UNIVERSITY OF CALIFORNIA,
IRVINE

Estimation of Cellular Wireless User Coordinates via Channel Charting and MUSIC

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical Engineering

by

Amr Aly

Thesis Committee:
Professor Ender Ayanoglu, Chair
Professor Lee Swindlehurst
Professor Filippo Capolino

2022

DEDICATION

To my wife Samar, for standing beside me, encouraging me, and taking care of me throughout this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Ender Ayanoglu for providing me with precious advice, and his invaluable experience, throughout my failed and succeeded trials to finish the research encompassed by this thesis.

ABSTRACT OF THE THESIS

Estimation of Cellular Wireless User Coordinates via Channel Charting and MUSIC

By

Amr Aly

Master of Science in Electrical Engineering

University of California, Irvine, 2022

Professor Ender Ayanoglu, Chair

We present a new way of producing a channel chart in polar coordinates. 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), Samson's method (SM), and autoencoder (AE). We show that ISQ, LR, and MM outperform all three in performance. The performance of MM is much better than LR and ISQ in LOS channel conditions. MM has better performance than ISQ and LR but is more complex. ISQ and LR have similar performance with ISQ having less complexity than LR.

Chapter 1

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, 2]. This chart helps the BS locate the UEs (relatively), which can help in many applications such as handover, cell search, user localization, and more. We need to stress that a channel chart is not intended to preserve the absolute distances between UEs, rather it is meant to give information on the proximity of UEs to each other and to the BS. Previous papers have proposed estimation of a channel chart using Cartesian coordinates. Reference [1] compared three algorithms, namely principal component analysis (PCA), Sammon's mapping (SM), and autoencoder (AE). All three algorithms try to convert the channel coefficients into the angular domain and then try to extract two features for each UE. These two features represent the UE location in the channel chart. Reference [3] 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. In this thesis our approach is model based. It is not based on training. We estimate the angle and distance based on the phase and magnitude of the channel state information (CSI). We need to stress that the novelty of this thesis is not about the estimation

methods we used to estimate the angle of arrival (AOA) or distance. Some of the estimation algorithms we used are discussed in previous papers such as [4]. The novelty comes from using these estimation algorithms jointly to calculate polar coordinates in order to produce the channel chart. We would like to stress that these estimation algorithms individually might have suboptimal mean squared errors and they can be biased estimators. On the other hand, in channel charting they can perform in a way that result in outperforming previous algorithms. This performance is measured by two quantities called Continuity and Trustworthiness which we explain later in this chapter. We first estimate the AoA θ using the MUSIC algorithm [4]. The MUSIC algorithm is used to estimate AoA based on the correlation matrix of the channel coefficients. For the distance between the BS and the UE ρ, we propose three algorithms. In the first algorithm, we sum the power on all antennas for each UE, then we take the inverse of the root of the sums and use it as ρ. In the second algorithm, we consider linear regression of 256 UEs known location with CSI power to estimate a slope and intercept. In the third algorithm, we use MUSIC with multiple subcarriers to estimate the distance from the phase difference. We then compare the results of these approaches and those of PCA, SM, and AE.

1.1 Background

In this subsection, we will provide an introduction to topics that will be used in the following chapters.

1.1.1 Channel Models

Throughout this thesis, we employ three channel models, namely Vanilla LOS, Quadriga LOS, and Quadriga NLOS [5]. We start with the simplest, Vanilla LOS. Vanilla LOS is one

line-of-sight (LOS) ray. It is described as

$$h = \frac{1}{d^r}\, e^{-j\left(\frac{2\pi d}{\lambda}\right)} \tag{1.1}$$

where d is the distance between transmitter and receiver and r is known as the path loss exponent. As can be seen from (1.1), the magnitude and phase of the channel are deterministic, and can be determined by distance only. Next we discuss the Quadriga channel model [5]. Quadriga stands for quasi deterministic radio channel generator. It is a statistical three-dimensional ray tracing channel model. According to [5], it has the following features

- Three-dimensional propagation (antenna modeling, geometric polarization, scattering clusters).
- Continuous time evolution.
- Spatially correlated large- and small-scale fading.
- Transition between varying propagation scenarios.

The Quadriga model is very customizable. It has many features and details. We employ the following set of parameters

- Coordinates of the transmitters and receivers.
- Carrier frequency, bandwidth, and number of sub carriers.
- Number of clusters.
- Antenna shape, polarization, number of elements, and spacing between them.
- NLOS or LOS scenario.

We note that in the simplest representation, the magnitude of the channel is proportional to a Rician distributed random variable for the QLOS case and to a Rayleigh distributed random variable for the QNLOS case. This is multiplied by the inverse of the distance to the power of path loss exponent r. In this thesis we use the following parameters

Table 1.1: Simulation parameters.

Parameter	Value
Antenna array	Uniform Linear Array (ULA) with spacing $\lambda/2 = 7.495$ cm
Number of array antennas	32
Number of transmitters	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. 2.2.3))

1.1.2 Angle of Arrival and Steering Vector

From Fig. 1.1, 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 $e^{j(\pi\cos(\theta))}$. With this shift, we get what is called the steering vector

$$\mathbf{A}(\theta) = (1, e^{j\pi\cos(\theta)}, e^{j\pi 2\cos(\theta)}, \ldots, e^{j\pi(N-1)\cos(\theta)})^T. \tag{1.2}$$

This vector is essential in beamforming applications and in determining AOA, as we will see later when we use the MUSIC algorithm.

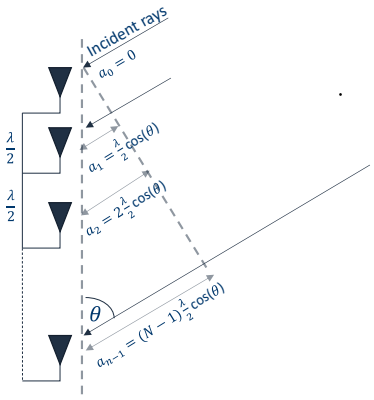

Figure 1.1: Angle of arrival (θ) relation with phase.

1.1.3 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. 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

$$\mathrm{CT}_i(K) = 1 - \frac{2}{K(2N-3K-1)} \sum_{j \in \mathcal{V}_K(\mathbf{u}_i)} (\hat{r}(i,j) - K). \tag{1.3}$$

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

$$\mathrm{CT}(K) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{CT}_i(K). \tag{1.4}$$

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

$$\mathrm{TW}_i(K) = 1 - \frac{2}{K(2N - 3K - 1)} \sum_{j \in \mathcal{U}_K(\mathbf{v}_i)} (r(i,j) - K). \tag{1.5}$$

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

$$\mathrm{TW}(K) = \frac{1}{N} \sum_{i=1}^{N} \mathrm{TW}_i(K). \tag{1.6}$$

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

Chapter 2

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. We will first discuss how to estimate θ by using the MUSIC algorithm and then we will discuss three algorithms to estimate ρ. We will investigate the bias and variability of each estimator with different channels, number of antennas, and number of subcarriers. We must note that the concept of channel charting is not trying to accurately estimate the absolute values of ρ and θ, rather it aims at preserving the geometry and shape of the radio environment and the relative distances of the UEs. An unbiased estimator does not mean that the continuity and thrustworthiness will be good and vice versa .

2.1 Estimating θ Using MUSIC

We discussed the steering vector $\mathbf{A}(\theta)$ in the last chapter. This steering vector 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 $N \times 1$ where N is the number of antennas at the

BS. 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 $N \times p$ 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.

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 $1/2$ **do**
    Calculate the steering vector $\mathbf{A}(\theta)$
    Calculate the PMF$(\theta) = \frac{1}{\text{Norm}_2(\mathbf{A}^H(\theta)\mathbf{N})}$
end for
Search the PMF for a peak and find the corresponding θ

2.2 Estimating ρ

We will now discuss how to estimate ρ. The simple channel ray model can be depicted as

$$h = a(d)\, e^{j\left(\frac{2\pi d}{\lambda} + \phi\right)} \text{ where } a(d) \sim d^{-2}. \tag{2.1}$$

In (2.1), the first term in channel phase is linearly proportional with the distance d between the transmitter and the receiver. 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, $\sim d^{-2}$. The

number 2 in this expression is called the path loss exponent. For more crowded environments the path loss exponent can be 3 or 4. In what follows, we will use ρ in place of d in (2.1).

2.2.1 Estimating ρ Using ISQ

Our proposal is a rather direct and simple approach. We calculate the square root inverse of the sum of CSI magnitudes for all antennas as

$$\rho = \frac{1}{\sqrt{\sum_{n=0}^{N-1} \text{abs}(h_n)}}. \tag{2.2}$$

We refer to this algorithm as ISQ (inverse square root sum). The motivation for this algorithm comes from (1.1) with the path loss component $r = 2$, or (2.1). Based on this formulation, $\rho = d = 1/\text{abs}(h_n)$ and (2.2) is a way of calculating this in an average sense. We note that in this case estimated ρ is not to scale with the real ρ, but that will not affect the TW and CT as we will see later.

2.2.2 Estimating ρ Using LR

Earlier, we tried a supervised approach by assuming 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

$$\rho = aX + b \ , \text{ where } X = \log \sum_{n=0}^{N-1} \text{abs}(h_n). \tag{2.3}$$

We call this algorithm the LR algorithm. As we will show later, the unsupervised performance is almost identical to the linear regression. Noting the log operation in (2.3), and the fact that linear regression will generate $a < 0$, this is a different way of expressing (2.2). As

can be observed from Figure 2.1, both techniques generate estimates that correlate linearly with real ρ.

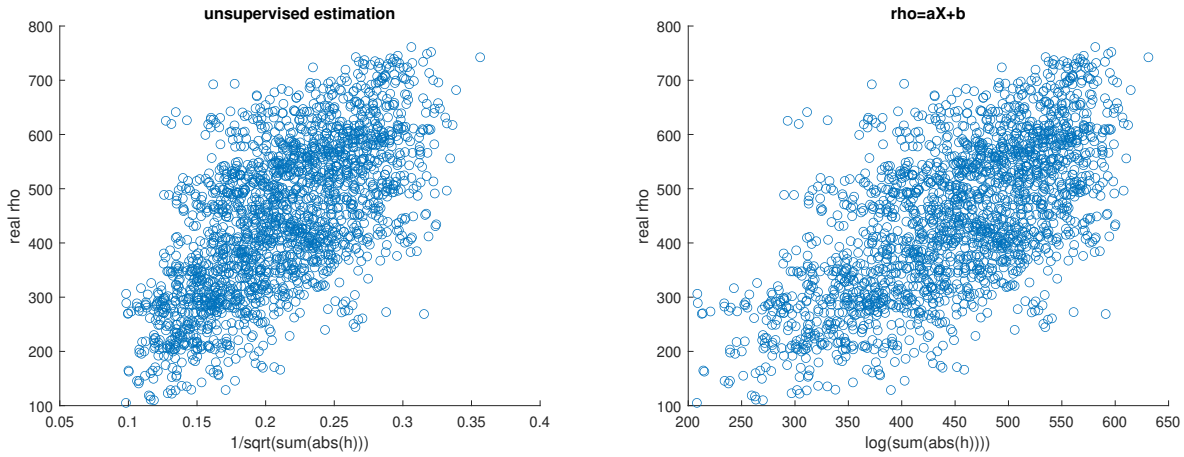

Figure 2.1: Correlation of real ρ vs estimated ρ.

2.2.3 Estimating ρ Using MUSIC

Here we use the same concept for estimating ρ as in estimating θ. The only difference is that we will use MUSIC to estimate the phase difference between subsequent subcarriers. As one can see from Figure 2.2, 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

$$\mathbf{B}(\rho) = (1, e^{-j2\pi\rho\Delta f/c}, e^{-j2\pi\rho 2\Delta f/c}, \ldots, e^{-j2\pi\rho(N_s-1)\Delta f/c})^T \tag{2.4}$$

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. We call the combination of using MUSIC to estimate θ and using MUSIC to estimate ρ the MUSIC/MUSIC (MM) algorithm.

In Chapter 4 we will present the performance of this algorithm separately than the others,

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 $B(\rho)$
    Calculate the PMF$(\rho) = \frac{1}{\text{Norm}_2(\mathbf{B}^H(\rho)\mathbf{N})}$
end for
Search the PMF for a peak and find the corresponding ρ

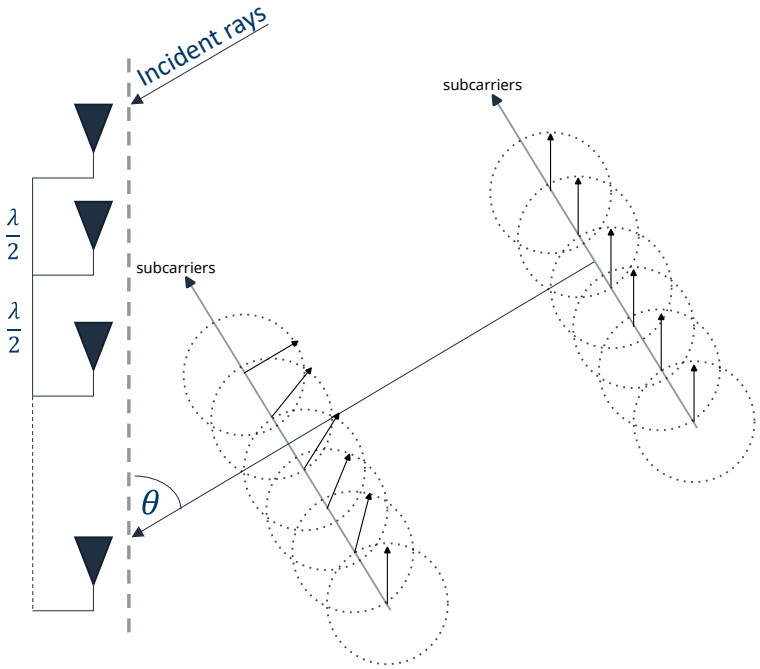

Figure 2.2: Phase change across subcarriers with distance.

as it requires multiple subcarriers. For this reason, we cannot compare fairly with the other algorithms, but we will show the performance with different number 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 similar algorithm was proposed in a different context [6].

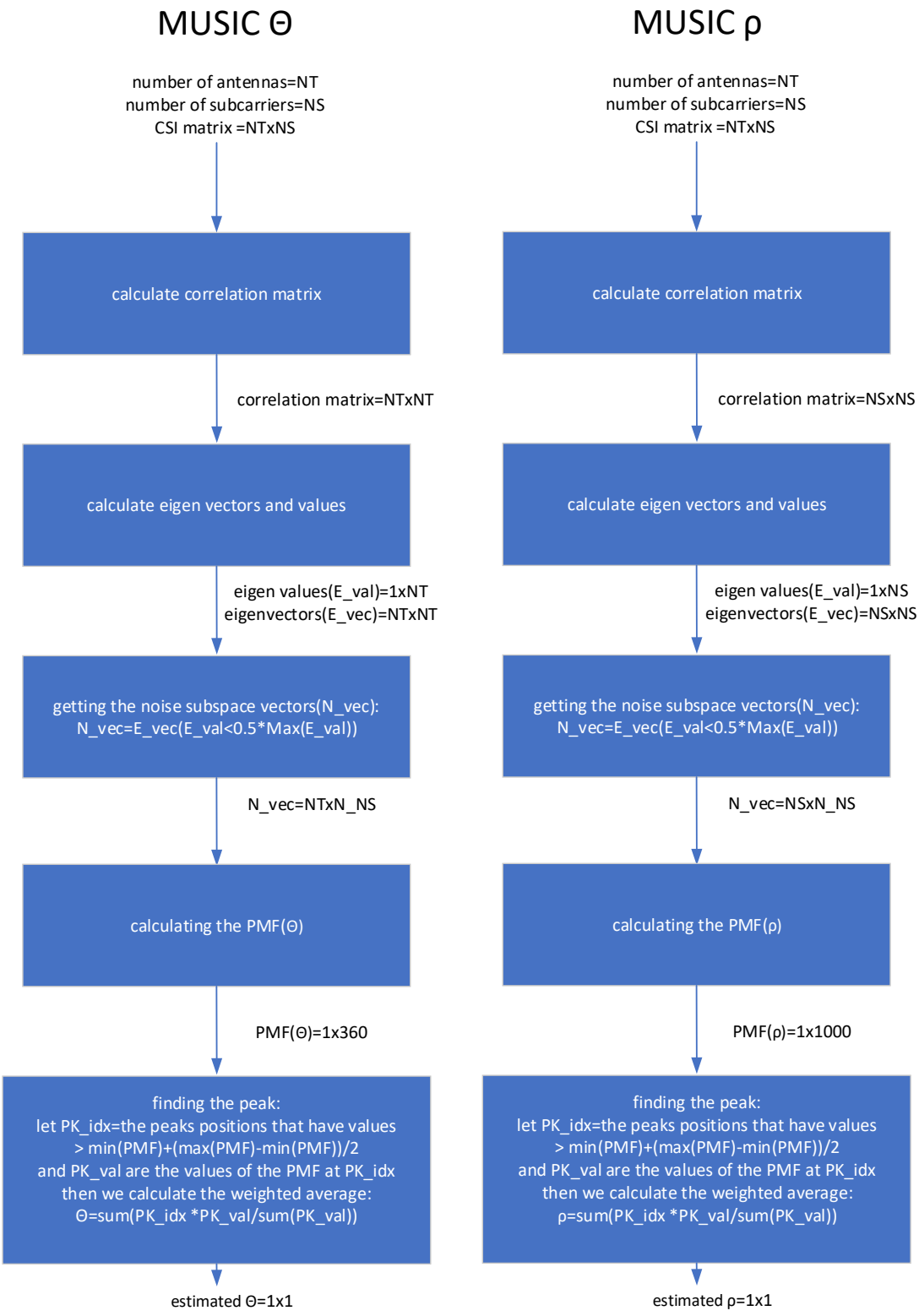

Figure 2.3: MUSIC algorithm to find θ and ρ. The combination is called the MM algorithm. Note NT is the number of antennas at the BS and NS is the number of subcarriers.

Chapter 3

Simulation Environment

In this thesis we reused and integrated our algorithm 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 1.1 at SNR = 0 dB. In the first scenario, we assume the antenna elements and the UEs are in the same plane as in Figure 3.1, and in the second scenario we use a three-dimensional environment exactly as in paper [1] as shown in Figure 3.2, where the antenna is 8.5 meters above the plane of the UEs. We call the first scenario 2D, and the second scenario 3D. We present three cases depending on the dimensions of the simulation environment which is 1000m x 500m. As in [1], some of the UEs are selected to make the word "VIP," so we can see if in the channel chart we preserve the shape. In the first case, as in leftmost subfigure of Figure 3.1 and Figure 3.2, the dimensions are the same as the dimensions in [1]. In the second case, as in the middle subfigure of Figure 3.1 and Figure 3.2, we are concerned with the performance of reducing this to 50% of the original dimensions. In the last case, as in the rightmost subfigure of Figure 3.1 and Figure 3.2, the dimensions are reduced to 25% of the original. We refer to our algorithms as ISQ (inverse square root sum), LR (linear regression), and MM (MUSIC/MUSIC). We integrated our algorithms inside the Matlab model of paper [1]. Figure 3.3 shows the flowchart of the

simulation model. We also integrated the latest Quadriga channel into the model. When we reproduced the results of paper [1], PCA, and SM were already inside the Matlab model of the paper. The auto encoder was written in Python so we integrated it as an executable Python environment, which we can call from Matlab as a system call.

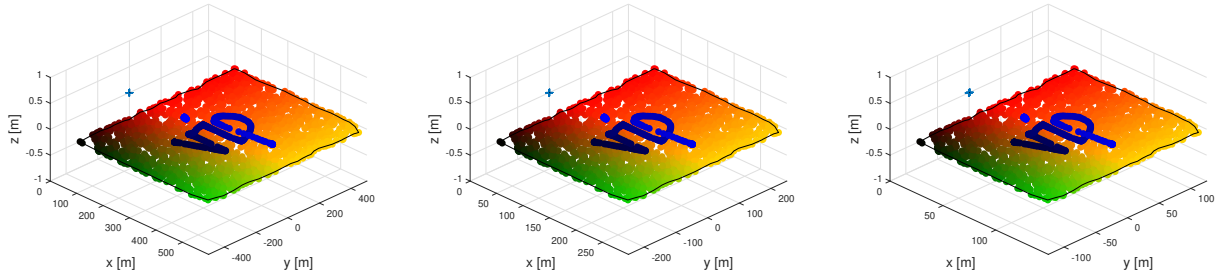

Figure 3.1: 2D environment for different environment dimensions: left: 100% scale, middle: 50% scale, right: 25% scale.

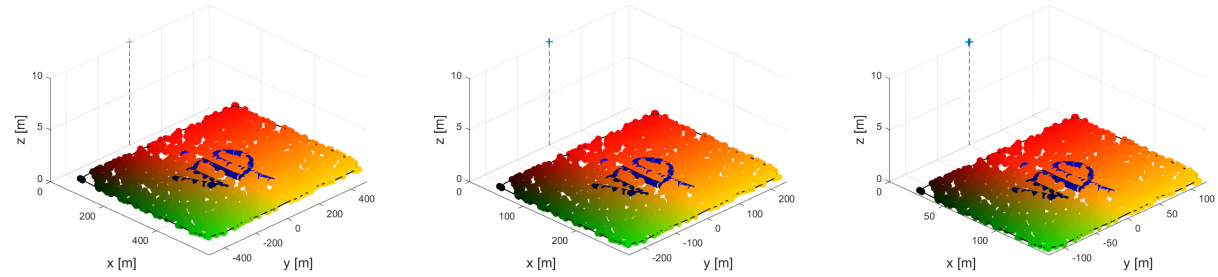

Figure 3.2: 3D environment for different environment dimensions: left: 100% scale, middle: 50% scale, right: 25% scale.

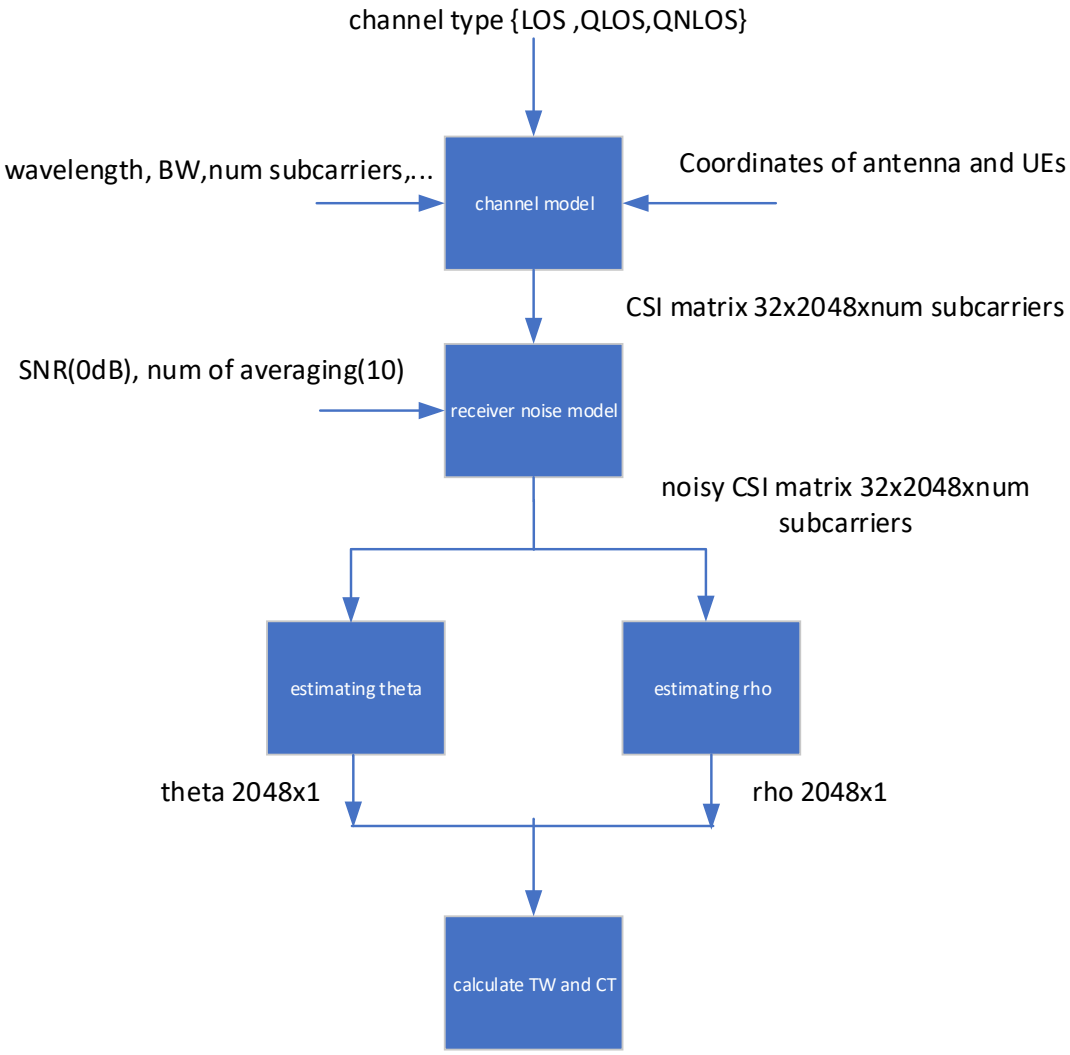

Figure 3.3: Simulation model flowchart.

Chapter 4

Performance Comparison and Complexity Analysis

We now compare the performance of our algorithms against the results in [1]. In doing so, we compare the performance under three channels used in [1]. These channels are Vanilla LOS (LOS), Quadriga LOS (QLOS), and Quadriga NLOS (QNLOS). For each channel, we compare three scenarios described in Chapter 2. The results are summarized in Table 4.1, 4.2, 4.3, and 4.4 for nearest-k neighbors equal to 102, as in most of the channel charting papers. We use CT and TW as performance measures to compare different algorithms. We will also discuss channel charts as well as CT and TW performance as a function of k-nearest neighbors. We will discuss LR and ISQ separately than MM, because MM uses more than one subcarrier, and is more complex.

4.1 LR and ISQ Performance Comparison

We will now compare the performance of our algorithm to the three algorithms PCA, SM, and AE from [1] in terms of TW and CT. These are given in Table 4.1 for the 2D channel and in Table 4.2 for the 3D channel. The value of k-nearest neighbors is 102 in both cases. We note that the two sets of tables have very similar entries. Therefore, our conclusions below apply to 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 (supervised) is only slightly better than ISQ or slightly worse than ISQ. The difference is less than 1%. This is negligible compared to the overhead of using GPS and relaying this information to the BS. We note that the reason we look at less than 100% area scale in Table 4.1 and Table 4.2 as well as in the sequel is the possibility of getting sufficiently good performance with smaller distances.

Fig. 4.1 and Fig. 4.2 present the channel charts for the 2D channel. Fig. 4.1 presents the case for 100% of the original dimensions and Fig. 4.2 presents the case for 25% of the original dimensions. In both figures, columns 1 through 5 correspond to PCA, SM, AE, LR, and ISQ algorithms. 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 [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. 4.1 and Fig. 4.2 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 [1, Fig. 1(a)] can be seen to be much more preserved with LR and ISQ algorithms. LR and ISQ channel charts are similar with a very slight preference towards LR.

TW and CT performance curves for the 2D channel are given in Figure 4.3 for the 100% of original dimensions, in Figure 4.4 for the 50% of original dimensions, and in Figure 4.5

for the 25% of the original dimensions against k nearest neighbors. The blue curves are for CT, while the red curves are for TW. ISQ results are given by dashed curves without any symbols, while LR curves are given by dashed curves with a star symbol. It can be observed that while the performance of LR and ISQ are close, they consistently beat PCA, SM, and AE results.

Channel charts for the 3D channel are given in Figure 4.6 and Figure 4.7 for the 100% and the 25% of the original dimensions. TW and CT performance curves for the 3D channel are given in Figure 4.8 for the 100% of original dimensions, in Figure 4.9 for the 50% of original dimensions, and in Figure 4.10 for the 25% of the original dimensions against k nearest neighbors. Although numerical values show difference, conclusions for these 3D channel results are the same as the 2D channel results presented above.

Table 4.1: Performance comparison for TW and CT at k-nearest = 102 for LR and ISQ algorithms in 2D channel.

metric/area	scale/chan		PCA	SM	AE	LR	ISQ
TW	100%	LOS	0.8565	0.7986	0.8303	0.9929	0.9887
		QLOS	0.8447	0.8359	0.8655	0.9119	0.9125
		QNLOS	0.8500	0.8459	0.8533	0.9090	0.9095
	50%	LOS	0.8596	0.8200	0.8349	0.9932	0.9889
		QLOS	0.8579	0.8343	0.8510	0.9056	0.9050
		QNLOS	0.8682	0.8661	0.8747	0.9376	0.9386
	25%	LOS	0.8570	0.7616	0.8118	0.9930	0.9886
		QLOS	0.8630	0.8724	0.8667	0.9494	0.9489
		QNLOS	0.8842	0.8885	0.8997	0.9544	0.9545
CT	100%	LOS	0.9270	0.8749	0.8921	0.9967	0.9941
		QLOS	0.9219	0.9007	0.8718	0.9445	0.9360
		QNLOS	0.9250	0.9242	0.9158	0.9339	0.9315
	50%	LOS	0.9279	0.9004	0.8915	0.9969	0.9941
		QLOS	0.9343	0.9047	0.9296	0.9604	0.9504
		QNLOS	0.9331	0.9296	0.9210	0.9516	0.9529
	25%	LOS	0.9271	0.8432	0.8616	0.9968	0.9940
		QLOS	0.9348	0.9390	0.9297	0.9699	0.9688
		QNLOS	0.9360	0.9377	0.9325	0.9593	0.9602

Table 4.2: Performance comparison for TW and CT at k-nearest = 102 for LR and ISQ algorithms in 3D channel.

metric/area scale/chan			PCA	SM	AE	LR	ISQ
TW	100%	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
	50%	LOS	0.8568	0.8370	0.8242	0.9930	0.9888
		QLOS	0.8577	0.8378	0.8503	0.9066	0.9065
		QNLOS	0.8698	0.8643	0.8661	0.9357	0.9365
	25%	LOS	0.8563	0.8167	0.8291	0.9932	0.9889
		QLOS	0.8637	0.8698	0.8677	0.9487	0.9483
		QNLOS	0.8794	0.8853	0.8975	0.9583	0.9582
CT	100%	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
	50%	LOS	0.9273	0.9183	0.8801	0.9968	0.9941
		QLOS	0.9333	0.9150	0.9210	0.9620	0.9517
		QNLOS	0.9327	0.9308	0.9146	0.9535	0.9546
	25%	LOS	0.9264	0.9132	0.8946	0.9969	0.9941
		QLOS	0.9334	0.9370	0.9279	0.9672	0.9677
		QNLOS	0.9337	0.9378	0.9361	0.9646	0.9655

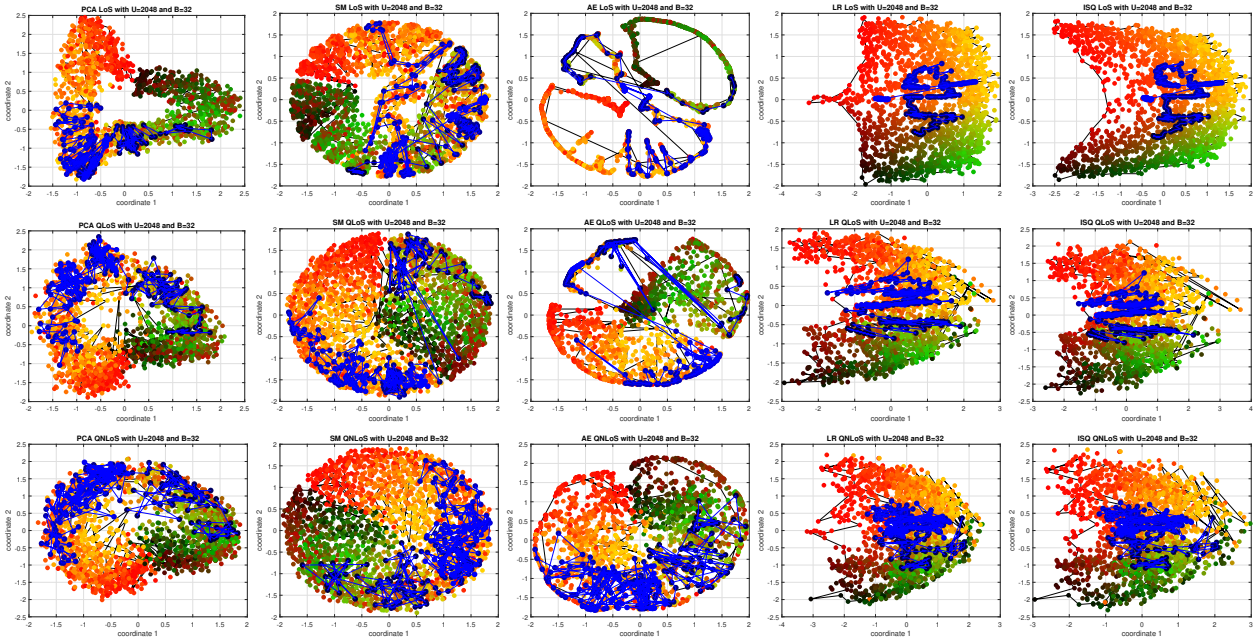

Figure 4.1: Channel charts with PSA, SM, AE, LR, and ISQ algorithms for the 2D LOS, QLOS, and QNLOS channels, original dimensions.

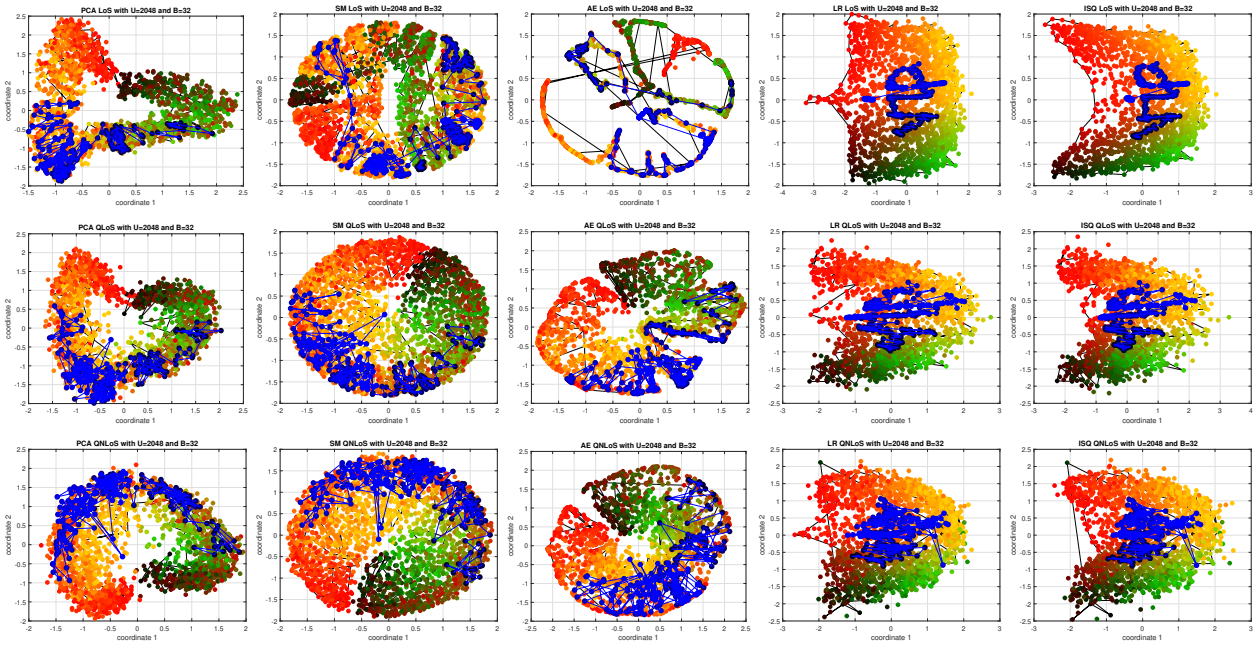

Figure 4.2: Channel charts with PSA, SM, AE, LR, and ISQ algorithms for the 2D LOS, QLOS, and QNLOS channels, 25% of original dimensions.

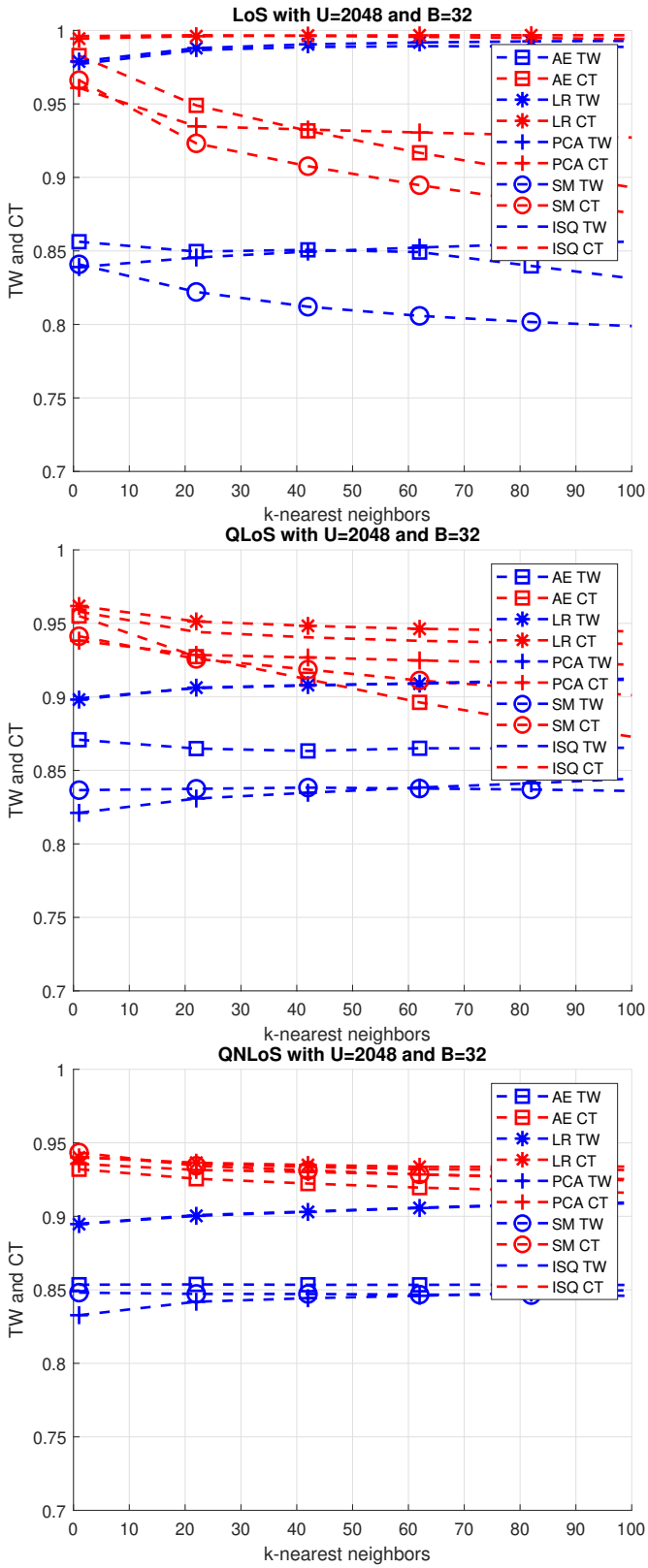

Figure 4.3: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 2D channel, original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

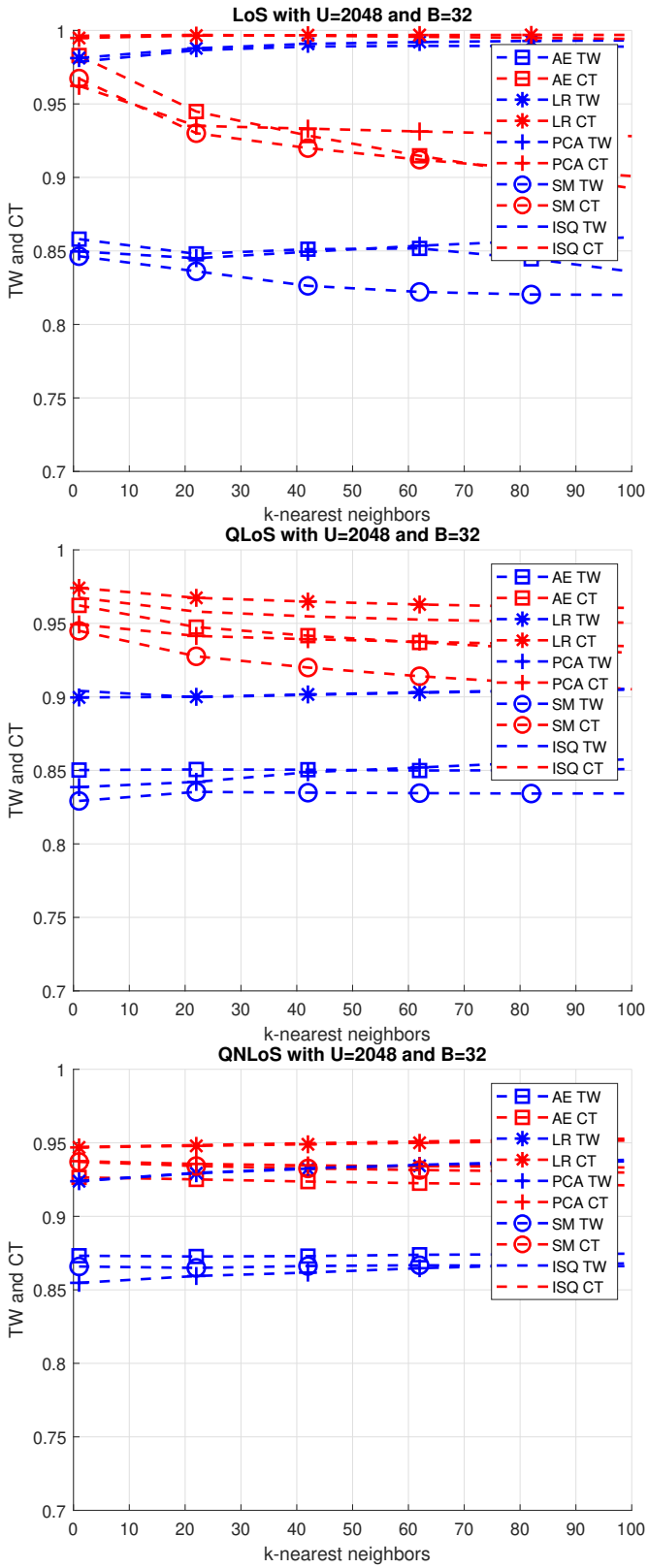

Figure 4.4: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 2D channel, 50% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

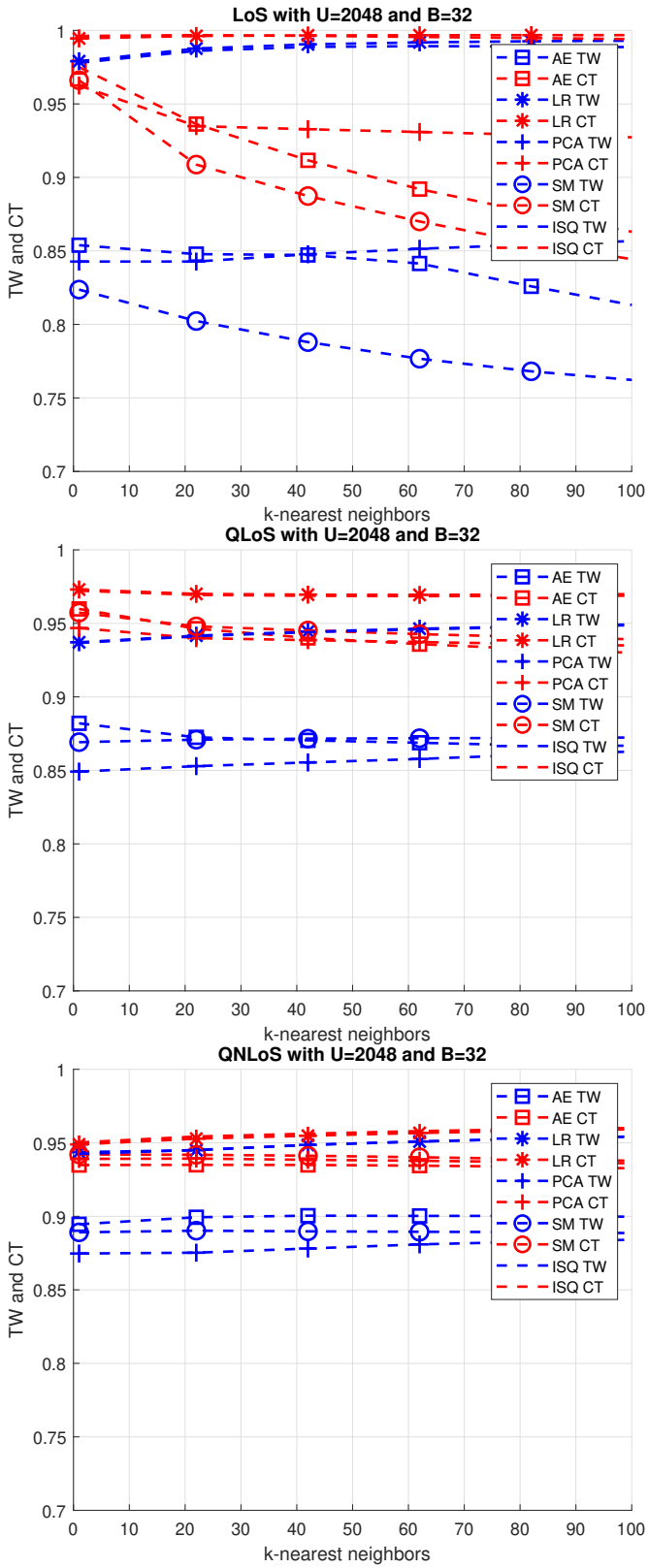

Figure 4.5: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 2D channel, 25% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

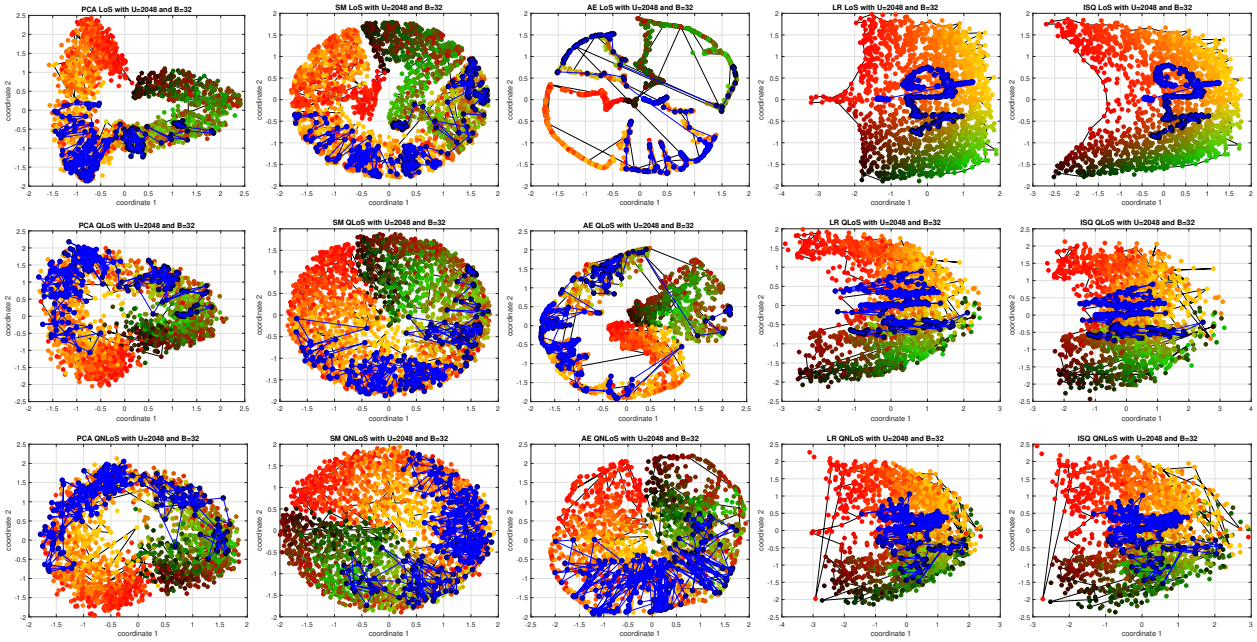

Figure 4.6: Channel charts with PSA, SM, AE, LR, and ISQ algorithms for the 3D LOS, QLOS, and QNLOS channels, original dimensions.

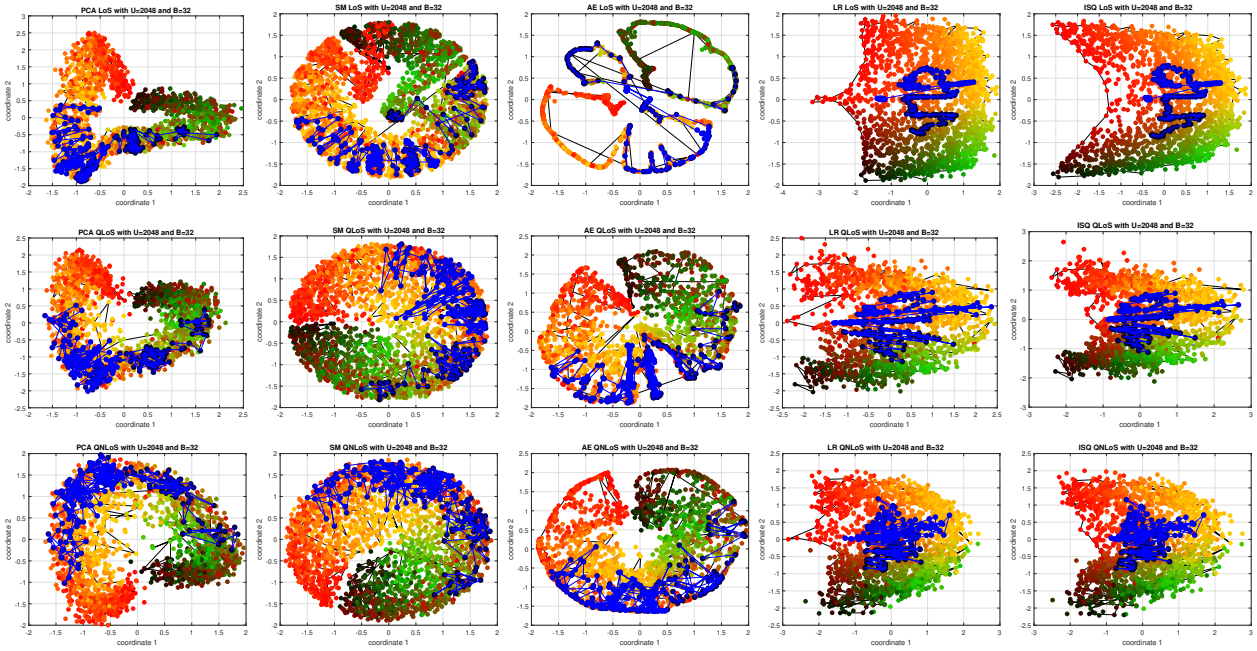

Figure 4.7: Channel charts with PSA, SM, AE, LR, and ISQ algorithms for the 3D LOS, QLOS, and QNLOS channels, 25% of original dimensions.

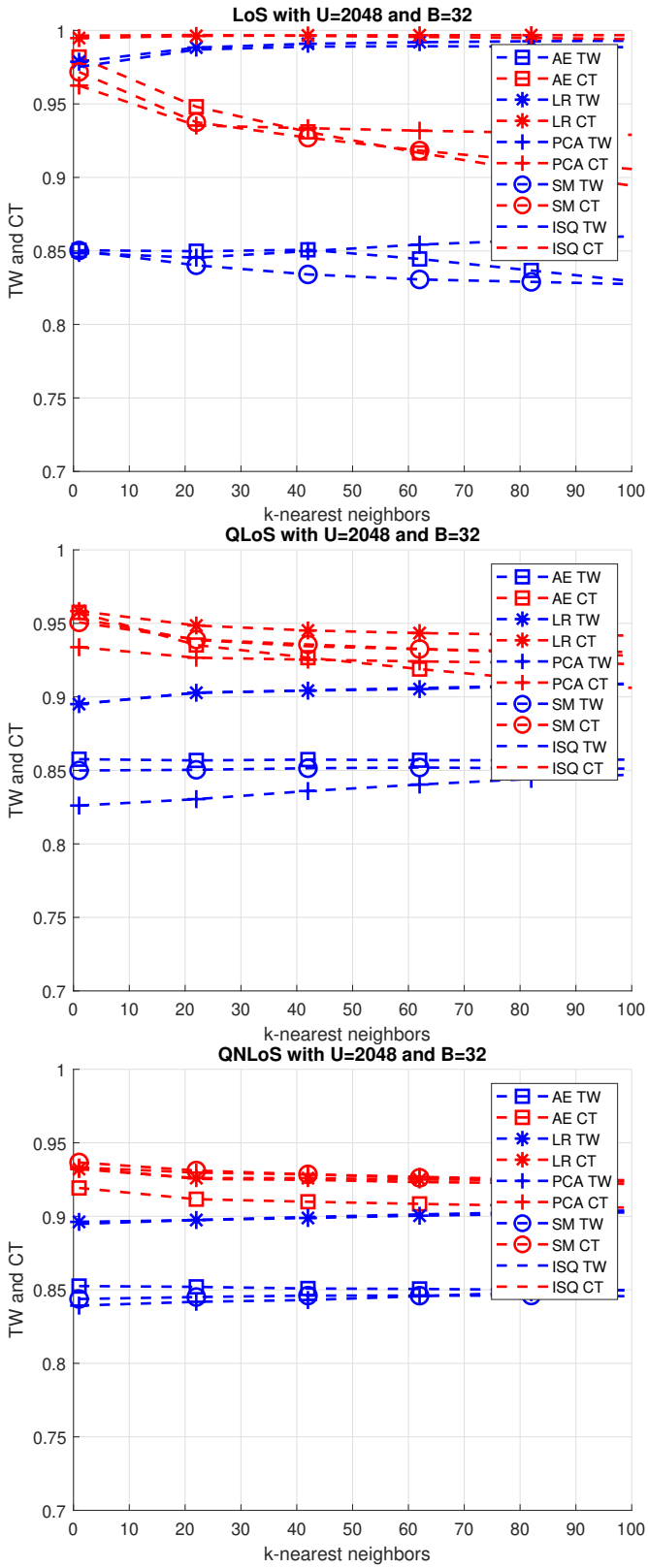

Figure 4.8: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 3D, original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

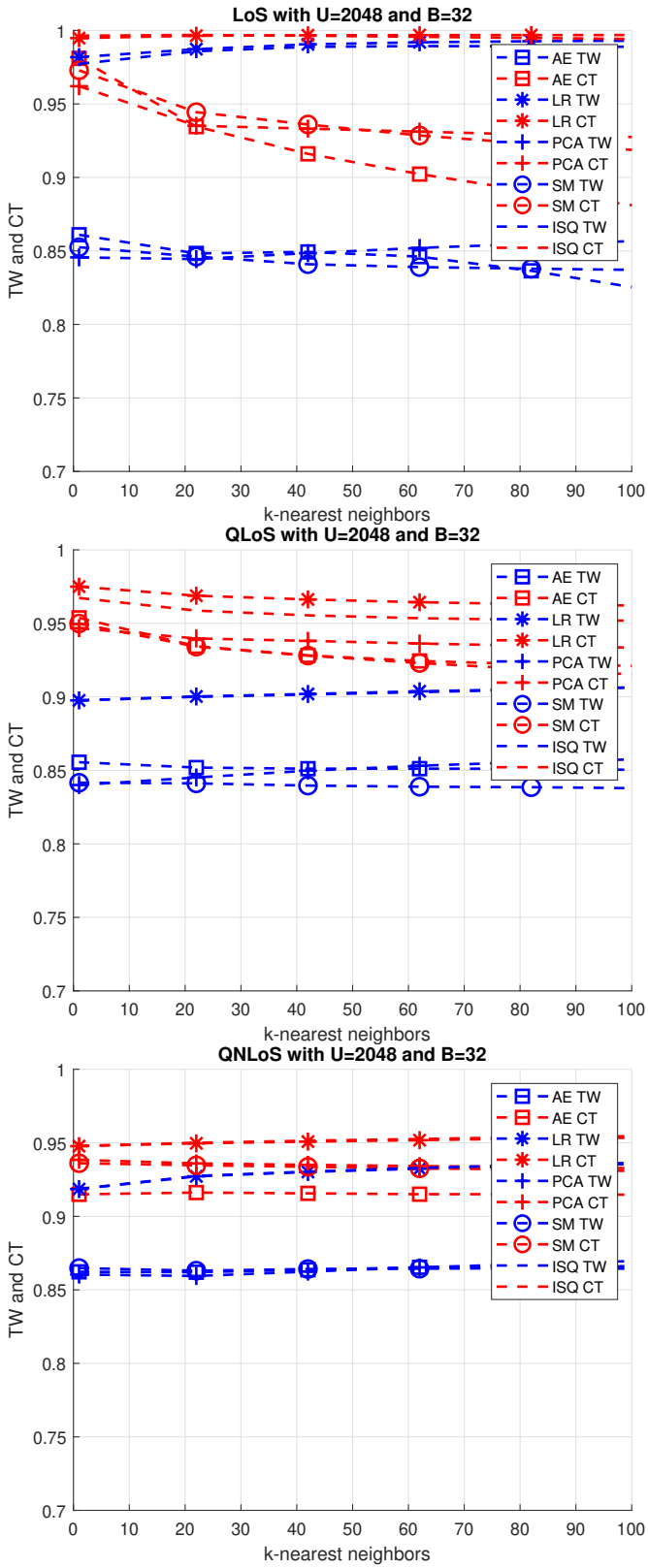

Figure 4.9: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 3D channel, 50% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

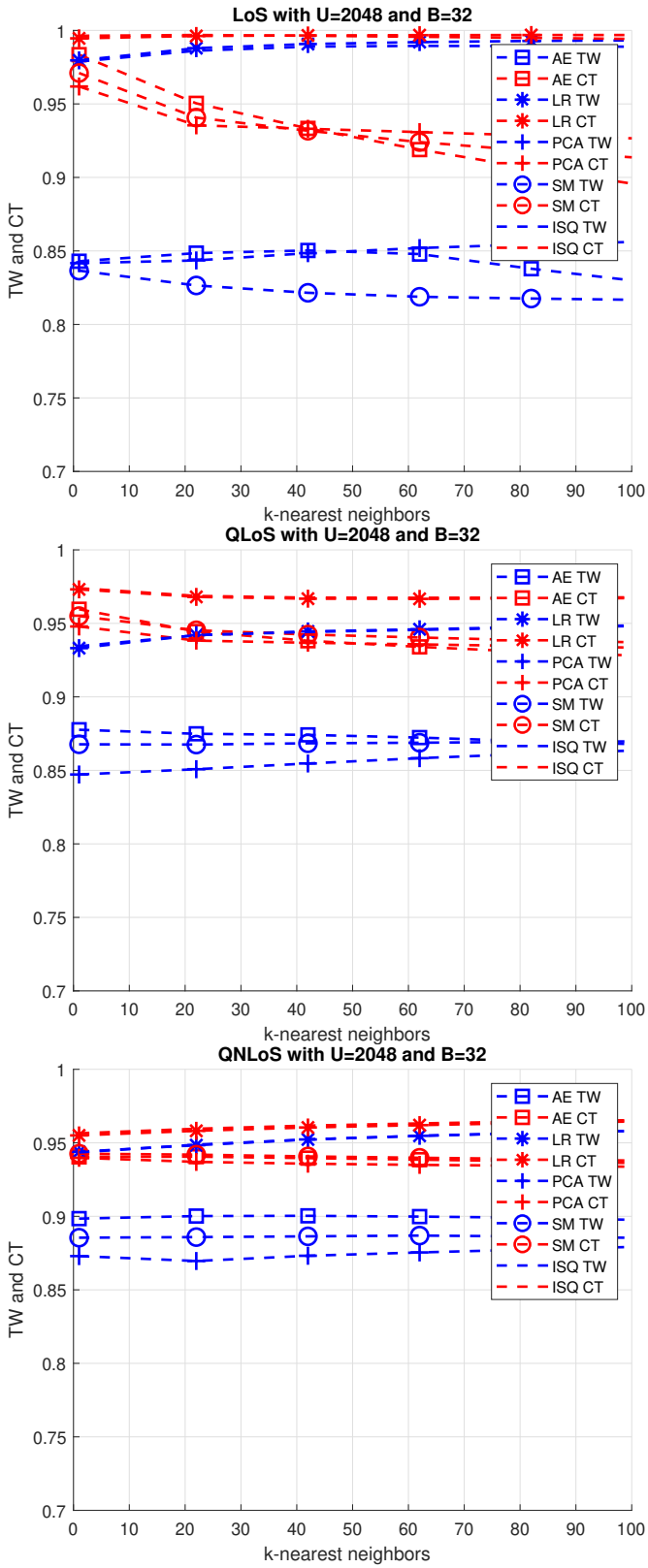

Figure 4.10: TW and CT performance against k-nearest neighbors for LR and ISQ algorithms in 3D channel, 25% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

4.2 MM Performance

We will now discuss the performance of the MM algorithm. We will discuss the 2D channel and the 3D channel separately. For the 2D channel, Table 4.3 presents the TW and CT results for k-nearest neighbors equal to 102, for the cases of 100%, 50%, and 25% of original dimensions. Then, Figure 4.11 and Figure 4.12 present the channel charts for 100% and 25% of original dimensions, respectively for the 2D-channel. Finally, Figure 4.13, Figure 4.14, and Figure 4.15 present TW and CT performance against k-nearest neighbors for 100%, 50%, and 25% respectively, for the 2D channel.

In a similar way, we have 3D channel results. Table 4.4 presents the TW and CT results for k-nearest neighbors equal to 102, for the cases of 100%, 50%, and 25% of original dimensions. Then, Figure 4.16 and Figure 4.17 present the channel charts for 100% and 25% of original dimensions, respectively. Finally, Figure 4.18, Figure 4.19, and Figure 4.20 present TW and CT performance against k-nearest neighbors for 100%, 50%, and 25% respectively.

In all cases, there are no comparisons made with the algorithms previously discussed, i.e., PSA, SM, AE, LR, and ISQ. The comparisons are made with different number of subcarriers, i.e., 2, 8, 20, and 32. The performance of this algorithm with even a small number of subcarriers is very good. It improves significantly by increasing the number of subcarriers. It can be observed by comparing with the LR and ISQ results provided earlier, the algorithm outperforms not only PSA, SM, and AE, but also our algorithms LR and ISQ. The performance with LOS is excellent, reducing somewhat with QLOS, and reducing more with QNLOS. However, even with QNLOS, we still get very good performance as compared with the cases for PSA, SM, AE, LR, and ISQ.

Table 4.3: Performance comparison for TW and CT at k-nearest = 102 for MM algorithm in 2D channel.

metric/area scale/chan			2sc	8sc	14sc	20sc	26sc	32sc
TW	100%	LOS	0.9975	0.9986	0.9992	0.9992	0.9996	0.9996
		QLOS	0.9819	0.9949	0.9966	0.9971	0.9978	0.9980
		QNLOS	0.9616	0.9817	0.9839	0.9848	0.9848	0.9849
	50%	LOS	0.9953	0.9986	0.9991	0.9993	0.9994	0.9995
		QLOS	0.9688	0.9927	0.9945	0.9955	0.9970	0.9972
		QNLOS	0.9418	0.9657	0.9725	0.9736	0.9764	0.9769
	25%	LOS	0.9862	0.9984	0.9991	0.9994	0.9996	0.9996
		QLOS	0.9526	0.9892	0.9949	0.9962	0.9969	0.9974
		QNLOS	0.9031	0.9371	0.9414	0.9443	0.9486	0.9511
CT	100%	LOS	0.9991	0.9999	1.0000	0.9999	0.9999	0.9999
		QLOS	0.9825	0.9961	0.9977	0.9986	0.9993	0.9994
		QNLOS	0.9598	0.9819	0.9851	0.9858	0.9858	0.9864
	50%	LOS	0.9968	0.9997	0.9999	0.9999	1.0000	0.9999
		QLOS	0.9711	0.9942	0.9956	0.9968	0.9981	0.9982
		QNLOS	0.9328	0.9629	0.9697	0.9716	0.9743	0.9743
	25%	LOS	0.9862	0.9995	0.9996	0.9995	0.9995	0.9995
		QLOS	0.9450	0.9858	0.9932	0.9958	0.9964	0.9971
		QNLOS	0.9086	0.9252	0.9269	0.9304	0.9333	0.9346

Table 4.4: Performance comparison for TW and CT at k-nearest = 102 for MM algorithm in 3D channel.

metric/area	scale/chan		2sc	8sc	14sc	20sc	26sc	32sc
TW	100%	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
	50%	LOS	0.9952	0.9985	0.9990	0.9993	0.9997	0.9996
		QLOS	0.9696	0.9927	0.9949	0.9954	0.9966	0.9977
		QNLOS	0.9474	0.9674	0.9734	0.9756	0.9769	0.9784
	25%	LOS	0.9863	0.9983	0.9989	0.9992	0.9997	0.9999
		QLOS	0.9524	0.9892	0.9951	0.9965	0.9972	0.9977
		QNLOS	0.9062	0.9379	0.9431	0.9465	0.9515	0.9546
CT	100%	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
	50%	LOS	0.9967	0.9998	0.9998	0.9999	0.9999	0.9999
		QLOS	0.9717	0.9947	0.9964	0.9968	0.9978	0.9989
		QNLOS	0.9419	0.9636	0.9709	0.9723	0.9739	0.9760
	25%	LOS	0.9863	0.9994	0.9996	0.9995	0.9996	0.9996
		QLOS	0.9448	0.9873	0.9939	0.9956	0.9970	0.9977
		QNLOS	0.9086	0.9243	0.9291	0.9286	0.9344	0.9370

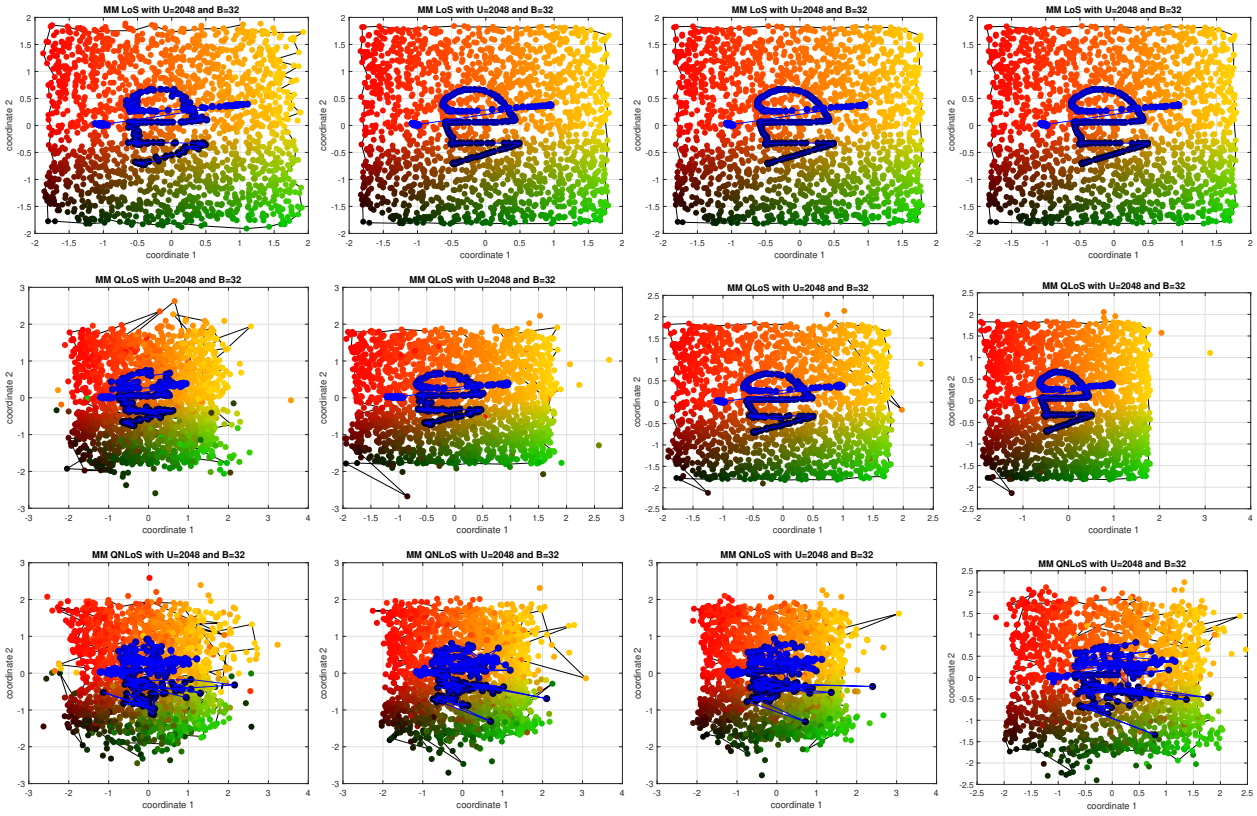

Figure 4.11: Channel charts with the MM algorithm for the 2D LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers, original dimensions.

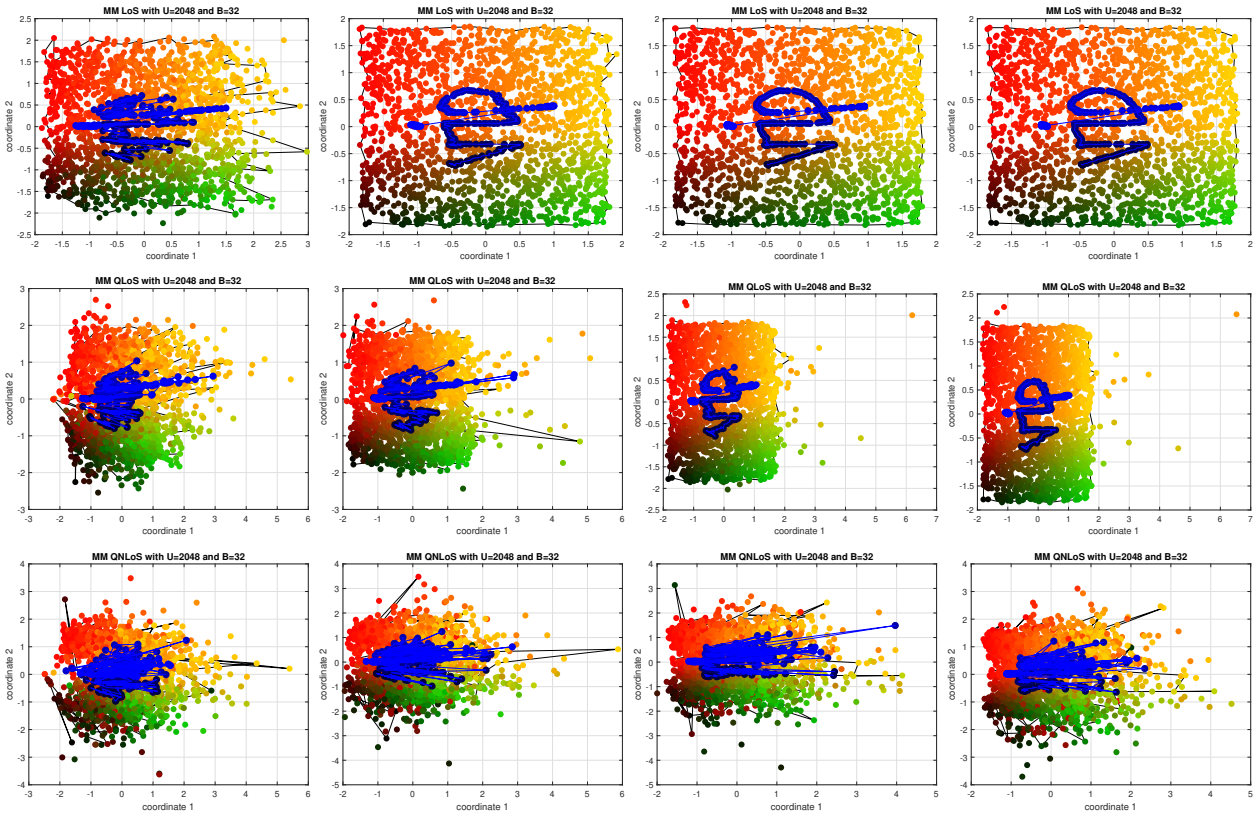

Figure 4.12: Channel charts with the MM algorithm for the 2D LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers, 25% original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

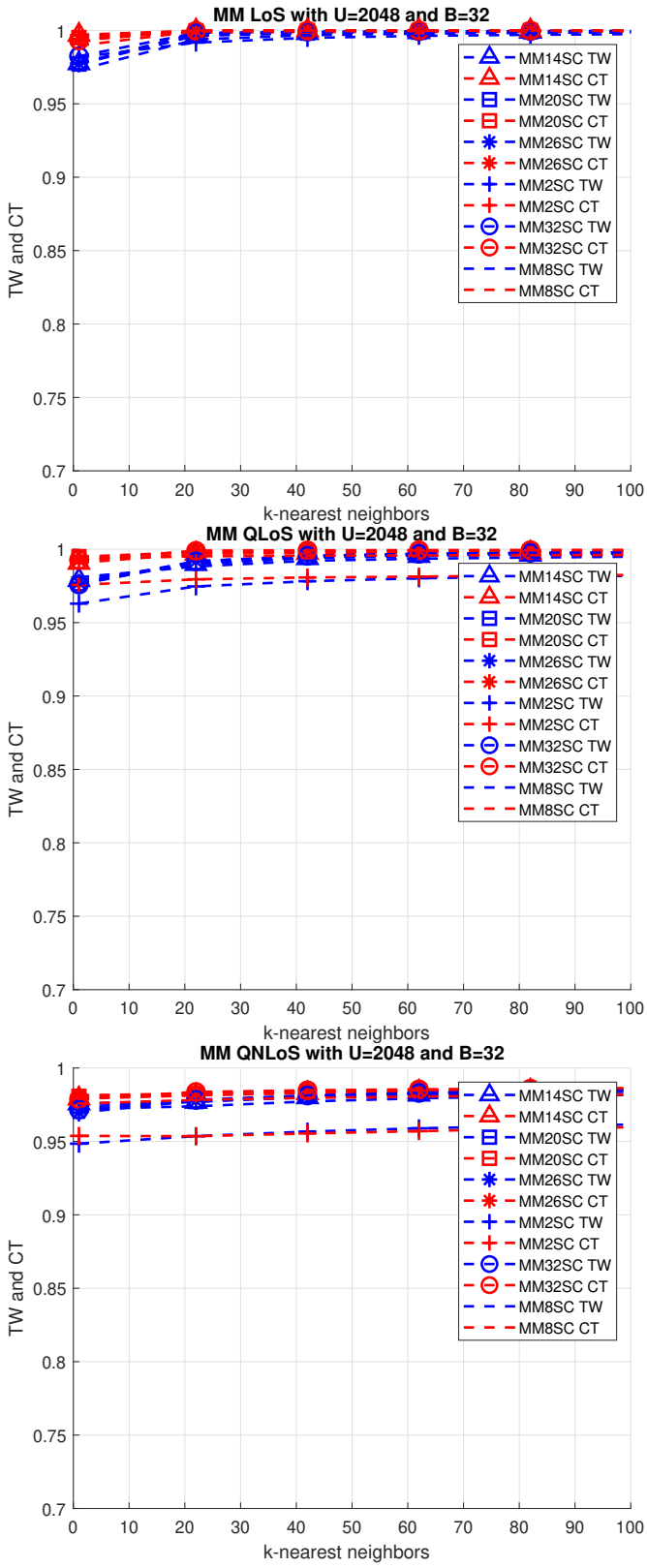

Figure 4.13: TW and CT performance against k-nearest neighbors for MM algorithm in 2D channel, original dimensions.

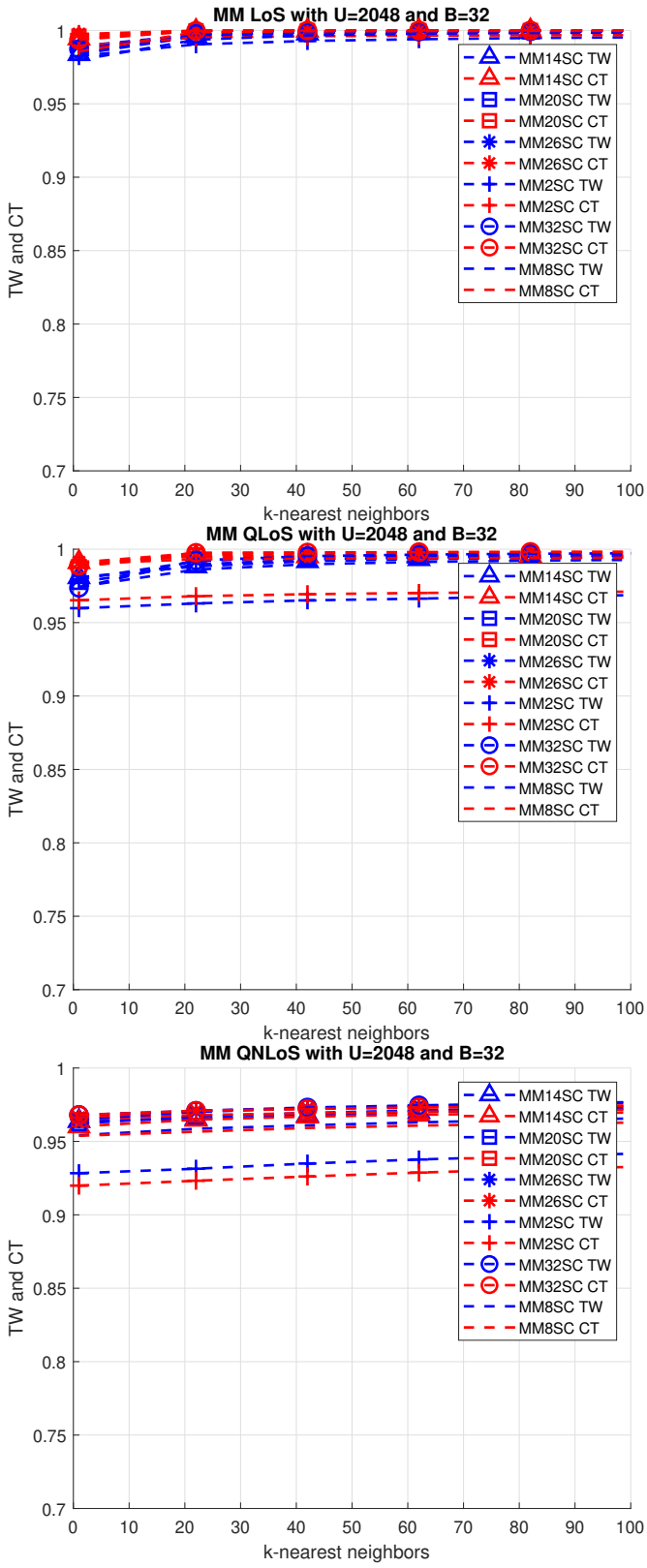

Figure 4.14: TW and CT performance against k-nearest neighbors for MM algorithm in 2D channel, 50% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

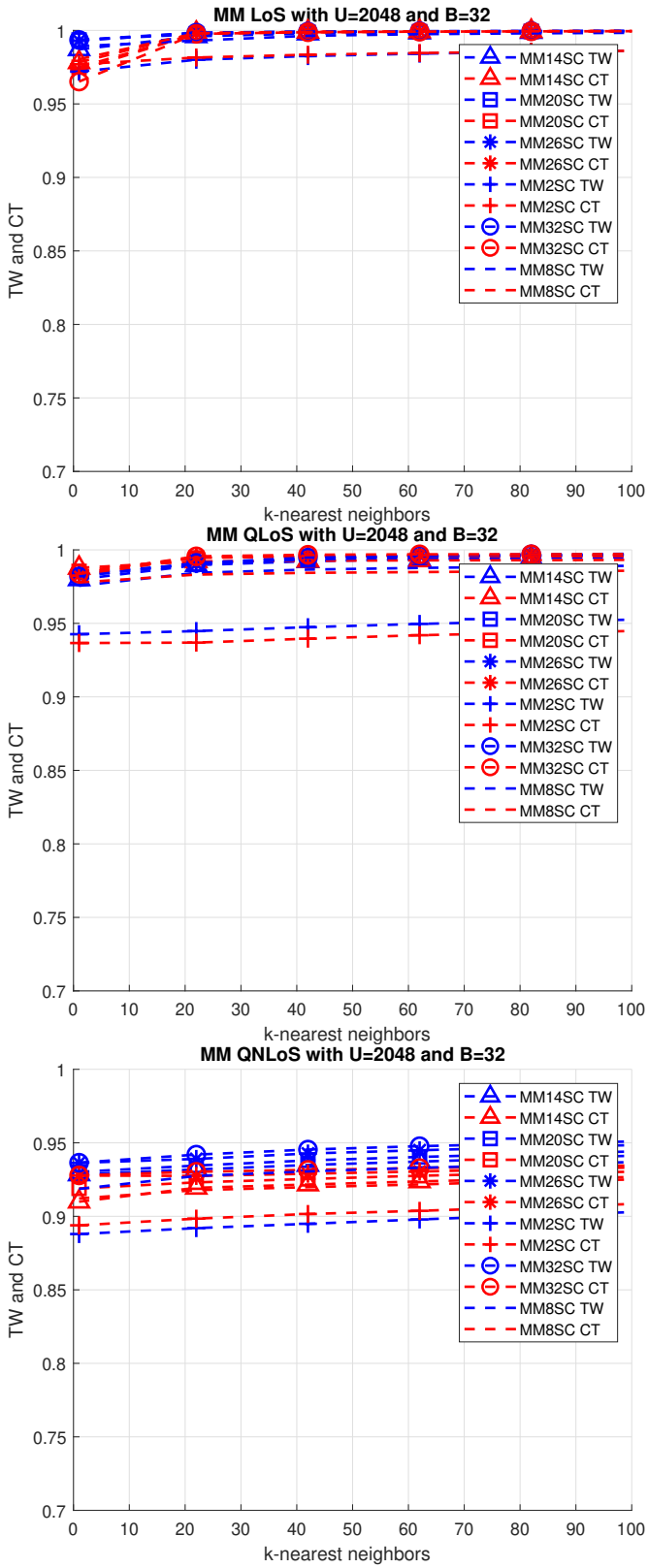

Figure 4.15: TW and CT performance against k-nearest neighbors for MM algorithm in 2D channel, 25% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

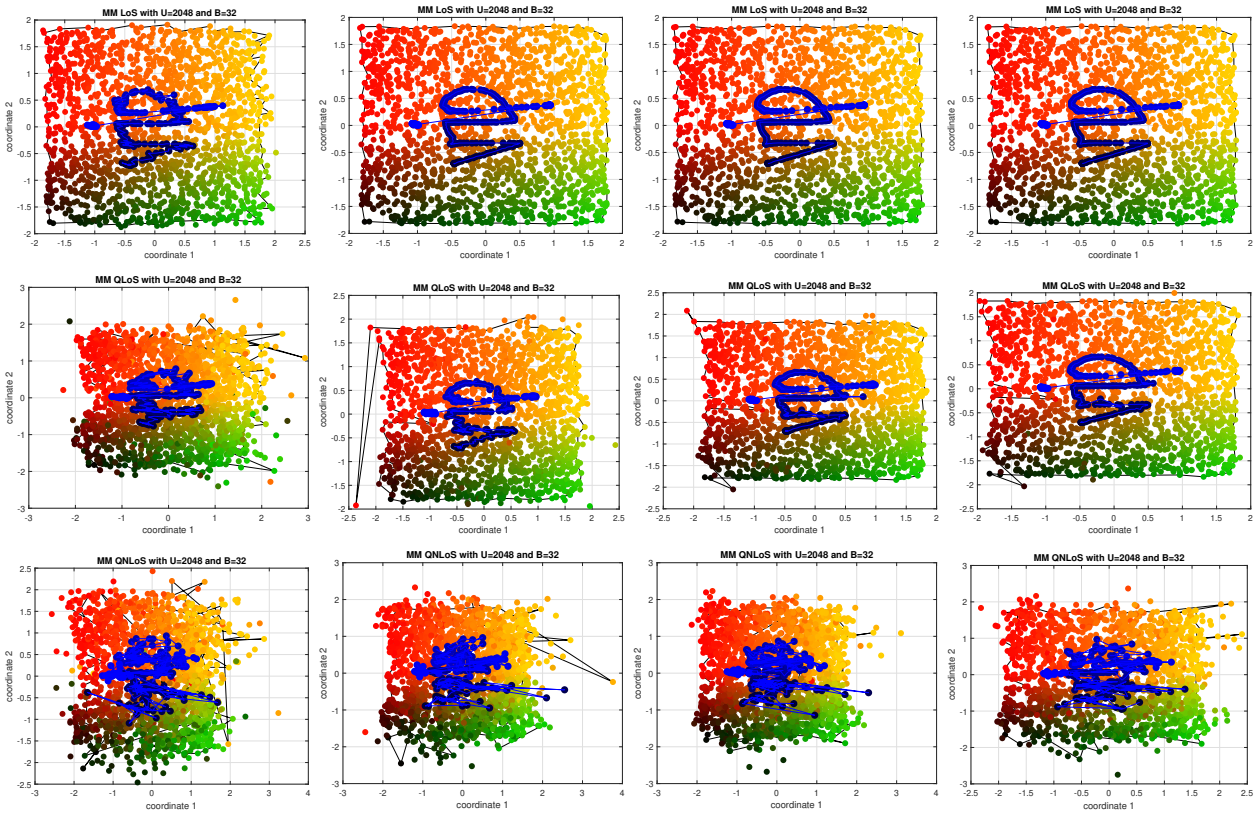

Figure 4.16: Channel charts with the MM algorithm for the 3D LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers, original dimensions.

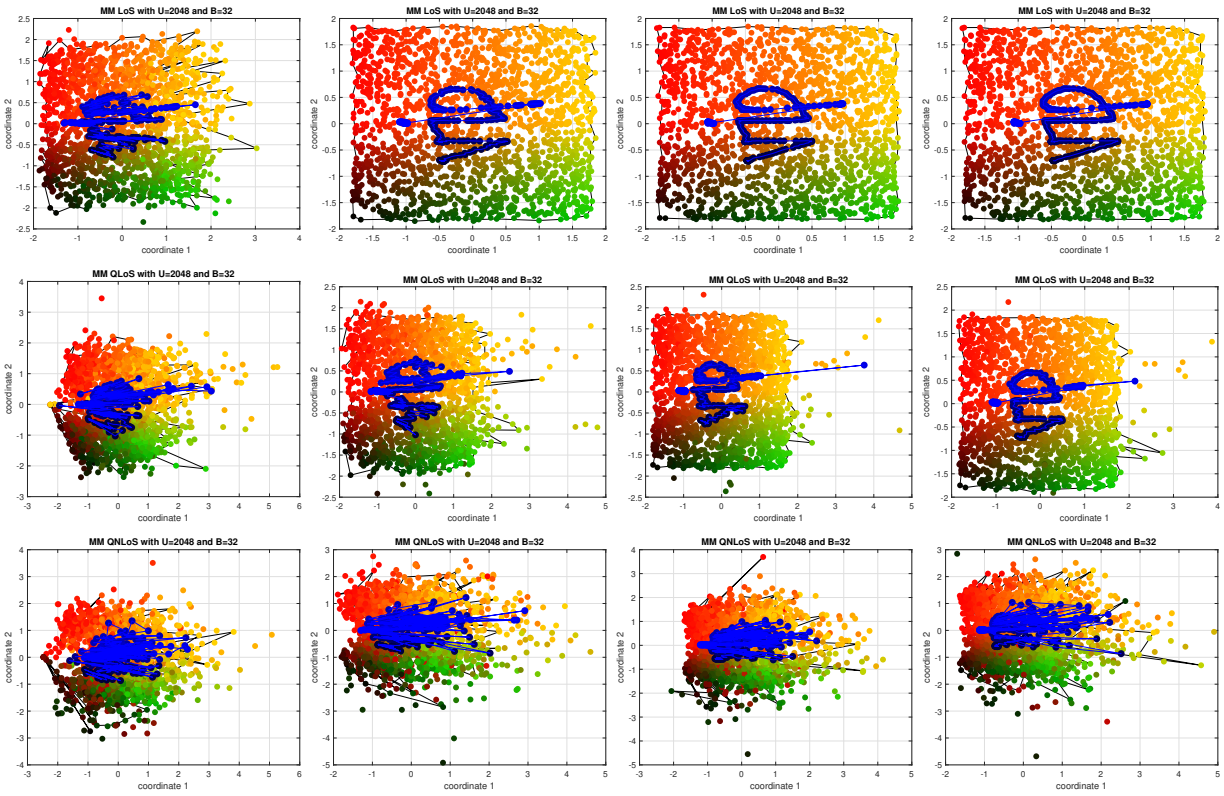

Figure 4.17: Channel charts with the MM algorithm for the 3D LOS, QLOS, and QNLOS channels at 2, 8, 20, and 32 subcarriers, 25% of original dimensions.

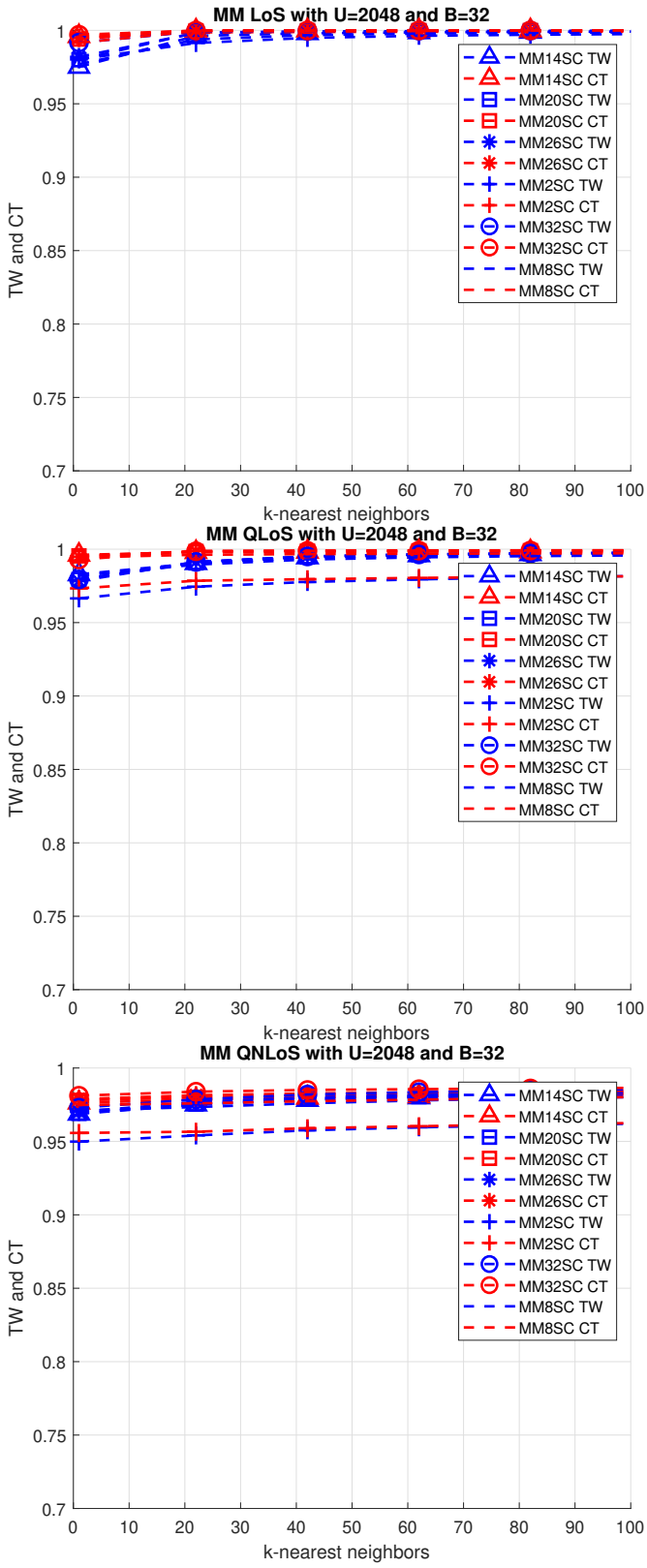

Figure 4.18: TW and CT performance against k-nearest neighbors for MM algorithm in 3D channel, original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

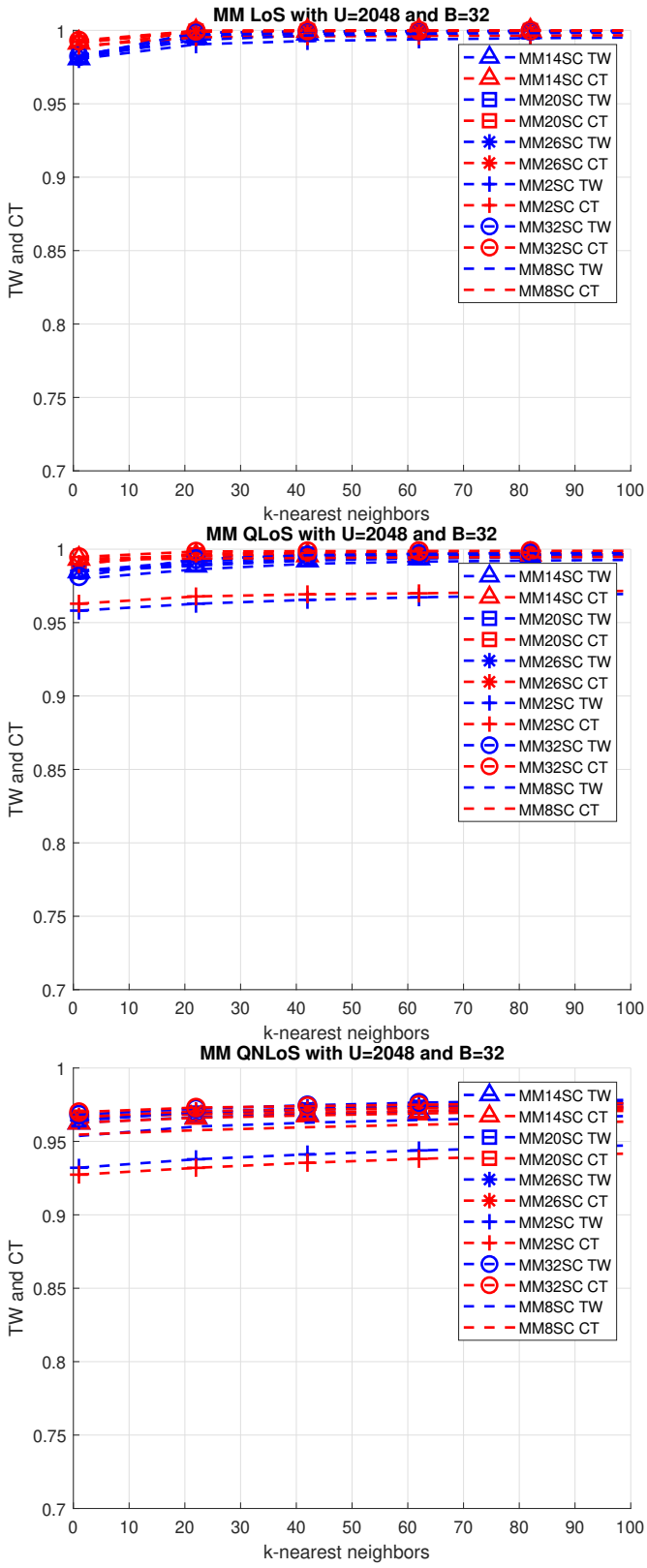

Figure 4.19: TW and CT performance against k-nearest neighbors for MM algorithm in 2D channel, 50% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

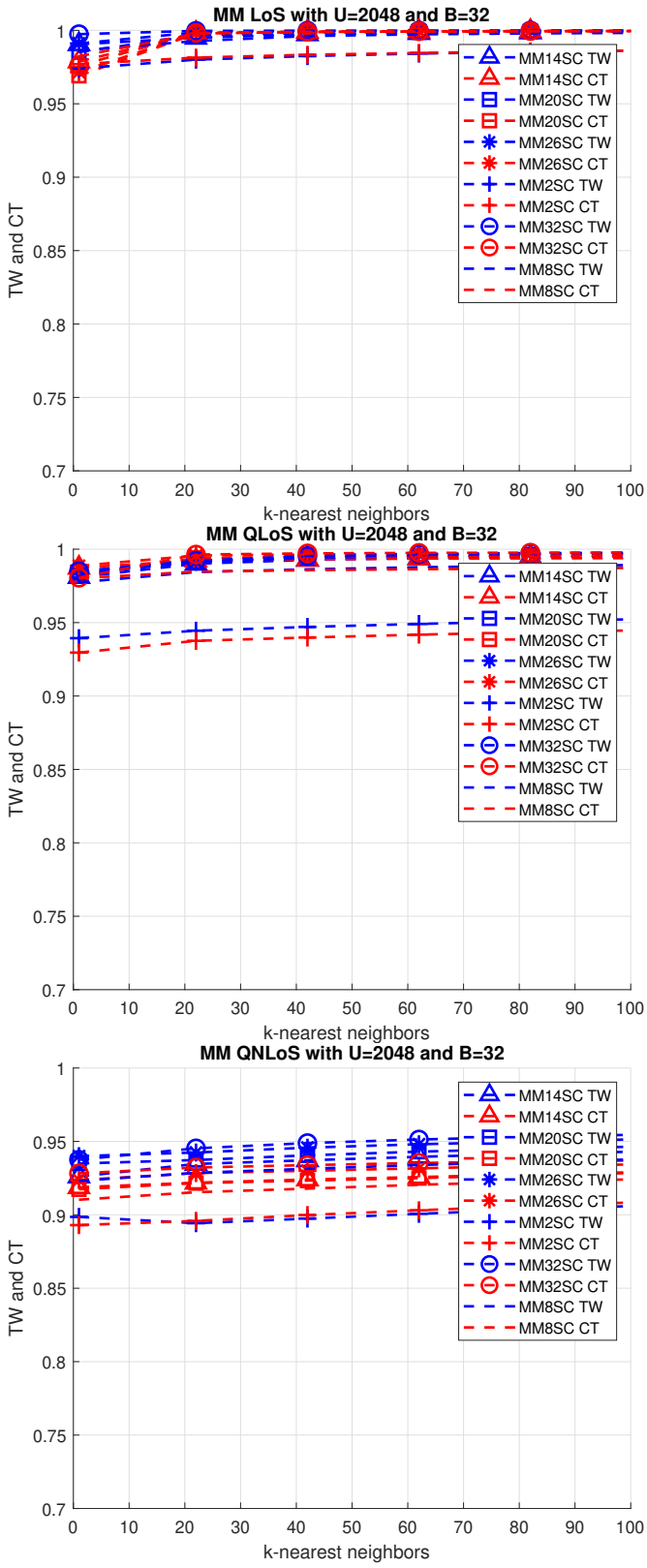

Figure 4.20: TW and CT performance against k-nearest neighbors for MM algorithm in 3D channel, 25% of original dimensions. Top: LOS, middle: QLOS, bottom: QNLOS.

4.3 Complexity Comparison

When we compare the complexity of our algorithms LR, 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 is 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.

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.

The computational complexity per UE in the calculation of θ for LR and ISQ is mostly due to the MUSIC algorithm complexity, which consists of calculating the covariance matrix, the decomposition of the eigenvalues, thresholding calculation of the PMF of θ, and the peak search. As far as the calculation of ρ is concerned, LR has matrix multiplication due to linear regression. For ISQ we need to calculate the magnitude of CSI, which is complex vector magnitude calculation, summation of these magnitudes, and then calculation of the inverse square root. The MM algorithm uses MUSIC for both θ and ρ estimation. Since they are independent of each other, they can be calculated separately. As previously discussed, for MM we use MUSIC for both θ and ρ. Moreover, we are using multiple subcarriers for both θ and ρ.

As an indication of the complexity, we will compare the simulation time for producing

Table 4.5: Comparison of simulation times.

Algorithm	Simulation time (seconds)
PCA	0.817
SM	12.163
AE	53.893
LR	7.147
ISQ	7.094
MM	20.448

channel chart for 2048 UEs for different algorithms. This is provided in Table 4.5 below. The simulation time is not dependent on the channnel, the scale, or the geometry of the environment. The simulation time only relies on the number UEs, number of antennas, and number of subcarriers. In Table 4.5, number of UEs is 2048, number of BS antennas is 32, and number of subcarriers is 32.

The basis of the simulation program is the Matlab code released by the authors of [1]. In this code, PCA and SM are available in Matlab in pre-compiled and optimized form. The AE code was supplied to us by the first author of [1]. We wrote the LR, ISQ, and MM codes.

We note that PCA has very low complexity, much lower than all of the other five algorithms. However, we know from earlier sections that LR, ISQ, and especially MM beat 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 complexity. MM has the best performance by far but its complexity is about 2.5 times those of LR and ISQ and about 1.5 times that of SM. It has less complexity than AE. Although its complexity is at a disadvantage than PCA, SM, LR, and ISQ, the performance gains with it are substantial.

Chapter 5

Conclusion

The LR, ISQ, and MM algorithms presented in this thesis significantly outperform the three algorithms in the seminal paper, PSA, SM, and AE [1] in terms of performance. As in [1], we measure the performance in terms of connectivity (CT) and (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 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. We assumed a 2-D as well as a 3D environment, the latter being the same as [1]. We also used static channels and, in the case of ISQ and LR, single subcarrier CSI as in [1]. A more complicated simulation model and employing mobility in the channel model can show that this algorithm will perform even better.

Bibliography

[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*, 6:47682–47698, 2018.

[2] L. Le Magoarou. Efficient channel charting via phase-insensitive distance computation. *IEEE Wireless Communications Letters*, 10(12):2634–2638, 2021.

[3] 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)*, pages 1–6, 2021.

[4] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, Mar. 1986.

[5] 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*, 62(6):3242–3256, 2014.

[6] S. Selleri, 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*, 2018.