

Model-Based Approaches to Channel Charting

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 [1] employing model-based approaches. We estimate the angle of arrival θ and the distance ρ between the base station and the user equipment by employing our algorithms, inverse of the root sum squares of channel coefficients (ISQ) algorithm, linear regression (LR) algorithm, and MUSIC/MUSIC (MM) algorithm. We compare these methods with the channel charting algorithms principal component analysis (PCA), Sammon's method (SM), and autoencoder (AE) [1]. We show that ISQ, LR, and MM surpass PCA, SM, and AE in performance. We also compare our algorithm MM with an algorithm from the literature that uses the MUSIC algorithm jointly on θ and ρ . We call this algorithm the JM algorithm. JM performs very slightly better than MM but at a substantial increase in complexity. Finally, we introduce the rotate-and-sum (RS) algorithm which has about the same performance as the MM and JM algorithms. Unlike MUSIC, RS does not employ eigenvalue and eigenvector analysis. Thus, it is more suitable for direct register transfer logic (RTL) implementation.

Index Terms

Channel charting, user equipment (UE), channel state information (CSI), multiple signal classification (MUSIC), principal component analysis (PCA), Sammon's method (SM), autoencoder (AE).

This work is partially supported by NSF grant 2030029.

This work was partially presented at the International Conference on Computing, Networking, and Communications (ICNC) 2023 and the IEEE International Conference on Communications (ICC) 2023.

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. Previous papers have proposed estimation of a channel chart using a number of dimensionality reduction techniques whereas in this paper we calculate the channel chart directly, employing model-based approaches.

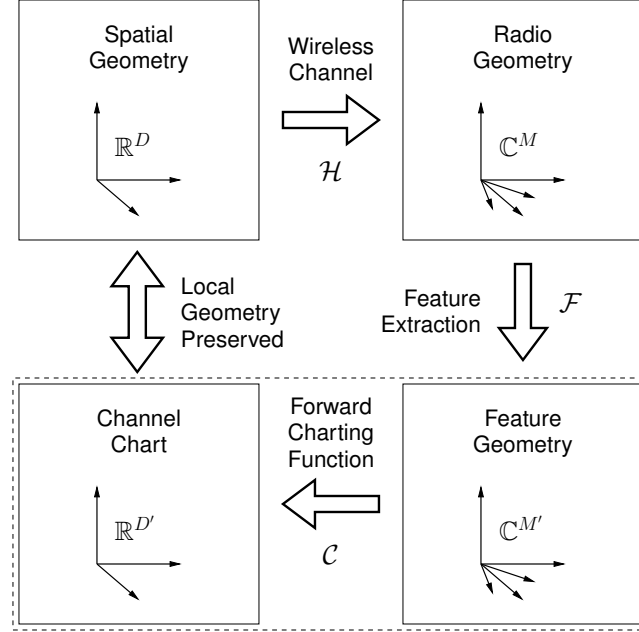


Fig. 1: Summary of channel charting via dimensionality reduction [1].

Channel charting was introduced in [1]. Consider Fig. 1, which is a redrawn and simplified version of Fig. 3 in [1]. As discussed in [1], UE transmitters are located in spatial geometry \mathbb{R}^D , where $D = 2$ or 3. The BS receiver extracts CSI in radio geometry \mathbb{C}^M where $M \gg D$. In the next two steps, a channel chart is generated in $\mathbb{R}^{D'}$ where $D' \leq D$ such that the representation in $\mathbb{R}^{D'}$ preserves the relative positions of the UEs. Reference [1] discusses and compares three dimensionality reduction algorithms from the machine learning literature, namely principal component analysis (PCA), Sammon's mapping (SM), and autoencoder (AE).¹ PCA is a linear method for dimensionality reduction. It maps a high-dimensional point set (e.g., CSI features) into a low-dimensional point set (e.g., the channel chart) in an unsupervised manner [1]. Although it is sometimes referred to as an unsupervised machine learning algorithm, it performs dimensionality reduction only for the data points used in calculations and does not form a function one can use to perform dimensionality reduction for future data points. SM is a nonlinear method for dimensionality reduction with the goal of retaining small pairwise distances between the two

¹Please see Sec. VI about the SM+ algorithm from [1].

point sets [1]. Similarly to PCA, it does not form a function for dimensionality reduction of future data points. Whereas, an AE is a deep artificial neural network used for unsupervised dimensionality reduction [1]. Unlike PCA and SM, it performs learning and can be used for future data points.

In Fig. 1, four blocks to carry out channel charting are presented. In the upper left, the spatial geometry in \mathbb{R}^D is shown. The upper right block creates the radio geometry in \mathbb{C}^M . The lower blocks perform feature extraction and forward charting to create channel charts. The approach proposed in this paper keeps the upper two blocks. It replaces the lower two blocks with model-based techniques to determine the angle of arrival and the distance from the BS of the UE directly. As a result, preservation of the relative positions of the UEs with respect to the BS, a goal of channel charting, is automatically satisfied.

Currently, channel charting is an active research area, see, e.g., [2]–[23]. Potential ways to improve channel charting are discussed in [3]–[5], [15], [18]. The use of channel charting for different applications is discussed in [7], [10], [12], [13], [17], [22], [23]. References [2], [6], [9], [14], [20] are on multipoint channel charting. References [8], [11], [21] discuss triplet-based channel charting, while [16], [19] concentrate on triplet loss. Reference [17] tries to use the AE algorithm in a supervised fashion by allowing some of the UEs to have global positioning system (GPS) data for the exact location and use it to improve the AE learning of the geometry.

A. Contributions of the Paper

We estimate the angle of arrival (AOA) θ and the distance ρ of the UE from the BS, based on the phase and magnitude of the CSI. Towards that end, we propose four techniques.

- We estimate the AOA θ using the MUSIC algorithm based on the correlation matrix of the channel coefficients [24]. For the distance ρ between the UE and the BS, we sum the magnitude of the channel gain on all antennas for each UE, then we take the inverse of the root of the sums and use it as ρ . We call this algorithm ISQ.
- We estimate the AOA θ using the MUSIC algorithm. For ρ , we consider linear regression of the known locations of a subset of UEs with the logarithm of the sum of the absolute value of the channel gain to estimate a slope and intercept. We call this algorithm LR.
- Our third algorithm requires multicarrier transmission from the UE to the BS. For θ , we use MUSIC. For ρ , we also use MUSIC with multiple subcarriers to estimate the distance from the phase difference. We call this algorithm MM.
- We model the ideal CSI matrix in the absence of noise and fading. This model happens to have entries proportional to powers of $A_\theta = e^{j\pi \cos(\theta)}$ and $A_\rho = e^{-j2\pi\rho\Delta f/c}$ where Δf is the difference in frequency between the subcarriers and c is the speed of light. We then use a novel procedure

to determine the best fits to θ and ρ in the actual CSI matrix among all possible θ and ρ . We call this algorithm RS. RS does not employ eigenvalue and eigenvector analysis and may be simpler to implement in register transfer logic (RTL).

We then compare the results of these approaches and those of PCA, SM, and AE. We also compare their performance among themselves. This performance is measured by two quantities CT and TW introduced in Sec. V-A as well as by using channel charts in terms of visual representation. We provide a comparison of the complexity and simulation times of PCA, SM, AE, ISQ, LR, and MM. Our algorithms perform better than the conventional algorithms in [1]. Note that in ISQ, LR, and MM, θ is calculated by employing the MUSIC algorithm. In each of these three cases, the novelty is in the calculation of ρ . This is the first use of the MUSIC algorithm in channel charting and this is the first paper that introduces model-based approaches to channel charting. In addition,

- We compare the performance and complexity of an algorithm from the literature [25] we call JM with that of MM. JM performs about the same as MM, but has significantly more complexity.

We note that in our algorithms ISQ, LR, and MM, the main contribution is new methods for estimating the range (distance ρ of the UE to the BS). In these algorithms, the AOA θ is calculated by using the conventional MUSIC algorithm. In our algorithms ISQ and LR, in addition to using the MUSIC algorithm for θ , heuristic algorithms are used to estimate ρ . In our algorithm MM, in addition to using the MUSIC algorithm for θ , the MUSIC algorithm is used for ρ while employing multicarrier transmission. On the other hand, our algorithm RS does not employ the MUSIC algorithm either to estimate θ or ρ , instead it uses properties of the CSI matrix to estimate these quantities. Unlike the original paper on channel charting [1], we do not explicitly attempt to preserve the local geometry of UEs. This property occurs as a consequence of estimating the UE positions accurately. As a result, the relative positions of UEs are mapped to the channel chart independently. This has many advantages such as reduced computational complexity, lower latency, incorporation of individual UE positions without waiting for the calculation of the channel chart for the whole ensemble, and easy accommodation of new services.

All of the four techniques ISQ, LR, MM, and RS are new. In ISQ, LR, and MM, the conventional MUSIC algorithm is employed to estimate θ , but novel algorithms are introduced to calculate ρ . In algorithm RS, both θ and ρ are calculated by using novel methods.

II. CHANNEL MODELS

Throughout this paper, we employ three channel models, namely vanilla line-of-sight (LOS), Quadriga LOS (QLOS), and Quadriga non-LOS (QNLOS) [26]–[28]. These are the same models used in the paper

Parameter	Value
Antenna array	Uniform Linear Array (ULA) with spacing $\lambda/2 = 7.495$ cm
Number of array antennas	32
Number of transmitters (UEs)	2048
Carrier frequency	2.0 GHz
Bandwidth	312.5 kHz
Number of clusters	0
Number of subcarriers	1 (up to 32 in the case of the MM algorithm (Sec. III-B3))

Table 1: Simulation parameters.

that introduced the technique of channel charting [1] so that we can compare our results with that work. We start with the simplest, vanilla LOS. Vanilla LOS is one LOS ray described as

$$h = \rho^{-r} e^{-j(\frac{2\pi\rho}{\lambda} + \phi)} \quad (1)$$

where ρ is the distance between the transmitter and the receiver and r is known as the path loss exponent. In (1), the first term in the channel phase is linearly proportional with the distance ρ . The second term ϕ is a uniformly distributed random variable in $[0, 2\pi)$. The channel amplitude is a random variable (Rician (QLOS) or Rayleigh (QNLOS)) which is inversely proportional to the distance square for free space, $\sim \rho^{-2}$, i.e., the path loss exponent $r = 2$. For more crowded environments, the path loss exponent r can be 3 or 4. We will use $r = 2$ for the vanilla LOS channel in this paper. Note that the model in (1) is an idealized one, we discuss a set of more realistic channel models next.

Next we discuss the Quadriga channel model [26]–[28]. Quadriga stands for quasi deterministic radio channel generator. It is a statistical three-dimensional geometry-based stochastic channel model employing ray tracing. According to [26], it has the following features: *i*) three-dimensional propagation (antenna modeling, geometric polarization, scattering clusters), *ii*) continuous-time evolution, *iii*) spatially correlated large- and small-scale fading, and *iv*) transition between varying propagation scenarios. The Quadriga model is very customizable. It has many features and details. We employ the following set of parameters: *i*) the coordinates of the transmitters and receivers, *ii*) the carrier frequency, the bandwidth, and the number of subcarriers, *iii*) the number of clusters, *iv*) the antenna shape, polarization, number of elements, and spacing between them, and *v*) QLOS or QNLOS scenario. The model was validated by measurements in downtown Dresden, Germany [28, Ch. 4] and in downtown Berlin, Germany [28, Ch. 5], with accuracies better than the 3GPP-3D model [28]. The channel noise is modeled as additive white Gaussian thermal noise and, in verifications of the model, the sensitivity of the measurement system is taken into account [28]. In this paper we used the parameters in Table II with the Urban Macro-Cell (UMa) version of the Quadriga mode in the simulations. Some details of the measurement setup are available in [26, Sec. III],

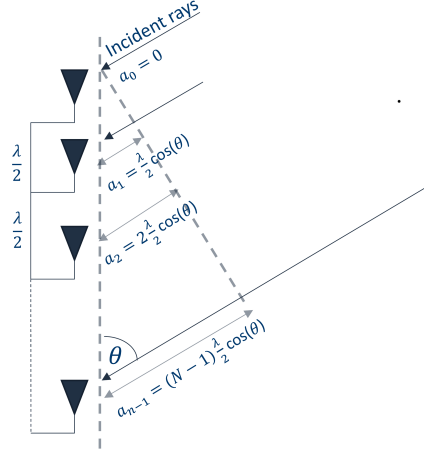


Fig. 2: Angle of arrival (θ) relation with phase.

in specific detail in [26, Table II].

The signal-to-noise ratio (SNR) in channel model is calculated by considering the power in the received signal (P_r) and the power in the noise measured at the receiver (P_n). We note that while the estimated channel would have some noise added to it, the most significant component of the noise at the receiver is additive white Gaussian thermal noise. Then, the SNR at the receiver is given as $\text{SNR} = P_r/P_n$ where P_r takes into account the transmitted power and the channel model, see, e.g., Sec. II-B in [29]. In the code [30] which we used as the basis for our simulations, the calculation of SNR is carried out by normalizing P_r and then properly scaling the additive white Gaussian thermal noise power P_n for all three channel models.

Many channel models have been developed within the last two decades. For example, see [31] for COST 231 and SUI models, [32] for 3GPP, 3GPP2, COST 259, COST 273, COST 2100, WINNER, and WINNER II models, and [33] for Quadriga, NYUSIM, and MG5G models. A discussion of these and similar models is out of scope of this paper. In this paper, we employ the same models used in [1] so as to be able to compare our results with this original paper on channel charting.

III. 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

From Fig. 2, we can see that each antenna element will receive a ray that travels an additional distance $\frac{\lambda}{2} \cos(\theta)$ than the previous element. This means, for each antenna element, the incremental phase shift is

Algorithm 1 MUSIC Procedure for Estimating θ

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 $\theta = 0 : 180$ in increments of 1 **do**
 Calculate the steering vector $\mathbf{A}(\theta)$
 Calculate the PMF $(\theta) = \frac{1}{\text{Norm}_2(\mathbf{N}^H \mathbf{A}(\theta))}$
end for
 Search the PMF for a peak and find the corresponding θ

$e^{j\pi \cos(\theta)}$. With this shift, we get what is called the steering vector

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

where N_R is the number of receive antennas at the BS. This vector is essential in beamforming applications and in determining the AOA.

The steering vector $\mathbf{A}(\theta)$ is embedded within the CSI correlation matrix ($\mathbf{R} = \mathbb{E}[\mathbf{h}\mathbf{h}^H]$), where \mathbf{h} is the received channel vector at the BS. along with noise. The vector \mathbf{h} is $N_R \times 1$. If we decompose \mathbf{R} into its eigenvectors and examine the corresponding eigenvalues, we can separate the eigenvectors into a signal subspace \mathcal{S} and a noise subspace \mathcal{N} , using the fact that the noise eigenvectors will correspond to very small eigenvalues compared to the signal space eigenvalues. The subspaces \mathcal{S} and \mathcal{N} are orthogonal to each other. Assume that the dimensionality of \mathcal{N} is p . Form the $N_R \times p$ matrix \mathbf{N} by concatenating the eigenvectors of \mathcal{N} next to each other. The multiplication of the noise subspace eigenvectors matrix \mathbf{N} and the steering vector will be almost zero. We can use this concept to find the correct angle by sweeping θ in the steering vector as illustrated in Algorithm 1 where PMF stands for probability mass function. Note that $\text{PMF}(\theta)$ is a PMF within a scale of constant.

B. Estimating ρ

We will now discuss how to estimate ρ . The simple channel ray model can be depicted as in (1) with $r = 2$.

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

$$\rho = \frac{1}{\sqrt{\sum_{n=0}^{N_R-1} \text{abs}(h_n)}}, \quad (3)$$

where N_R is the number of antennas at the base station and h_n is the channel between the UE and the n -th antenna at the BS. We refer to this algorithm as ISQ (inverse square root sum). The motivation

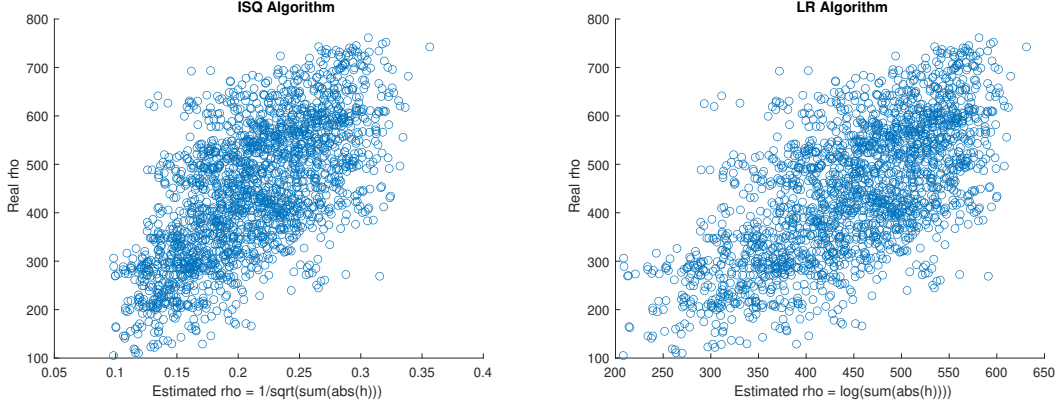


Fig. 3: Correlation of real ρ vs estimated ρ under the channel model QNLOS.

for this algorithm comes from (1) with the path loss component $r = 2$. Based on this formulation, $\rho = 1/\sqrt{\text{abs}(h_n)}$ and (3) is a way of calculating this in an average sense.²

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

$$\rho = aX + b, \quad \text{where } X = \log \sum_{n=0}^{N_R-1} \text{abs}(h_n). \quad (5)$$

For the first 256 UEs, we use the known ρ and X values to generate a and b in a linear regression, and then for the rest of the UES, we use (5) to estimate ρ based on their X values. We call this algorithm the LR algorithm. As we will show later, the unsupervised performance of the ISQ algorithm is almost identical to the LR algorithm. Noting the log operation in (5), and the fact that linear regression will generate $a < 0$, this is a different way of expressing (3).³ We would like to point out to the subtlety that while the LR algorithm employs the model in (5), because of the use of linear regression being based on the first 256 (in our case) UE locations, it may be considered training based. We also note that the regression in (5) is called a linear-log regression. Another approach could be to use log-log regression where ρ in the left hand side of (5) is replaced by $\log \rho$ [34]. Although this is closer to the model in (3), we did not observe a significant difference in numerical simulations considering linear-log and log-log

²Note that

$$\rho' = \frac{1}{\sqrt{\frac{1}{N_R} \sum_{n=0}^{N_R-1} \text{abs}(h_n)}} = \sqrt{N_R} \rho. \quad (4)$$

Thus the true average ρ' is proportional to ρ . In other words, estimated ρ is not to scale with the real ρ , but that will not affect the TW and CT.

³Note that

$$X' = \log \left(\frac{1}{N_R} \sum_{n=0}^{N_R-1} \text{abs}(h_n) \right) = X - \log(N_R). \quad (6)$$

Therefore, the true average X' differs from X by a constant term, which can be absorbed by b in (5).

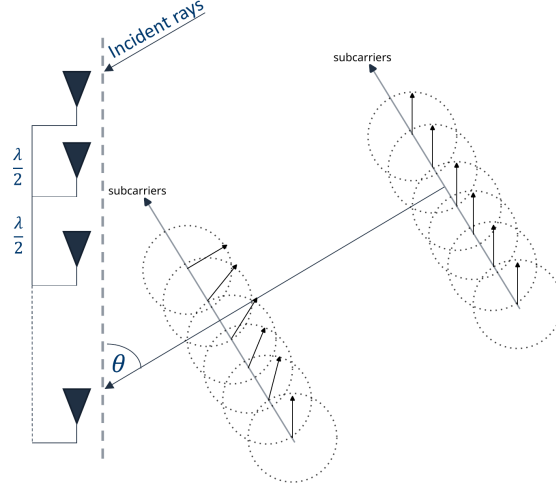


Fig. 4: Phase change across subcarriers with distance.

Algorithm 2 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 $\text{PMF}(\rho) = \frac{1}{\text{Norm}_2(\mathbf{N}^H \mathbf{B}(\rho))}$

end for

Search the PMF for a peak and find the corresponding ρ

regression techniques in terms of CT and TW. In addition, as can be observed from Fig. 3, both ISQ and LR generate estimates that correlate linearly with real 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 one can see from Fig. 4, 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

$$\mathbf{B}(\rho) = (1, e^{-j2\pi\rho\Delta f/c}, e^{-j2\pi\rho2\Delta f/c}, \dots, e^{-j2\pi\rho(N_S-1)\Delta f/c})^T \quad (7)$$

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 2. Note that in Algorithm 2, $\text{PMF}(\rho)$ is a PMF within a scale of constant. We call the combination of using MUSIC to estimate θ and using MUSIC to estimate ρ the MUSIC/MUSIC (MM) algorithm. A flowchart for this algorithm is given in Fig. 5. In this flowchart, N_{NS} is the number of noise subspace eigenvectors, i.e., N_{NS} eigenvectors are

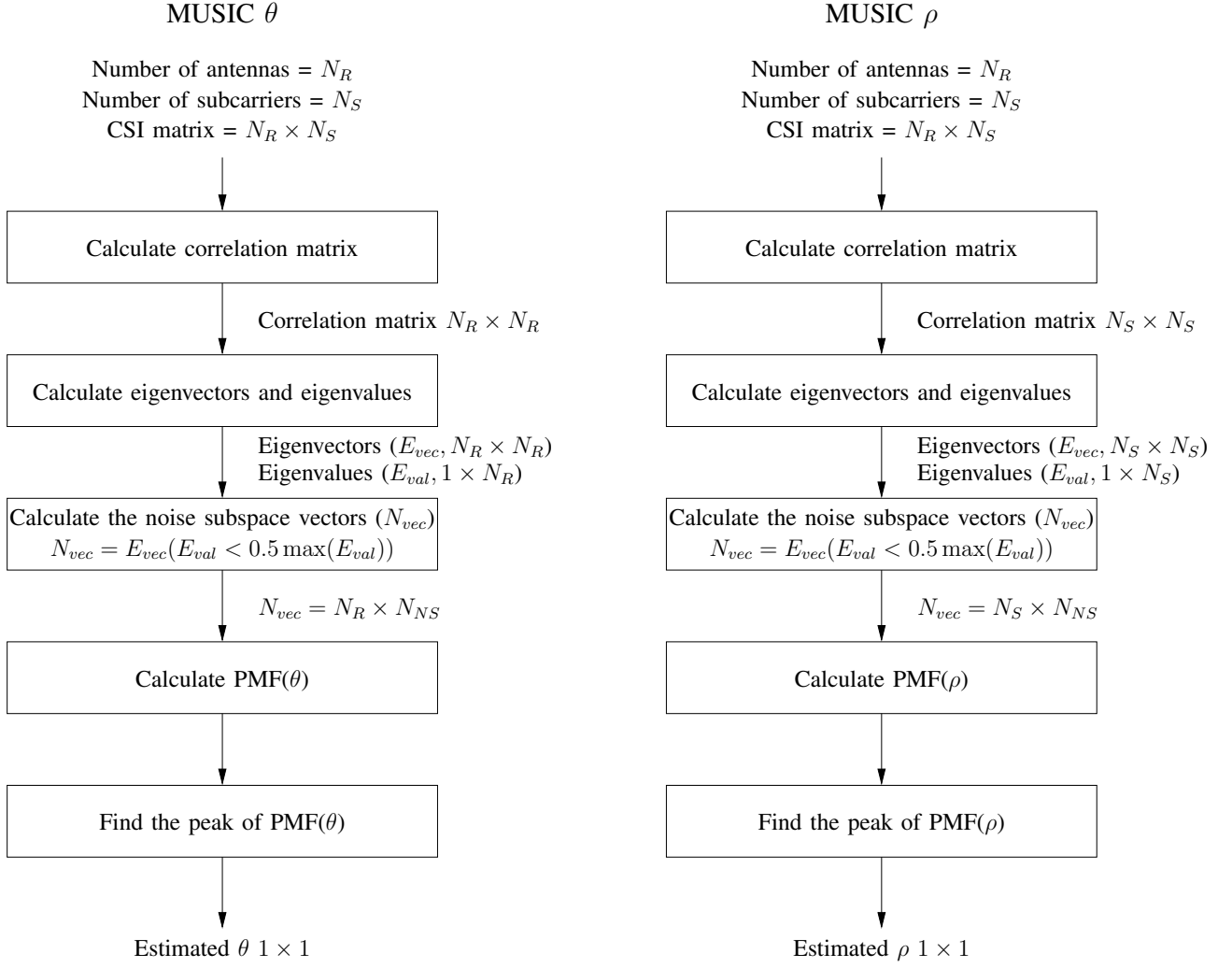


Fig. 5: MUSIC algorithm to find θ and ρ . The combination is called the MM algorithm..

selected for noise subspace.

In Sec. V we will present the performance of the MM algorithm separately than the others, as it requires multiple subcarriers. For this reason, we cannot compare fairly with the other algorithms, but we will show the performance with different numbers of antennas and subcarriers. We came up with this algorithm within the context of channel charting. After calculating our results with it, we became aware that a related algorithm was proposed in a different context [25].

C. JM Algorithm

We will now discuss the algorithm in [25]. This is a MUSIC algorithm that calculates θ and ρ jointly, as opposed to our technique of calculating them separately in our MM algorithm in Sec. III-B3. For this reason, we will call it the joint MUSIC (JM) algorithm in this paper.

The JM algorithm considers N_R receive antennas at the BS, transmitted over N_S subcarriers in a multicarrier transmission system. It forms an $N_S \times N_R$ matrix to use a joint MUSIC algorithm on, in

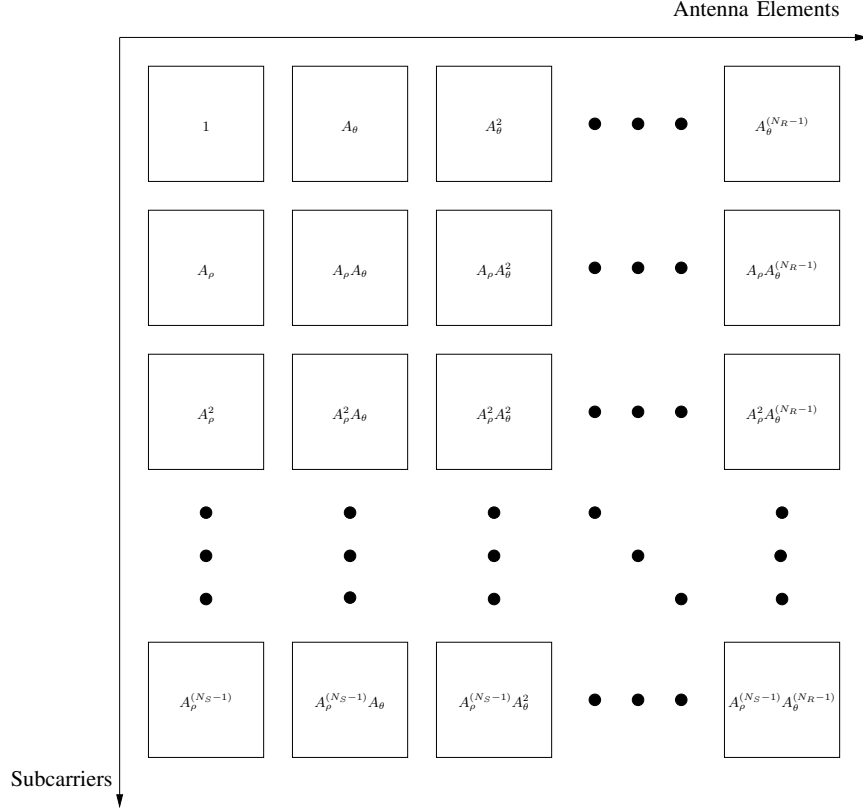


Fig. 6: Ideal CSI matrix ($N_S \times N_R$) in the absence of fading and noise.

order to jointly determine θ and ρ . To that end, it employs a subarray of dimension $N_{Sa} \times M_{Sa}$ to prepare the data in the $N_S \times N_R$ matrix for smoothing and generates an $N_{Sa}M_{Sa} \times (N_S - N_{Sa} + 1)(N_R - M_{Sa} + 1)$ matrix. The purpose of this two-dimensional smoothing is to remove noise at the expense of increasing the complexity of calculating the covariance matrix. After the smoothing operation, the resulting covariance matrix is $N_{Sa}M_{Sa} \times N_{Sa}M_{Sa}$. The MUSIC algorithm is then applied to the covariance matrix. With this algorithm, the search vector \mathbf{A} is a function of both θ and ρ . Then, the eigenvectors and eigenvalues of the covariance matrix are calculated. The resulting PMF is also a function of both θ and ρ . Furthermore, the search is carried out in two dimensions.

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. 6, 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 (2). For each row, we have a rotation factor A_ρ . This factor is the phase shift between two horizontal

Algorithm 3 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 **S** equal to 0
for subcarrier_num = 1 : N_S **do**
 Let **c** be the row vector with index subcarrier_num of the CSI matrix
 Calculate $\mathbf{S} = \mathbf{S} + \mathbf{c}^H \mathbf{c}$
end for
Divide all entries of **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 **B** as follows
 for $i = 1 : N_R$ in increments of 1 **do**
 for $j = i : N_R$ in increments of 1 **do**
 $\mathbf{B}_{i,j} = A_\phi^{j-i}$
 end for
 for $j = 1 : i - 1$ in increments of 1 **do**
 $\mathbf{B}_{i,j} = [A_\phi^*]^{i-j}$
 end for
 end for
 Let $C(\phi) = \sum_{i=1}^{N_R} \text{row}_i(\mathbf{S}) \text{row}_i^H(\mathbf{B})$
end for
Set $\theta = \arg \max_\phi C(\phi)$

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 (7).

We will discuss two rotate-and-sum procedures in Algorithm 3 and Algorithm 4, corresponding to the estimation of θ and ρ respectively. The algorithms can be understood with the help of Fig. 6. In the case of Algorithm 3, consider the ideal CSI matrix shown in Fig. 6. Recall that this matrix does not include the effects of fading and noise. Let **c** be any row vector of this matrix and consider

$$\mathbf{c}^H \mathbf{c} = \begin{bmatrix} 1 & A_\theta & A_\theta^2 & \dots & A_\theta^{N_R-1} \\ A_\theta^* & 1 & A_\theta & \dots & A_\theta^{N_R-2} \\ A_\theta^{*2} & A_\theta^* & 1 & \ddots & A_\theta^{N_R-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_\theta^{*N_R-1} & A_\theta^{*N_R-2} & A_\theta^{*N_R-3} & \dots & 1 \end{bmatrix} \quad (8)$$

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 $\mathbf{c}^H \mathbf{c}$ for all row vectors **c** in the real CSI matrix expecting that the effects of fading and noise will be reduced. We call this average $\mathbf{c}^H \mathbf{c}$ matrix **S**. Due to this averaging, we expect **S** to be close to the ideal $\mathbf{c}^H \mathbf{c}$ matrix in (8) even when fading and noise are present.

Algorithm 4 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 S equal to 0
for antenna_num = 1 :  $N_R$  do
    Let c be the column vector with index antenna_num of the CSI matrix
    Calculate  $\mathbf{S} = \mathbf{S} + \mathbf{c}\mathbf{c}^H$ 
end for
Divide all entries of S by  $N_R$ 
for  $\sigma = 0 : 1000$  in increments of 1 do
    Let  $A_\sigma = e^{-j2\pi\sigma\Delta f/c}$  where  $\sigma, \Delta f/c$  are in m.
    Form matrix B as follows
    for  $i = 1 : N_S$  in increments of 1 do
        for  $j = i : N_S$  in increments of 1 do
             $\mathbf{B}_{i,j} = [\mathbf{A}_\sigma^*]^{j-i}$ 
        end for
        for  $j = 1 : i - 1$  in increments of 1 do
             $\mathbf{B}_{i,j} = [\mathbf{A}_\sigma]^{i-j}$ 
        end for
    end for
    Let  $C(\sigma) = \sum_{i=1}^{N_S} \text{row}_i(\mathbf{S})\text{row}_i^H(\mathbf{B})$ 
end for
Set  $\rho = \arg \max_{\sigma} C(\sigma)$ 

```

Our next task is to figure out what θ is. To that effect, we generate matrices in the same form as (8) 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 **B** in Algorithm 3. For each ϕ we compare **S** and **B**, assigning a number representing how close they are. That value of ϕ for which the **B** matrix associated with it is closest to **S** is deemed to be the best estimate of θ .

IV. 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 Table II at SNR = 0 dB. SNR is defined as the ratio of the signal power to the noise power and 0 dB means that the two are equal. We used a three-dimensional environment exactly as in paper [1] as shown in Fig. 7, where the antenna is 8.5 meters above the plane of the UEs. We call this three-dimensional scenario as 3D. A similar two-dimensional (2D) scenario is investigated in [35]. 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.

The basis of the simulation program is the Matlab code [30] released by the authors of [1]. In this code, PCA and SM are available in Matlab in pre-compiled and optimized form. We used the AE code kindly

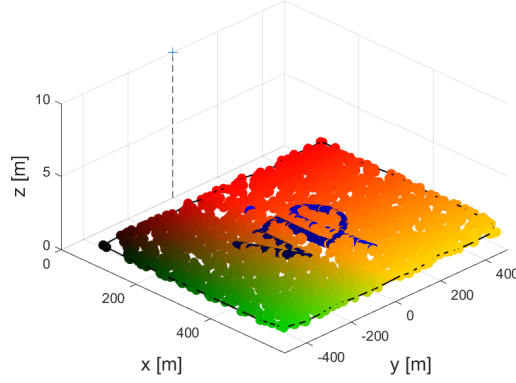


Fig. 7: 3D environment.

supplied to us by the first author of the original paper on channel charting [1]. The AE code was written in Python so we integrated it as an executable Python environment, which we can call from Matlab as a system call. We wrote the LR, ISQ, and MM codes. We integrated the latest Quadriga channel into the model. This is important because the Quadriga channel model employed in [1] has been changed online, with the channel employed in [1] being no longer available [36].

V. PERFORMANCE COMPARISON AND COMPLEXITY ANALYSIS

A. 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. Let $\mathcal{V}_K(\mathbf{u}_i)$ be the K -neighborhood of point \mathbf{u}_i in the original space. Also, let $\hat{r}(i, j)$ be the ranking of point \mathbf{v}_j among the neighbors of point \mathbf{v}_i , ranked according to their similarity to \mathbf{v}_i . Then the point-wise continuity of the representation \mathbf{v}_i of the point \mathbf{u}_i is defined as

$$\text{CT}_i(K) = 1 - \frac{2}{K(2N - 3K - 1)} \sum_{j \in \mathcal{V}_K(\mathbf{u}_i)} (\hat{r}(i, j) - K). \quad (9)$$

The (global) continuity of a point set $\{\mathbf{u}_n\}_{n=1}^N$ and its representation $\{\mathbf{v}_n\}_{n=1}^N$ is

$$\text{CT}(K) = \frac{1}{N} \sum_{i=1}^N \text{CT}_i(K). \quad (10)$$

Now, let $\mathcal{U}_K(\mathbf{v}_i)$ be the set of “false neighbors” that are in the K -neighborhood of \mathbf{v}_i , but not of \mathbf{u}_i in the original space. Also, let $r(i, j)$ be the ranking of point \mathbf{u}_i in the neighborhood of point \mathbf{u}_i , ranked

according to their similarity to \mathbf{u}_i . The point-wise trustworthiness of the representation of point \mathbf{u}_i is then

$$\text{TW}_i(K) = 1 - \frac{2}{K(2N - 3K - 1)} \sum_{j \in \mathcal{U}_K(\mathbf{v}_i)} (r(i, j) - K). \quad (11)$$

The (global) trustworthiness between a point set $\{\mathbf{u}_n\}_{n=1}^N$ and its representation $\{\mathbf{v}_n\}_{n=1}^N$ is

$$\text{TW}(K) = \frac{1}{N} \sum_{i=1}^N \text{TW}_i(K). \quad (12)$$

Both point-wise and global CT and TW are between 0 and 1, with larger values being better [1].

B. LR and ISQ Performance Comparison

We will now compare the performance of our algorithms LR and ISQ to the three algorithms PCA, SM, and AE from [1] in terms of TW and CT. These are given in Table V-B for the 3D channel. The value of k -nearest neighbors is 102 in both cases. We first note that LR and ISQ outperform the techniques in [1], namely PCA, SM, and AE. Comparing our two techniques LR and ISQ, we find that LR performance is very close to that of ISQ. The difference is less than 1%.

In close examination, we note that TW values with our algorithms LR and ISQ are close to 1 for the LOS channel, while the other three algorithms have performance in the range 0.8272–0.8603, with PCA performing better. Yet, LR and ISQ have almost perfect TW performance for this channel. For the QLOS and QNLOS channels, LR and ISQ perform in the range of 0.9029–0.9092 for TW, whereas the other three algorithms have TW performance in the range of 0.8456–0.8574. Thus, LR and ISQ have significantly better TW performance than the other three algorithms for the LOS, QLOS, and QNLOS channels. Moving on to the CT performance, for the LOS channel, LR and ISQ have performance in the range of 0.9940–0.9968, which is significantly better than the other three algorithms whose performance is in the range of 0.8932–0.9288. For the QLOS and QNLOS channels, the performance of LR and ISQ is in the range of 0.9220–0.9416, while the other three algorithms have performance in the range of 0.9055–0.9278.

Fig. 8 presents the channel charts for the 3D channel. In Fig. 8, columns 1 through 5 correspond to PCA, SM, AE, LR, and ISQ algorithms, respectively. Therefore, one should compare the fourth and the fifth columns with the first three columns on a row-by-row basis. The considered system geometry is given in Fig. 7 or [1, Fig. 1(a)]. The goal of the channel chart is to employ CSI and then derive a chart which preserves the distances in the system geometry. It can be seen from Fig. 8 that our algorithms LR and ISQ do a significantly better job than PCA, SM, and AE in that regard. In particular, the letters VIP present in Fig. 7 ([1, Fig. 1(a)]) can be seen to be much more preserved with LR and ISQ. LR and ISQ channel charts are similar with a very slight preference towards LR. In close examination, we first look

at the first row of Fig. 8, corresponding to the LOS channel. Our algorithms LR and ISQ have visually very appealing performance, reproducing the word “VIP” in very close representation. Whereas, the other three algorithms do not have performance that is close. Among the other three algorithms, AE is a clear worst, PCA being better, and perhaps SM the best. Yet, their performance is nowhere close to those of ISQ or LR. The visual appearance of the channel charts for the LR and ISQ algorithms on the QLOS and QNLOS channels deteriorate but the word “VIP” is still somewhat visible. Interestingly, the PCA, SM, and AE algorithms appear to result in better channel charts on the QLOS and QNLOS channels. But, as in the case with the numerical results in Table V-B, LR and ISQ yield clearly better channel charts than PCA, SM, and AE.

TW and CT performance curves for the 3D channel are given in Fig. 9 against k -nearest neighbors. The blue curves are for TW, while the red curves are for CT. ISQ results are given by dashed curves with a circular symbol, while LR curves are given by dashed curves with a plus symbol. It can be observed that while the performance of LR and ISQ are close, they consistently beat PCA, SM, and AE results. In particular, considering the LOS results in Fig. 9, it can be observed that LR and ISQ results in terms of both TW and CT are very close to each other, while PCA, SM, and AE results are much worse. Furthermore, LR and ISQ results improve with k -nearest neighbors in terms of both TW and CT, while PCA, SM, and AE results get worse with increasing k -nearest neighbors in the case of CT, and remain about the same in the case of TW. Considering NLOS and QNLOS cases, in the case of TW, LR and ISQ perform significantly better than the other three algorithms. In the case of CT, LR and ISQ still have better performance than AE, and essentially the same as PCA and SM.

C. MM Performance Comparison

We will now discuss the performance of the MM algorithm. Table V-C presents the TW and CT results for k -nearest neighbors equal to 102 at 2, 8, 14, 20, 26, and 32 subcarriers. Then, Fig. 10 presents the channel charts at 2, 8, 20, and 32 subcarriers. Finally, Fig. 11, presents TW and CT performance against k -nearest neighbors at 2, 8, 14, 20, 26, and 32 subcarriers. First, we note that, in most 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 V-C, 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 V-C, Fig. 10, and Fig. 11. For that purpose, one needs to compare Table V-C, Fig. 10, and Fig. 11 one-by-one with Table V-B, Fig. 8, and Fig 9, respectively.

Let us first consider TW and CT results for the MM algorithm in Table V-C at k -nearest = 102. We will compare these results against those in Table V-B. For the LOS channel, the TW results for the PCA, SM, and AE algorithms are in the range of 0.8272–0.8603, whereas for the LR and ISQ algorithms, they are in the range of 0.9885–0.9930. On the other hand, for the same channel, TW results with the MM algorithm are in the range of 0.9975–0.9998, a big difference. Looking at the TW performance for the QLOS and QNLOS channels, the performance of the PCA, SM, and AE algorithms are in the range of 0.8474–0.8574 while those of the LR and ISQ algorithms are in the 0.9029–0.9092 range. Yet, the TW results of the MM algorithm for these two channels are in the range of 0.9622–0.9976. We will now carry out the same analysis for the CT measure. In the LOS channel, the performance of PCA, SM, and AE are in the range of 0.8932–0.9288, whereas those of the LR and ISQ are in 0.9940–0.9968. On the other hand, for the same channel, the MM algorithm performance is in the range of 0.9992–1.000. For the QLOS and QNLOS channels, the CT performance of the PCA, SM, and AE algorithms are in the range 0.9055–0.9278, whereas those of the LR and ISQ algorithms are in 0.9220–0.9416. On the other hand, the performance of the MM algorithm is in the range 0.9626–0.9992, another big jump. As a result of all of this analysis, we can say that the MM algorithm significantly outperforms all of PCA, SM, AE, LR, and ISQ algorithms. The differences are very significant against those of the PCA, SM, and AE algorithms.

We will now study the channel charts created by the MM algorithm for the LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers in Fig. 10, comparing them with the channel charts by the PCA, SM, and AE algorithms on the same channels in Fig. 8. For the LOS channel, clearly, the MM algorithm has the best visual outcome, preserving the word “VIP” very clearly. The quality of the channel chart generated by the MM algorithm is clearly much better than those by the PCA, SM, and AE algorithms, and better than those by LR and ISQ. It can be stated that this is the case even for the QLOS channel, the word “VIP” is being mostly visible for the MM algorithm, while that is not the case for PCA, SM, and AE, and not as clearly for LR and ISQ. Even for the QNLOS channel, the visual outcome by the MM algorithm is better than those by PCA, SM, and AE, albeit the clarity of the word “VIP” is no longer the case. It can be stated that, for this channel, the visual outcome by the MM algorithm is better than those by LR and ISQ, although the difference is not as clear as in the cases of the LOS and QLOS channels.

Finally, we will compare the TW and CT performance of the MM algorithm on LOS, QLOS, and QNLOS channels in Fig. 11 against those of the PCA, SM, AE, LR, and ISQ algorithms in Fig. 9. The MM algorithm clearly outperforms the PCA, SM, and AE algorithms for all of LOS, QLOS, and QNLOS channels. It also outperforms the LR and ISQ algorithms on the QLOS and QNLOS channels. For the LOS channel, the performance of the LR and ISQ algorithms are close to that of the MM algorithm. In summary, comparison of Fig. 11 with Fig. 9 shows that the MM algorithm overperforms all of the PCA,

SM, AE, LR, and ISQ algorithms, except for the LOS channel, where the MM, LR, and ISQ algorithms have similar TW and CT performance.

D. JM Algorithm

We will now discuss performance results of the JM algorithm introduced in Sec. III-C. After some experimentation, we decided a subarray of dimensions $N_{Sa} = 4$ and $M_{Sa} = 4$ worked best. We employed our thresholding and peak search algorithm. The complexity increase is very high. We presented the results of the MM algorithm in Table V-C, Fig. 10, and Fig 11. We present similar results for the JM algorithm in Table IV, Fig. 12, and Fig 13 except we do not present the case with 2 subcarriers because the smoothing subarray is 4×4 and the case of 2 subcarriers cannot run.

First, let us compare the TW and CT results at k -nearest = 102 for the MM algorithm in Table V-C and for the JM algorithm in Table IV. It can be observed that for the LOS and QLOS channels, the TW and CT performance of the MM algorithm and the JM algorithm are almost the same. For the QNLOS channel, there is a very slight advantage towards the JM algorithm, for both TW and CT. Because the MM and JM algorithm results are very close, we can say that the JM algorithm, like the MM algorithm, outperforms PCA, SM, AE, ISQ, and LR algorithms at k -nearest = 102 in terms of both TW and CT.

Second, we compare the channel charts for the MM algorithm in Fig. 10 and the JM algorithm in Fig. 12. We deem that the visual outcome in both figures is very similar for the same channel and for the same number of subcarriers. And because of this similarity, we can say that the JM algorithm outperforms the PCA, SM, AE, ISQ, and LR algorithms in terms of channel charts.

Finally, we compare the TW and CT performance of the JM algorithm in Fig. 13 with that of the MM algorithm in Fig. 11 against k -nearest neighbors. For the LOS channel, it can be stated that the performance of both algorithms, in the left plots of Fig. 11 and Fig. 13, are very close. On the other hand, a very slight advantage towards the JM algorithm can be observed in the case of QLOS and QNLOS channels in the middle and right subplots of Fig. 11 and Fig. 13. Since the performance of the JM algorithm and the MM algorithm are very close, we can say that the JM algorithm outperforms the PCA, SM, AE, LR, and ISQ algorithms for TW and CT against k -nearest neighbors.

In summary, the performance of the MM algorithm and that of the JM algorithm are very close, with a very slight advantage towards the JM algorithm. But, in the face of the enormous complexity of the JM algorithm, we cannot say that it would be the preferred choice.

E. RS Algorithm

We have carried out extensive simulations on the performance of the RS algorithm as compared to the MM and JM algorithms. We will state that the performance of the three algorithms are very close, almost

the same. In order not to be very repetitive, we only provide Table V for comparison with Table V-C. Clearly, the results are very close. Algorithms 3 and 4 employ a large number of multiplications. On the other hand, there may be a very important advantage of the RS algorithm against the MM and JM algorithms in terms of complexity. Both MM and JM algorithms employ 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. For that reason, its implementation can be carried out in RTL whereas the implementation of MUSIC may require the use of an additional processor such as a digital signal processing (DSP) unit. DSP cores are available in embedded systems for wireless applications and there are implementations of MUSIC on them [37]. Thus, although the operation of RS requires many multiplications, its implementation may be more straightforward than that of MUSIC.

F. 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 algorithms do 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 V-F. The simulation time is not dependent on the channel, the scale, or the geometry of the environment. The simulation time only relies on the number of UEs, the number of antennas, and the number of subcarriers. In Table V-F, 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. Assume we have N vectors of feature size D , to be reduced to a size d . The computationally intensive operations of PCA consist of two major steps: *i*) computing the covariance matrix, and *ii*) computing the eigenvalue decomposition of the covariance matrix. The first step involves the multiplication of two matrices of size $D \times N$ and $N \times D$, resulting in $O(ND^2)$. The second step involves the eigenvalue decomposition of a $D \times D$ matrix, contributing $O(D^3)$, with a total complexity of $O(ND^2 + D^3)$, see e.g., [38].

Although it has advantages over PCA, SM is stated to be substantially more complex than PCA [1]. We note that SM is nonlinear and needs multiple runs until convergence, whose number is not known prior to running the algorithm [39]. The high complexity of the SM method is further emphasized in the literature, e.g., [40]–[42].

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 [43]. 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 $O(N_R^2P)$ to determine θ , where we ignore a second term $N_R^2 \cdot 1$ since $P \gg 1$. 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 $2N_R$ multiplications, $N_R + 1$ 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 [44]. In any case, the complexity of the ISQ algorithm will be dominated by the MUSIC algorithm for θ , which has $O(N_R^2P)$, as discussed above. 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 $2N_R + 1$ multiplications, 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 $O(N_R^2P)$, as in the ISQ algorithm. For the MM algorithm, we use MUSIC to estimate 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 $O(N_R^2P + N_R^2N_S)$. 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 $O(N_R^2 P + N_R^2 N_S + N_S^2 Q + N_S^2 N_R)$.

We will now compare two training-based algorithms AE and LR in terms of complexity. For LR, estimating AOA is carried out by using the MUSIC algorithm for θ . For ρ , linear regression is performed. Let the training set be $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)$, where $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $i = 1, 2, \dots, N$. Note that, in our case, $d = 2$, $N = 256$. Then, the training computational complexity is given as $O(d^2 N + d^3)$, see e.g., [45]. On the other hand, it is difficult to come up with a simple formulation for the training complexity of the AE algorithm as it depends on many factors such as the number of layers, the number of neurons at each layer, the number of training symbols, the number of epochs, and the algorithm used for training. However, the AE in [1] consists of ten layers with 500, 100, 50, 20, D' , D' , 20, 50, 100, and 500 neurons in its layers where $D' = 2$ or 3. It uses the Tensorflow algorithm for learning. It is very clear that the computational complexity of training or running this AE is substantially higher than the LR algorithm, including the MUSIC algorithm employed to calculate the AOA. The prediction complexity of the LR algorithm has $O(N_R^2 P)$ due to the MUSIC algorithm for θ . In addition, $2N_R + 1$ multiplications, N_R square roots, N_R adds, and a log operation are needed. Whereas, the prediction complexity of the AE algorithm in [1] is extremely high due to the large number of neurons employed, i.e., 2×500 , 2×100 , 2×50 , 2×20 , $2 \times D'$.

To address the complexity of the RS algorithm, consider Algorithm 3. Calculation of the \mathbf{S} matrix takes $N_S N_R^2$ multiplications. As before, let P be the number of ϕ values to be tested (given as 180 in Algorithm 3). Then, first, P times $N_R^2 + 1$ multiplications are needed. Then, to calculate $C(\phi)$, P times N_R^2 multiplications are needed. As a result, the total number of multiplications for the RS algorithm to estimate θ is $P(2N_R^2 + 1) + N_S N_R^2$.

Algorithm 4 is based on the same technique as Algorithm 3, but using column vectors of the CSI matrix, instead of row vectors. It then provides an estimate of ρ . We will now calculate its complexity in a fashion similar to Algorithm 3. Calculation of the \mathbf{S} matrix takes $N_R N_S^2$ multiplications. As before, let Q be the number of σ values to be tested (given as 1000 in Algorithm 4). Then, Q times $N_S^2 + 1$ multiplications are needed for the \mathbf{B} matrices. Ignoring division of \mathbf{S} by N_R , to calculate $C(\sigma)$, Q times N_S^2 multiplications are needed. As a result, the total number of multiplications for the RS algorithm to estimate ρ is $N_\sigma(2N_S^2 + 1) + N_R N_S^2$.

We would like to note that, both Algorithm 3 and 4, 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 provide a summary of the required number of multiplications in the algorithms we discussed in Table V-F.

G. Performance at Different SNR

We would like to compare the performance of the eight algorithms discussed in this paper at SNR values different than 0 dB. To that end, first, we provide results for the eight algorithms at SNR = -10 dB in Table V-G. These results are obtained employing the number of antennas equal to 32 and the number of subcarriers equal to 32. Thus, they should be compared with Table V-B for PCA, SM, AE, LR, and ISQ, and the last columns of Tables V-C–V. First, we observe that there are significant drops in TW and CT performance for the first five algorithms PCA, SM, AE, LR, and ISQ. As can be expected, the drops are more significant for the QLOS and QNLOS channels. On the other hand, although there are some drops in the TW and CT performance of the MM, JM, and RS algorithms, they are much less as compared to the first five algorithms PCA, SM, AE, LR, and ISQ. We ran simulations at SNR = 10 dB as well, but found out that the improvements in TW and CT remained modest.

We conclude that although PCA, SM, AE, LR, and ISQ algorithms present TW and CT performance degradation in the case of SNR drop from 0 dB to -10 dB, the algorithms MM, JM, and RS, albeit showing some drop, remain quite robust, much more so than PCA, SM, AE, LR, and ISQ.

H. Effect of Phase Noise

As the techniques LR, ISQ, MM, and JM in this paper depend on the MUSIC algorithm for determination of the AOA, the effect of phase noise on them can be questioned. To that end, we refer to [46], which specifies that, at 1 MHz offset, the phase noise of the phase-locked loop should be at or below -102 dBc/Hz (about -100 dBc/Hz). Incorporating phase noise to the matrix \mathbf{R} via a Matlab function, we tabulated the TW and CT performance of the eight algorithms in this paper with phase noise equal to -100 dBc/Hz at 1 MHz offset in Table V-G with an SNR = 0 dB. In its second row for each entry in Table V-G, we have made a comparison of the percentage changes in the first row of the entry in Table V-G with the corresponding value in Tables V-B–V. These changes can be positive, indicating we have *worse* performance with phase noise, or negative, indicating *better* performance with phase noise, which may be considered counterintuitive. We note that the randomness in the process does give rise to both positive and negative entries. First, the TW and CT performance of all the eight algorithms remains within close proximity (positive or negative) of the performance without phase noise. Second, we note that

in Table V-G, there are 21 negative entries out of 48 total. When we increase the phase noise to -70 dBc/Hz at 1 MHz with an SNR = 0 dB, the number of negative entries reduces to 6 out of 48, indicating worsening performance with increased phase noise, as expected. Nevertheless, as we stated above, we conclude from Table V-G that the performance with phase noise when it is within acceptable limits indicated in [46], employing the algorithms studied in this work, remains within close proximity of the performance without phase noise on TW and CT performance. Note that TW and CT are neighborhood measures and although on the average they become worse with increasing phase noise, this behavior happens not to be uniform as discussed above. On the other hand, we measured the Error Vector Magnitude (EVM) performance with the simulated system, which illustrated the uniformly degrading EVM performance with increased phase noise, as expected.

VI. CONCLUSION

The LR, ISQ, and MM algorithms presented in this paper significantly outperform the three algorithms in the seminal paper [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. We studied an algorithm from the literature, intended for a different application, we call the JM algorithm. This algorithm has only very slight improvement over the MM algorithm at a very large increase in computational complexity. Finally, we introduced the RS algorithm whose performance is about the same as the MM and JM algorithms. The advantage of this algorithm is that it does not need the eigenvector and eigenvalue decomposition needed by the MM and JM algorithms, and furthermore, its RTL implementation may be more straightforward as it does not require eigenvalue and eigenvector analysis.

Note that the MUSIC algorithm for θ is model based. In addition, the ISQ algorithm for ρ is also model based. Finally, the MM, JM, and RS algorithms are completely model based for both θ and ρ . The performance of ISQ, MM, JM, and RS are better than those of PCA, SM, and AE of [1]. PCA and SM employ standard techniques of dimensionality reduction whereas AE is training based.

Reference [1] discusses an extension of the SM algorithm, called the SM+ algorithm. This algorithm employs a penalty in the objective function that keeps temporally adjacent points in the local geometry nearby in the channel chart. In [1, Fig. 6 and Fig. 7], it is stated that the TW and CT performance of

SM and SM+ are close while SM+ provides more visually satisfying results than PCA, SM, and AE. Yet, comparing [1, Fig. 6] with our Fig. 8 and Fig 10, one can observe that LR, ISQ, and MM provide better visual outcomes than SM+. We have observed the same conclusion for the RS algorithm, which we are not presenting due to avoiding repetitive figures. For that reason, we did not attempt to simulate the SM+ algorithm in this paper. One can conclude based on the comparisons discussed above that MM, JM, and RS will outperform SM+ in TW and CT as well as the visual quality of the channel charts.

We have carried out extensive simulations for the bias and variability of our estimation algorithms. The results show that their bias and variability vanish asymptotically. A theoretical analysis of these facts is beyond the scope of this paper and is left as potential future work.

VII. ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviewers whose comments improved the presentation in the paper.

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 LR and ISQ algorithms in 3D channel.

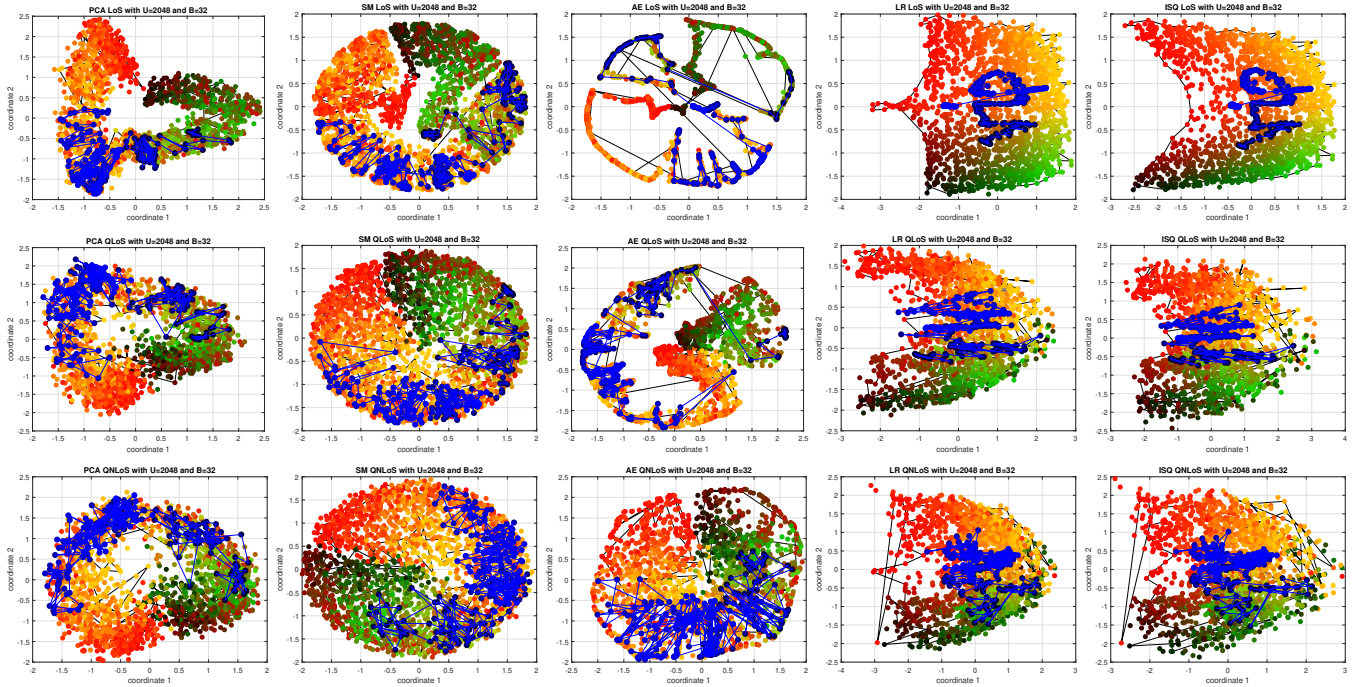


Fig. 8: Channel charts with PCA, SM, AE, LR, and ISQ algorithms for the 3D LOS, QLOS, and QNLOS channels.

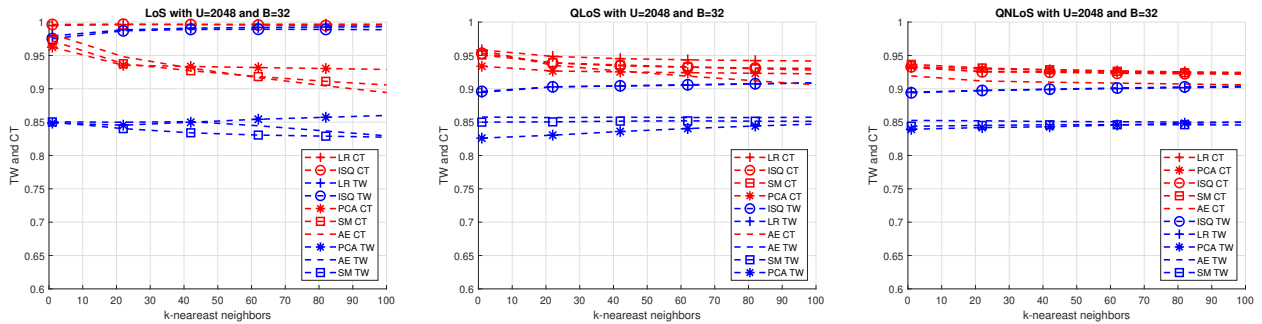


Fig. 9: TW and CT performance against k -nearest neighbors for LR and ISQ algorithms in 3D. Left: LOS, middle: QLOS, right: QNLOS.

Measure	Channel	2sc	8sc	14sc	20sc	26sc	32sc
TW	LOS	0.9975	0.9986	0.9992	0.9997	0.9996	0.9998
	QLOS	0.9817	0.9958	0.9970	0.9973	0.9976	0.9976
	QNLOS	0.9622	0.9802	0.9818	0.9825	0.9837	0.9856
CT	LOS	0.9992	0.9999	1.0000	0.9999	0.9999	1.0000
	QLOS	0.9816	0.9972	0.9985	0.9990	0.9993	0.9992
	QNLOS	0.9626	0.9803	0.9821	0.9833	0.9845	0.9865

Table 3: Performance comparison for TW and CT at k -nearest = 102 for MM algorithm in 3D channel at 2, 8, 14, 20, 26, and 32 subcarriers.

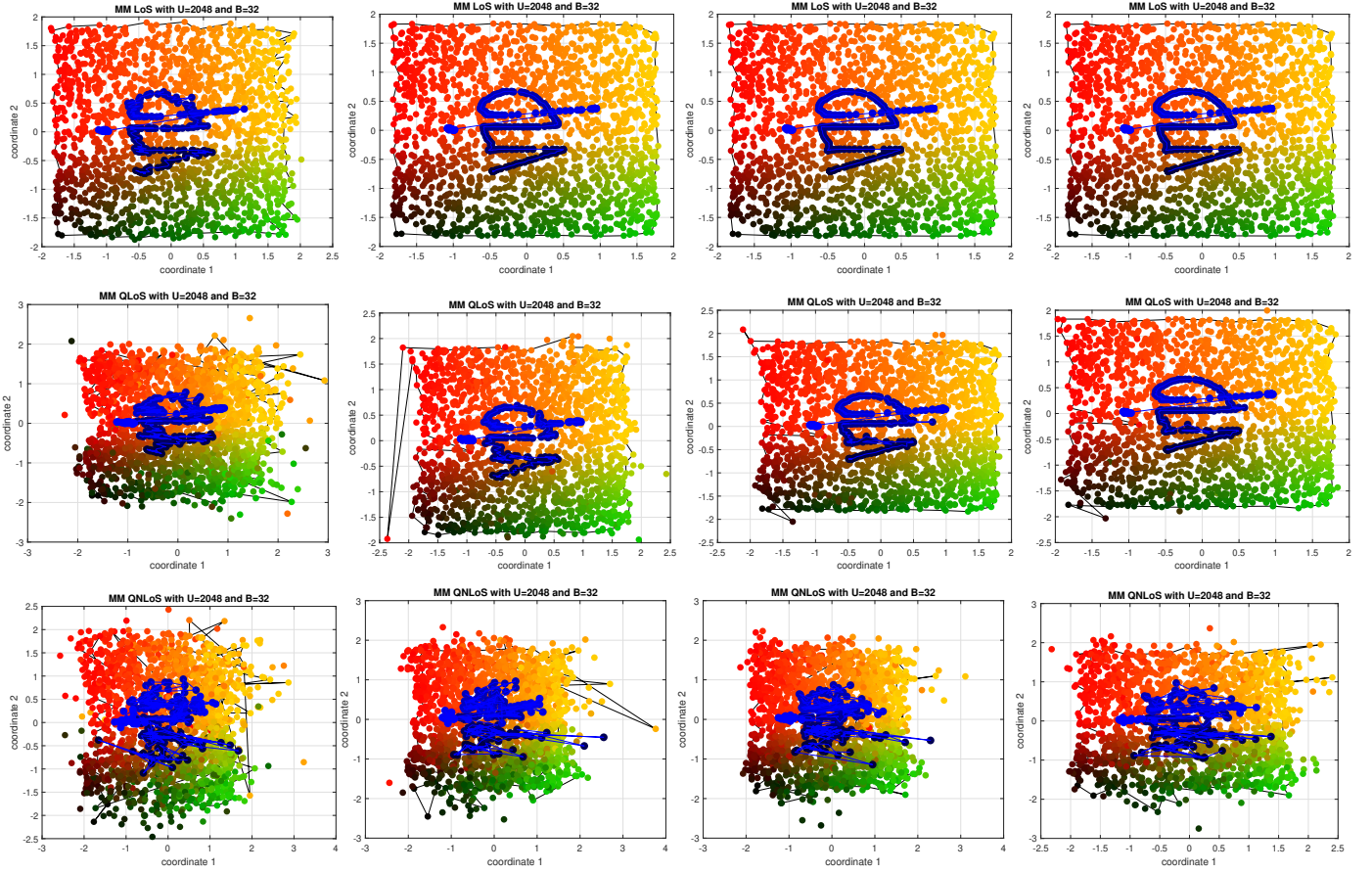


Fig. 10: Channel charts with the MM algorithm for the 3D LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers.

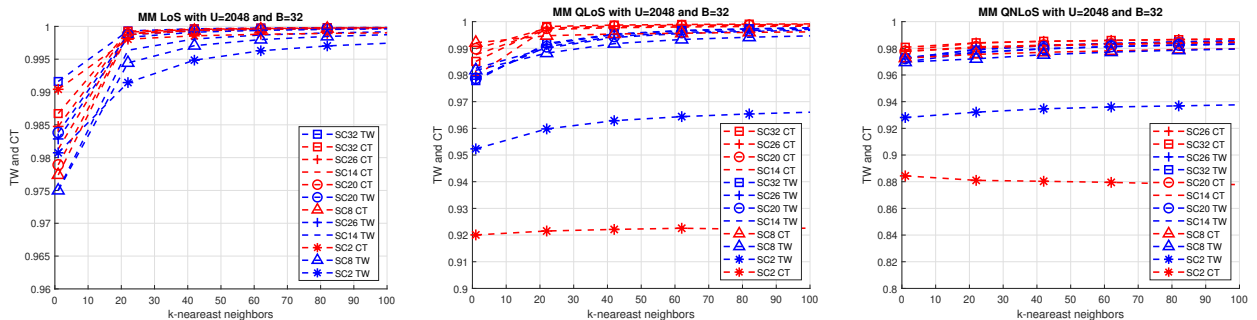


Fig. 11: TW and CT performance against k -nearest neighbors for MM algorithm in 3D channel. Left: LOS, middle: QLOS, right: QNLOS.

Measure	Channel	8sc	14sc	20sc	26sc	32sc
TW	LOS	0.9981	0.9984	0.9983	0.9982	0.9982
	QLOS	0.9936	0.9967	0.9966	0.9963	0.9958
	QNLOS	0.9798	0.9899	0.9888	0.9870	0.9894
CT	LOS	1.0000	1.0000	1.0000	1.0000	1.0000
	QLOS	0.9951	0.9985	0.9984	0.9981	0.9977
	QNLOS	0.9821	0.9916	0.9904	0.9886	0.9910

Table 4: Performance comparison for TW and CT at k -nearest = 102 for JM algorithm in 3D channel at 8, 14, 20, 26, and 32 subcarriers.

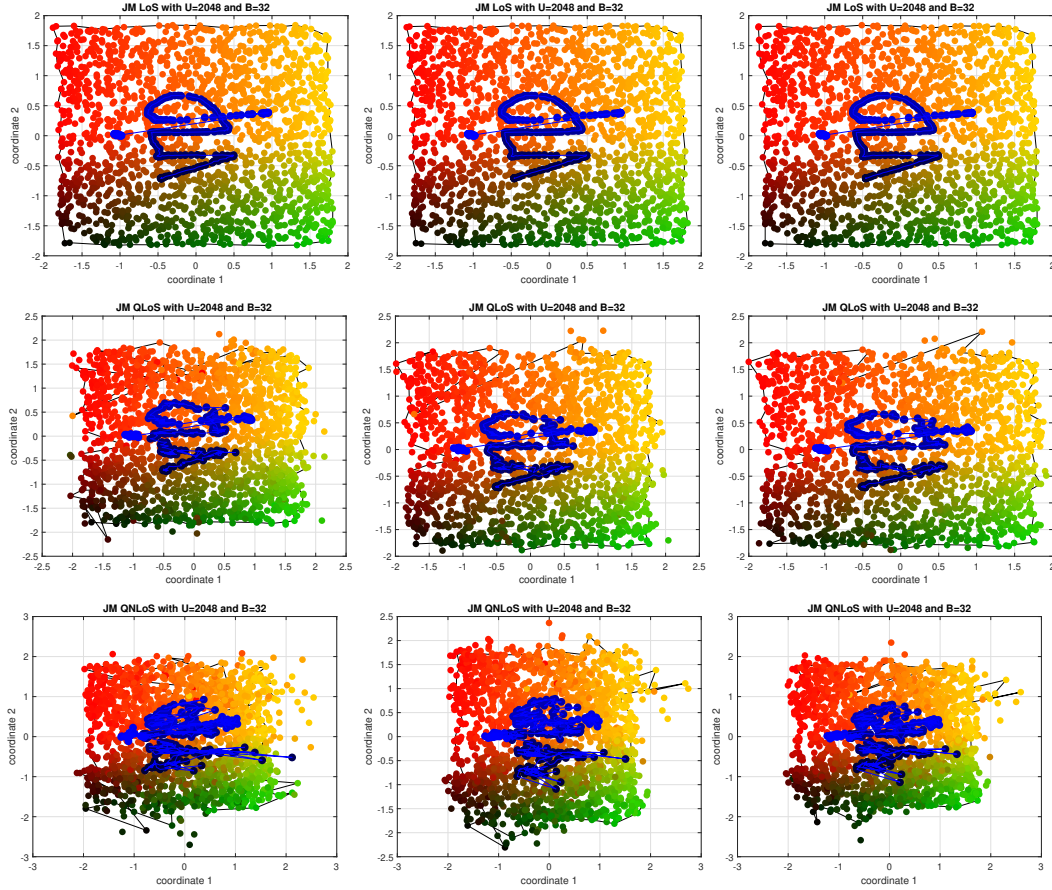


Fig. 12: Channel charts with the JM algorithm for the 3D LOS, QLOS, and QNLOS channels at 8, 20, and 32 subcarriers.

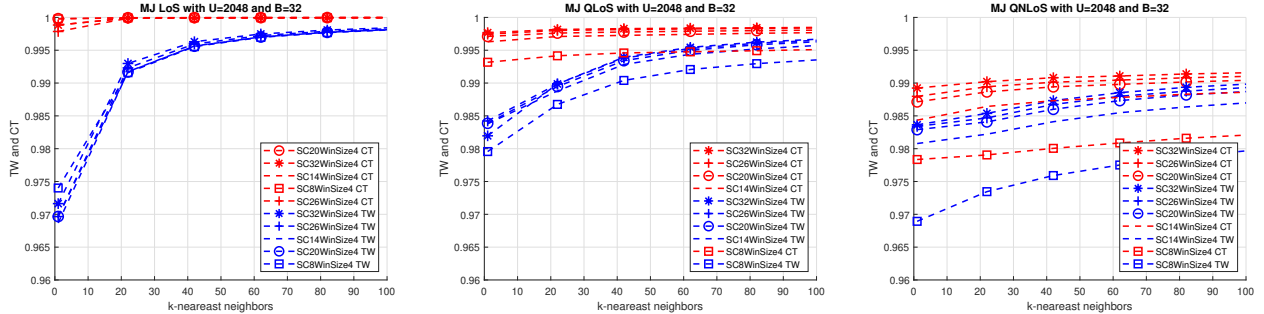


Fig. 13: TW and CT performance against k -nearest neighbors for JM algorithm in 3D channel. Left: LOS, middle: QLOS, right: QNLOS.

Measure	Channel	2sc	8sc	14sc	20sc	26sc	32sc
TW	LOS	0.9972	0.9986	0.9991	0.9996	0.9997	0.9998
	QLOS	0.9648	0.9903	0.9933	0.9943	0.9946	0.9950
	QNLOS	0.9421	0.9664	0.9753	0.9779	0.9804	0.9821
CT	LOS	0.9982	0.9997	0.9998	0.9998	0.9998	0.9998
	QLOS	0.9572	0.9920	0.9951	0.9959	0.9965	0.9968
	QNLOS	0.9337	0.9680	0.9784	0.9813	0.9837	0.9854

Table 5: Performance comparison for TW and CT at k -nearest = 102 for RS algorithm in 3D channel at 2, 8, 14, 20, 26, and 32 subcarriers.

Algorithm	Simulation time (seconds)
PCA	0.817
SM	12.2
AE	53.9
LR	7.15
ISQ	7.09
MM	20.4
JM	442.0

Table 6: Simulation times.

Algorithm	Number of Multiplications
PCA	$O(ND^2 + D^3)$
SM	\gg PCA
AE	\gg LR, ISQ
LR	$O(N_R^2 P)$
ISQ	$O(N_R^2 P)$
MM	$O(N_R^2 P + N_R^2 N_S + N_S^2 Q + N_S^2 N_R)$
RS	$O(P(2N_R^2 + 1) + N_S N_R^2 + Q(2N_S^2 + 1) + N_R N_S^2)$

Table 7: Number of multiplications employed by the algorithms in the paper. N : number of vectors of feature size D , to be reduced to size d . N_R : Number of antennas at the BS, N_S : number of subcarriers. P : Number of potential AOAs to determine θ , Q : number of potential distances to determine ρ .

Measure	Channel	PCA	SM	AE	LR	ISQ	MM	JM	RS
TW	LOS	0.7494	0.8132	0.8155	0.9275	0.9146	0.9991	0.9990	0.9958
	QLOS	0.6665	0.6968	0.7260	0.7467	0.7470	0.9774	0.9362	0.9612
	QNLOS	0.6860	0.6813	0.6981	0.7223	0.7192	0.9361	0.9652	0.9133
CT	LOS	0.7919	0.8594	0.8467	0.9403	0.9364	0.9998	0.9998	0.9942
	QLOS	0.7570	0.7852	0.8035	0.7670	0.7686	0.9809	0.9456	0.9725
	QNLOS	0.7649	0.7679	0.7773	0.7404	0.7356	0.9465	0.9687	0.9372

Table 8: Performance comparison for TW and CT at k -nearest = 102 in 3D channel, SNR = -10 dB.

Measure	Channel	PCA	SM	AE	LR	ISQ	MM	JM	RS
TW	LOS	0.8566 0.43%	0.7741 6.4%	0.8402 -1.4%	0.9932 -0.020%	0.9888 -0.030%	0.9994 0.040%	0.9984 -0.020%	0.9960 0.38%
	QLOS	0.8450 0.28%	0.8307 2.4%	0.8536 4.4%	0.9078 0.12%	0.9085 0.077%	0.9977 -0.010%	0.9969 -0.11%	0.9939 0.11%
	QNLOS	0.8487 0.18%	0.8451 0.059%	0.8528 -0.38%	0.9059 -0.33%	0.9068 -0.30%	0.9859 -0.030%	0.9910 -0.16%	0.9820 0.010%
CT	LOS	0.9276 0.13%	0.8868 2.0%	0.8534 4.5%	0.9968 0	0.9941 -0.010%	0.9999 0.010%	1.0000 0	0.9943 0.55%
	QLOS	0.9215 0.087%	0.8881 4.3%	0.9268 -2.4%	0.9402 0.15%	0.9292 0.13%	0.9991 0.010%	0.9987 -0.10%	0.9933 0.35%
	QNLOS	0.9239 -0.22%	0.9219 -0.022%	0.9108 -0.56%	0.9283 -0.40%	0.9254 -0.37%	0.9874 -0.091%	0.9929 -0.19%	0.9829 0.25%

Table 9: Performance comparison for TW and CT at k -nearest = 102 in 3D channel, phase noise -100 dBc/Hz @ 1 MHz offset, SNR = 0 dB. Numbers in the second rows for each entry are the percentage differences with the entries in Tables V-B–V.

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] J. Deng, S. Medjkouh, N. Malm, O. Tirkkonen, and C. Studer, "Multipoint channel charting for wireless networks," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 286–290.
- [3] P. Huang, O. Castaneda, E. Gonultas, S. Medjkouh, O. Tirkkonen, T. Goldstein, and C. Studer, "Improving channel charting with representation-constrained autoencoders," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.
- [4] E. Lei, O. Castaneda, O. Tirkkonen, T. Goldstein, and C. Studer, "Siamese neural networks for wireless positioning and channel charting," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 200–207.
- [5] P. Agostini, Z. Utkovski, and S. Stanczak, "Channel charting: An Euclidean distance matrix completion perspective," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5010–5014.
- [6] C. Geng, H. Huang, and J. Langerman, "Multipoint channel charting with multiple-input multiple-output convolutional autoencoder," in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1022–1028.
- [7] P. Kazemi, H. Al-Tous, C. Studer, and O. Tirkkonen, "SNR prediction in cellular systems based on channel charting," in *2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*, 2020, pp. 1–8.
- [8] P. Ferrand, A. Decurninge, L. G. Ordonez, and M. Guillaud, "Triplet-based wireless channel charting," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.
- [9] J. Pihlajasalo, M. Koivisto, J. Talvitie, S. Ali-Loytty, and M. Valkama, "Absolute positioning with unsupervised multipoint channel charting for 5G networks," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, 2020, pp. 1–5.
- [10] L. Ribeiro, M. Leinonen, H. Djelouat, and M. Juntti, "Channel charting for pilot reuse in mMTC with spatially correlated MIMO channels," in *2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6.
- [11] P. Ferrand, A. Decurninge, L. G. Ordonez, and M. Guillaud, "Triplet-based wireless channel charting: Architecture and experiments," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2361–2373, 2021.
- [12] H. Al-Tous, O. Tirkkonen, and J. Liang, "Adaptive sector splitting based on channel charting in massive MIMO cellular systems," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–6.
- [13] P. Kazemi, T. Ponnada, H. Al-Tous, Y.-C. Liang, and O. Tirkkonen, "Channel charting based beam SNR prediction," in *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2021, pp. 72–77.
- [14] J. Deng, O. Tirkkonen, J. Zhang, X. Jiao, and C. Studer, "Network-side localization via semi-supervised multi-point channel charting," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 2021, pp. 1654–1660.
- [15] L. Le Magoarou, "Efficient channel charting via phase-insensitive distance computation," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2634–2638, 2021.
- [16] B. Rappaport, E. Gonultas, J. Hoydis, M. Arnold, P. K. Srinath, and C. Studer, "Improving channel charting using a split triplet loss and an inertial regularizer," in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, 2021, pp. 1–6.
- [17] Q. Zhang and W. Saad, "Semi-supervised learning for channel charting-aided IoT localization in millimeter wave networks," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.
- [18] P. Agostini, Z. Utkovski, S. Stanczak, A. A. Memon, B. Zafar, and M. Haardt, "Not-too-deep channel charting (N2D-CC)," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 2160–2165.
- [19] T. Yassine, L. L. Magoarou, S. Paquelet, and M. Crussiere, "Leveraging triplet loss and nonlinear dimensionality reduction for on-the-fly channel charting," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.
- [20] P. Agostini, Z. Utkovski, and S. Stanczak, "Federated learning for multipoint channel charting," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.

- [21] F. Euchner, P. Stephan, M. Gauger, S. Dorner, and S. Ten Brink, "Improving triplet-based channel charting on distributed massive MIMO measurements," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.
- [22] L. Ribeiro, M. Leinonealn, I. Rathnayaka, H. Al-Tous, and M. Juntti, "Channel charting aided pilot allocation in multi-cell massive MIMO mMTC networks," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022, pp. 1–5.
- [23] P. Kazemi, H. Al-Tous, C. Studer, and O. Tirkkonen, "Channel charting assisted beam tracking," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 2022, pp. 1–5.
- [24] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
- [25] R. Zhang, Y.-H. Quan, S.-Q. Zhu, L. Yang, Y. Li, and M.-D. Xing, "Joint high-resolution range and DOA estimation via MUSIC method based on virtual two-dimensional spatial smoothing for OFDM radar," *Hindawi International Journal of Antennas and Propagation*, vol. 2018, 2018.
- [26] 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 Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3242–3256, 2014.
- [27] "Quadriga: The next generation radio channel model," <https://quadriga-channel-model.de/>.
- [28] S. Jaeckel, "Quasi-deterministic channel modeling and experimental validation in cooperative and massive MIMO deployment topologies," Ph.D. dissertation, Technische Universitat Ilmenau, 2016.
- [29] L. L. Magoarou, T. Yassine, S. Paquelet, and M. Crussiere, "Channel charting based beamforming," in *2022 56th Asilomar Conference on Signals, Systems, and Computers*, 2022, pp. 1185–1189.
- [30] C. Studer, E. Gonultas, and S. Medjkouh, "Simple channel charting MATLAB simulator," <https://github.com/IIP-Group/ChannelCharting>.
- [31] R. Jain, "Channel models: A tutorial," [Online]. Available: https://www.cse.wustl.edu/~jain/wimax/ftp/channel_model_tutorial.pdf, Feb. 2007.
- [32] D. Pinchera, A. L. Imoize, A. E. Ibhaze, A. A. Atayero, and K. V. N. Kavitha, "Standard propagation channel models for MIMO communication systems," *Hindawi Wireless Communications and Mobile Computing Journal*, vol. 2021, pp. 1–36, 2021.
- [33] L. Pang, J. Zhang, Y. Zhang, X. Huang, Y. Chen, and J. Li, "Investigation and comparison of 5G channel models: From QuaDRiGa, NYUSIM, and MG5G perspectives," *Chinese Journal of Electronics*, vol. 31, pp. 1–17, 2022.
- [34] K. Benoit, "Linear regression models with logarithmic transformations," Methodology Institute, London School of Economics, UK. [Online]. Available: <https://kenbenoit.net/assets/courses/ME104/logmodels2.pdf>, 2011.
- [35] A. Aly, "Estimation of cellular wireless user coordinates via channel charting and MUSIC," Master's thesis, University of California, Irvine, 2022.
- [36] C. Studer, Private communication, Nov. 2021.
- [37] Y. Liu and H. Cui, "Antenna array signal direction of arrival estimation on digital signal processor (DSP)," *Procedia Computer Science*, vol. 55, pp. 782–791, 2015.
- [38] A. Banerjee, "Computational complexity of PCA," [Online]. Available: alekhyo.medium.com/computational-complexity-of-pca-4cb61143b7e5, Aug. 2020.
- [39] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein, "On the initialisation of Sammon's nonlinear mapping," *Springer Pattern Analysis and Applications Journal*, vol. 3, pp. 61–68, Feb. 2000.
- [40] G. Biswas, A. K. Jain, and R. C. Dubes, "Evaluation of projection algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 6, pp. 701–708, 1981.
- [41] H. Yin, "ViSOM - A novel method for multivariate data projection and structure visualization," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 237–243, 2002.
- [42] —, "Nonlinear multidimensional data projection and visualisation," in *Intelligent Data Engineering and Automated Learning, IDEAL 2003. Lecture Notes in Computer Science*, J. Liu, Y.-m. Cheung, and H. Yin, Eds. Springer Berlin Heidelberg, 2003, pp. 377–388.

- [43] 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.
- [44] H. Alt, "Square rooting is as difficult as multiplication," *Computing*, vol. 21, pp. 221–232, 1979.
- [45] C.-J. Hsieh, "CS260: Machine learning algorithms Lecture 2: Linear regression and classification," [Online]. Available: web.cs.ucla.edu/~chohsieh/teaching/CS260_Winter2019/lecture2.pdf, Jan. 2019.
- [46] Ericsson, "TP to TR 38.220: Phase noise trends and example parameterized phase noise model in subclause 5.5.3 and Annex B. 3GPP TSG-RAN Working Group 4 (Radio) meeting # 94-c Electronic Meeting," Feb. 2020.