$
  end for
  for do $j = 1 : i-1$ in increments of 1
   $\mathbf{B}_{i,j} = [A_\phi^*]^{i-j}$
  end for
 end for
 Let $\mathbf{C}(\phi) = \sum_{i=1}^{N_R} \mathrm{row}_i^H(\mathbf{S})\mathrm{row}_i(\mathbf{B})$
end for
Set $\theta = \arg\max_\phi \mathbf{C}(\phi)$

Algorithm 3 Rotate-and-Sum Procedure for Estimating ρ

Calculate the CSI matrix across antennas and subcarriers with size $N_S \times N_R$
Set all entries of $N_S \times N_S$ matrix $\mathbf{S}$ equal to 0
for $\texttt{antenna\_num} = 1 : N_R$ **do**
 Let $\mathbf{c}$ be the column vector with index $\texttt{antenna\_num}$ of the CSI matrix
 Calculate $\mathbf{S} = \mathbf{S} + \mathbf{c}\mathbf{c}^H$
end for
Divide all entries of $\mathbf{S}$ by N_R
for $\sigma = 0 : 1000$ in increments of 1 **do**
 Let $A_\sigma = e^{-j2\pi\sigma\Delta f/c}$ where σ, $\Delta f/c$ are in m.
 Form matrix $\mathbf{B}$ as follows
 for do $i = 1 : N_S$ in increments of 1
  for do $j = i : N_S$ in increments of 1
   $\mathbf{B}_{i,j} = [A_\sigma^*]^{j-i}$
  end for
  for do $j = 1 : i-1$ in increments of 1
   $\mathbf{B}_{i,j} = [A_\sigma]^{i-j}$
  end for
 end for
 Let $\mathbf{C}(\sigma) = \sum_{i=1}^{N_S} \mathrm{row}_i^H(\mathbf{S})\mathrm{row}_i(\mathbf{B})$
end for
Set $\rho = \arg\max_\sigma \mathbf{C}(\sigma)$

vector of this matrix and consider

$$\mathbf{c}\mathbf{c}^H = \begin{bmatrix} 1 & A_\theta & A_\theta^2 & \cdots & A_\theta^{N_R-1} \\ A_\theta^* & 1 & A_\theta & \cdots & A_\theta^{N_R-2} \\ A_\theta^{*2} & A_\theta^* & 1 & \ddots & A_\theta^{N_R-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_\theta^{*N_R-1} & A_\theta^{*N_R-2} & A_\theta^{*N_R-3} & \cdots & 1 \end{bmatrix} \tag{3}$$

which is independent of the row index. In the case of a real CSI matrix, the elements will deviate from the values of this matrix due to fading and noise. We form an average of $\mathbf{c}\mathbf{c}^H$ for all row vectors $\mathbf{c}$ in the real CSI matrix expecting that the effects of fading and noise will be reduced. We call this average $\mathbf{c}\mathbf{c}^H$ matrix $\mathbf{S}$. Due to this averaging, we expect $\mathbf{S}$ to be close to the ideal $\mathbf{c}\mathbf{c}^H$ matrix in (3) even when fading and noise are present.

Our next task is to figure out what θ is. To that effect, we generate matrices in the same form as (3) but θ is replaced by ϕ. We generate a number of such matrices for values of ϕ in $[0, \pi]$ radians. Suppressing ϕ, each such matrix is called $\mathbf{B}$ in Algorithm 2. For each ρ we compare $\mathbf{B}$ and $\mathbf{S}$, assigning a number representing how close they are. That value of ϕ for which the $\mathbf{B}$ matrix associated with it is closest to $\mathbf{B}$ is deemed to be the best estimate of θ.

Algorithm 3 is based on the same technique, but using column vectors of the CSI matrix, instead of row vectors. It then provides an estimate of ρ. We would like to note that, in both algorithms, the numbers 180 and 1000 are just examples. Instead of uniformly spacing the search numbers ϕ and σ, nonuniform search techniques can be developed. Also, the closeness of the matrices $\mathbf{S}$ and $\mathbf{B}$ can be measured by other techniques, such as a norm of the difference of $\mathbf{S}$ and $\mathbf{B}$.

We have carried out extensive simulations on the performance of the RS algorithm as compared to the MM algorithm. We will state that the performance of the two algorithms is very close, almost the same. In order not to be very repetitive, we only provide Table 4 for comparison with Table 3. Clearly, the results are very close. On the other hand, there is a very important advantage of the RS algorithm against the MM algorithm in terms of complexity. The MM algorithm employs the MUSIC algorithm which is based on an eigenvector and eigenvalue decomposition of an autocorrelation matrix. The computational complexity of this operation is very high. The RS algorithm avoids this operation. If implemented in a general purpose processor, the many multiplications in the RS algorithm would result in high computational complexity. But, it can be implemented in register transfer logic (RTL) which results in a highly simplified implementation.

V. Conclusion

The LR, ISQ, and MM algorithms presented in this paper significantly outperform the three algorithms in [1], PCA, SM, and AE, in terms of performance. As in [1], we measure the performance in terms of connectivity (CT) and trustworthiness (TW). An important advantage of ISQ and MM over the three algorithms from [1] is that we can calculate each UE data independently as it comes, so it is much faster and simpler. In the case of LR, a similar advantage exists, however a number of UE data is first needed in order to perform the regression. The MM algorithm has more complexity than LR and ISQ but the advantage it provides in terms of TW and CT measures is significantly better than those of ISQ and LR. In addition, the MM algorithm results in channel charts with significantly better visual outcome. Finally, we introduced the RS algorithm whose performance is about the same as the MM algorithm. The advantage of this algorithm is that it does not need the eigenvector and eigenvalue decomposition needed by the MM algorithm, and furthermore, it has an RTL implementation which reduces complexity substantially.

We assumed a 3D environment, the same as [1]. We also used static channels and, in the case of ISQ and LR, single subcarrier CSI as in [1].

Note that the MUSIC algorithm for θ is model based. In addition, the ISQ algorithm for ρ is also model based. Finally, the MM and RS algorithms are completely model based for both θ and ρ. Their performance is better than the PCA and SM [1], and the training-based algorithms AE and LR. As a result, we can say that the model-based algorithms ISQ, MM, and RS outperform the algorithms PCA, SM, and AE of [1].

REFERENCES

[1] C. Studer, S. Medjkouh, E. Gonultas, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47 682–47 698, 2018.

[2] S. Jaeckel, L. Raschkowski, K. Borner, and L. Thiele, "Quadriga: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242–3256, 2014.

[3] "Quadriga: The next generation radio channel model," https://quadriga-channel-model.de/.

[4] A. Aly and E. Ayanoglu, "Estimation of cellular wireless user coordinates via channel charting and MUSIC," in *Proc. International Conference on Computing, Networking and Communications (ICNC 2023)*, 2023, pp. 1–5.

[5] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[6] C. Stoeckle, J. Munir, A. Mezghani, and J. A. Nossek, "DoA estimation performance and computational complexity of subspace- and compressed sensing-based methods," in *WSA 2015; 19th International ITG Workshop on Smart Antennas*, 2015, pp. 1–6.

[7] H. Alt, "Square rooting is as difficult as multiplication," *Computing*, vol. 21, pp. 221–232, 1979.

Measure	Channel	PCA	SM	AE	LR	ISQ
TW	LOS	0.8603	0.8272	0.8286	0.9930	0.9885
	QLOS	0.8474	0.8512	0.8574	0.9089	0.9092
	QNLOS	0.8502	0.8456	0.8496	0.9029	0.9041
CT	LOS	0.9288	0.9051	0.8932	0.9968	0.9940
	QLOS	0.9223	0.9278	0.9055	0.9416	0.9304
	QNLOS	0.9237	0.9217	0.9057	0.9246	0.9220

Table 2: Performance comparison for TW and CT at k-nearest = 102 for PCA, SM, AE, LR, and ISQ algorithms in 3D channels.

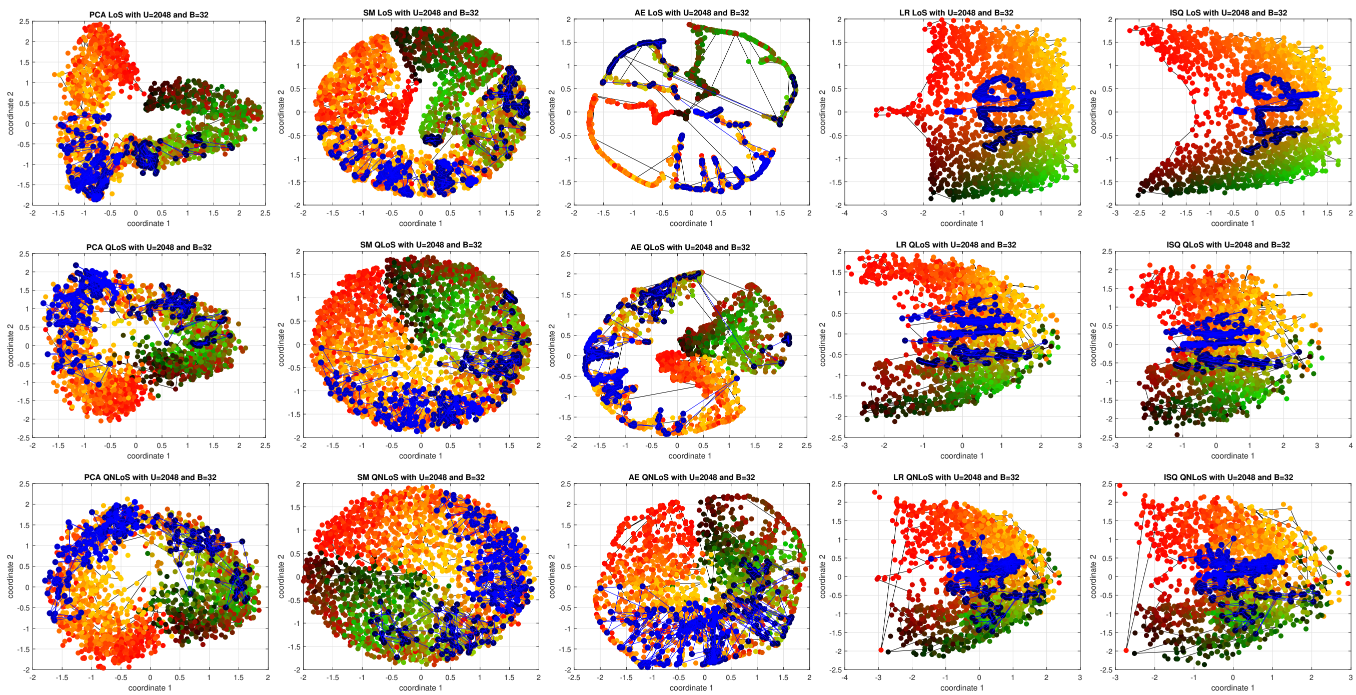

Fig. 2: Channel charts with PCA, SM, AE, LR, and ISQ algorithms for the 3D LOS, QLOS, and QNLOS channels.

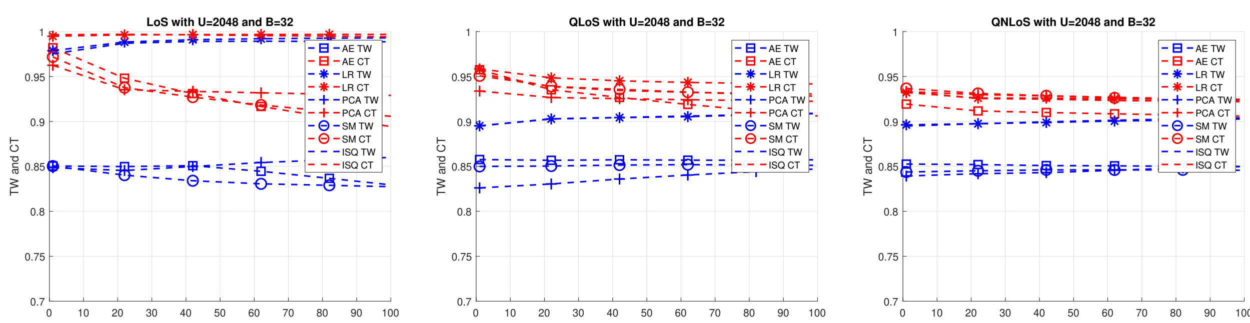

Fig. 3: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 3D. Left: LOS, middle: QLOS, right: QNLOS.

Measure	Channel	2sc	8sc	20sc	32sc
TW	LOS	0.9975	0.9986	0.9997	0.9998
	QLOS	0.9817	0.9958	0.9973	0.9976
	QNLOS	0.9622	0.9802	0.9825	0.9856
CT	LOS	0.9992	0.9999	0.9999	1.0000
	QLOS	0.9816	0.9972	0.9990	0.9992
	QNLOS	0.9626	0.9803	0.9833	0.9865

Table 3: Performance comparison for TW and CT at k-nearest = 102 for MM algorithm in 3D channel at 2, 8, 20, and 32 subcarriers.

Measure	Channel	2sc	8sc	20sc	32sc
TW	LOS	0.9939	0.9954	0.9962	0.9966
	QLOS	0.9751	0.9901	0.9932	0.9941
	QNLOS	0.9551	0.9732	0.9797	0.9827
CT	LOS	0.9934	0.9942	0.9942	0.9943
	QLOS	0.9763	0.9906	0.9928	0.9934
	QNLOS	0.9573	0.9750	0.9805	0.9832

Table 4: Performance comparison for TW and CT at k-nearest = 102 for RS algorithm in 3D channel at 2, 8, 20, and 32 subcarriers.

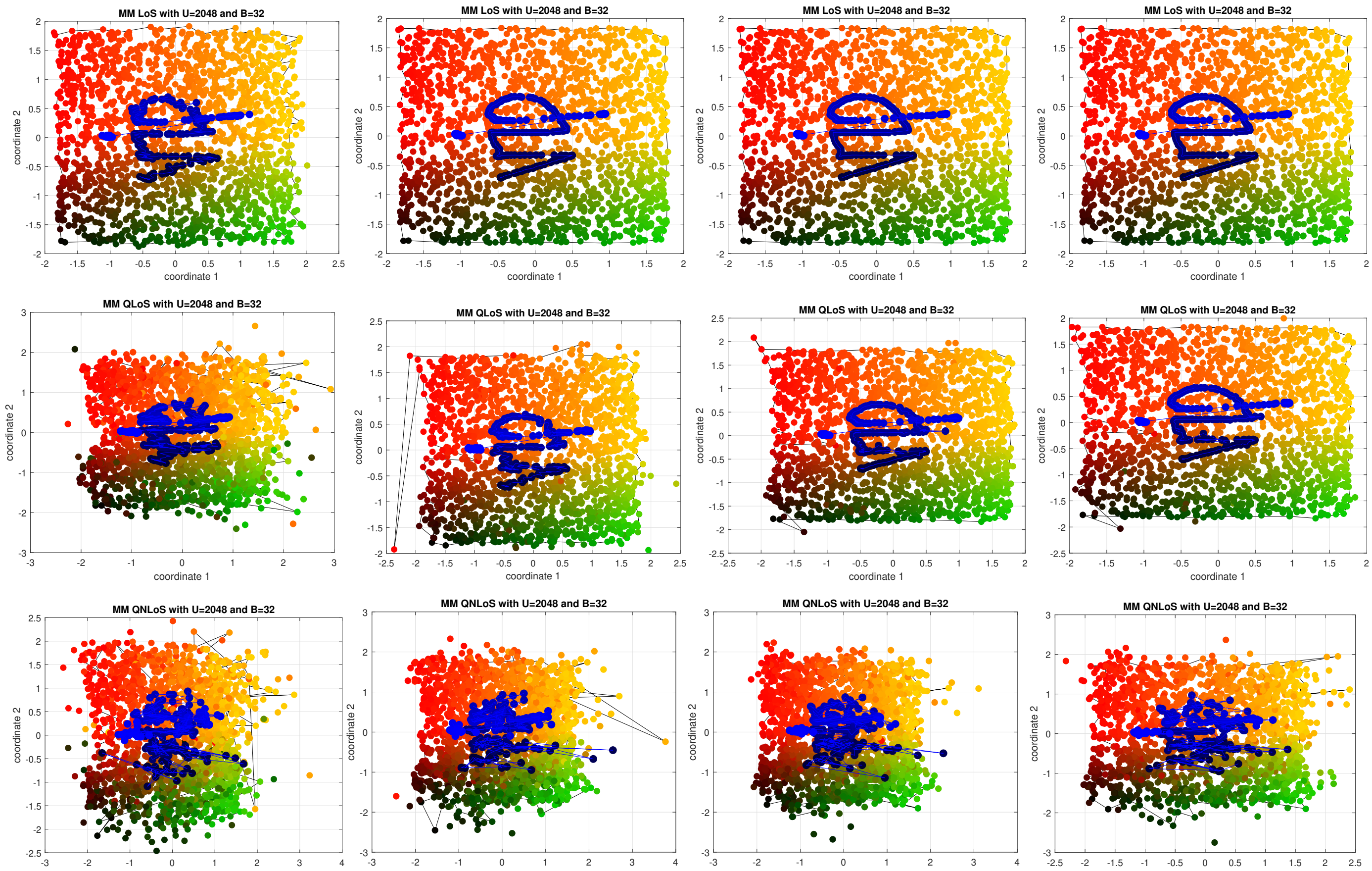

Fig. 4: Channel charts with the MM algorithm for the 3D LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers.

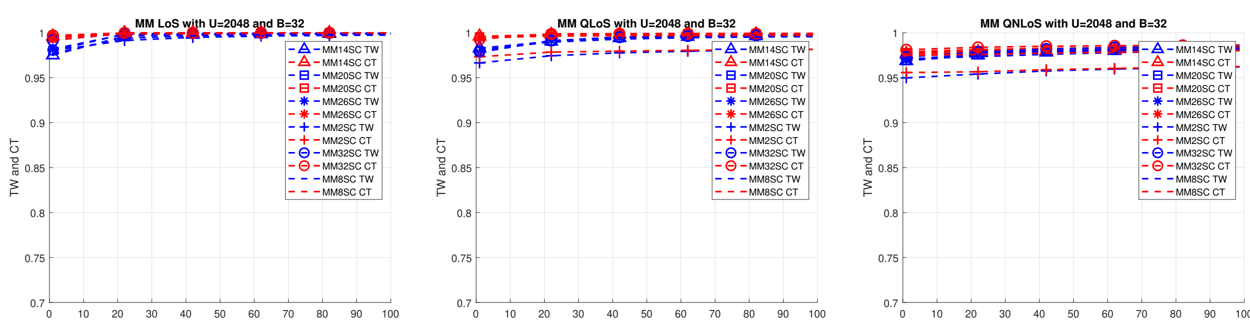

Fig. 5: TW and CT performance against k-nearest neighbors for MM algorithm in 3D channel. Left: LOS, middle: QLOS, right: QNLOS.