NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

LOCATION APPROXIMATION USING CHANNEL STATE INFORMATION BASED PRE-CODING IN A MASSIVE MIMO SCENARIO

by

Miriam H. Ewall-Wice

June 2022

Thesis Advisor: Ralph C. Robertson
Second Reader: Murali Tummala

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 2022	3. REPORT TYPE AND DATES COVERED Master's thesis

4. TITLE AND SUBTITLE
LOCATION APPROXIMATION USING CHANNEL STATE INFORMATION BASED PRE-CODING IN A MASSIVE MIMO SCENARIO

5. FUNDING NUMBERS

6. AUTHOR(S) Miriam H. Ewall-Wice

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The objective of this study is to investigate, in a pure MIMO scenario, how the ability of the system to determine the location of the user is affected when the number of antennas surpasses the number of users. With the increase in the number of antennas, the multi-user MIMO becomes a massive MIMO scenario. In this study, the established pre-coding matrices for different users will be used to determine the location of a user in a line-of-sight, or Rician, channel set-up with random non line-of-sight elements. The localization approximation will be a matter of comparing the "closeness" of the pre-coding matrix from the transmission to the actual user channel matrix. Zero-forcing is used to establish the matrix necessary to determine the location of users in multiple grid sizes for one, two, and five users.

14. SUBJECT TERMS
massive MIMO, MIMO, localization, pre-coding matrices, channel state information, mmWave

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

LOCATION APPROXIMATION USING CHANNEL STATE INFORMATION BASED PRE-CODING IN A MASSIVE MIMO SCENARIO

Miriam H. Ewall-Wice
Lieutenant Junior Grade, United States Navy
BS, Computer Engineering, United States Naval Academy, 2019
BS, Electrical Engineering, United States Naval Academy, 2019

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2022

Approved by: Ralph C. Robertson
Advisor

Murali Tummala
Second Reader

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The objective of this study is to investigate, in a pure MIMO scenario, how the ability of the system to determine the location of the user is affected when the number of antennas surpasses the number of users. With the increase in the number of antennas, the multi-user MIMO becomes a massive MIMO scenario. In this study, the established pre-coding matrices for different users will be used to determine the location of a user in a line-of-sight, or Rician, channel set-up with random non line-of-sight elements. The localization approximation will be a matter of comparing the "closeness" of the pre-coding matrix from the transmission to the actual user channel matrix. Zero-forcing is used to establish the matrix necessary to determine the location of users in multiple grid sizes for one, two, and five users.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

5G	5th Generation Wireless Networks
BS	Base Station
CDF	Cumulative Distribution Function
CIR	Channel Impulse Response
CSI	Channel State Information
LOS	Line-of-Sight
MIMO	Multiple-Input Multiple-Output
MM Wave	Millimeter wave
NLOS	Non Line-of-Sight
PDF	Probability Density Function
RP	Reference Point
RSS	Received Signal Strength
UE	User Equipment
ZF	Zero-Forcing

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Professor Roth for assisting me in the beginning of this thesis, and Professor Robertson for helping me to finish this. Also, I would like to extend my gratitude to LCDR Chapman. Finally, I want to thank Sarah Barkley for her unending support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In this chapter, we outline the goal of this thesis and go over a brief background on the history and relevancy of multiple-input, multiple-output (MIMO) communication.

A. OBJECTIVE AND PROBLEM STATEMENT

The objective of this thesis to explore the ability to determine user location with previously given channel-state information (CSI) by looking at linear pre-coding matrices from the CSI of several channels modeled in a multiple-input, multiple-output (MIMO) downlink. An overview of what a MIMO downlink looks like is given in [1]. We provide more details on this in Chapter II. The channels are modeled in an urban environment with various scattered or non–line-of-sight paths (NLOS), a strong single line-of-sight (LOS) path, and a strong reflected path. CSI fingerprinting has been previously explored for indoor localization in Wi-Fi environments in [2], [3] and outdoor environments in [4], and these are only a few methods of many covered in a survey in [5]. The goal of this thesis was to explore a CSI based fingerprinting method for an outdoor environment and scale up the size of simulation models to be concurrent with a massive MIMO scenario. More information regarding massive MIMO, and what "scaling up" refers to, is explored in later chapters.

B. ORGANIZATION

This thesis is organized into five chapters. In Chapter I, we examine the problem statement and objective of the research. A brief explanation of MIMO communications and the downlink is also provided. An overview of massive MIMO, a discussion of previous fingerprinting and localization methods, and an explanation of channel modeling is presented in Chapter II. An overview of simulation methodology is given in Chapter III. Simulation results and analyses are discussed in Chapter IV. The conclusions and a discussion of possibilities for future work are discussed in Chapter V. Attached to the end of this thesis is an appendix that contains the MATLAB code used for the simulations, figures containing simulation results, and a list of references used to conduct literature reviews and background understanding of MIMO communications and channel fading.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

In this chapter, we provide a brief overview of MIMO communications. We explain how a basic MIMO downlink works. Additionally, we provide an overview of previous work done in fingerprint-based localization methods. We describe the relevancy of massive MIMO in modern communication systems and describe the Rayleigh and Rician fading channels used for modeling.

A. MIMO

A key aspect of this thesis is understanding MIMO communications systems and channel behavior for such communications. As described in [1], MIMO was initially investigated for a single-user scenario but was later developed for multi-user cases where the users could have one or multiple receiver antennas as well as multiple transmitter antennas in the case of an uplink. The basic set-up for MIMO communications consists of *m* transmit antennas from a single base station to one user with *n* receiver antennas, as shown in Figure 1.

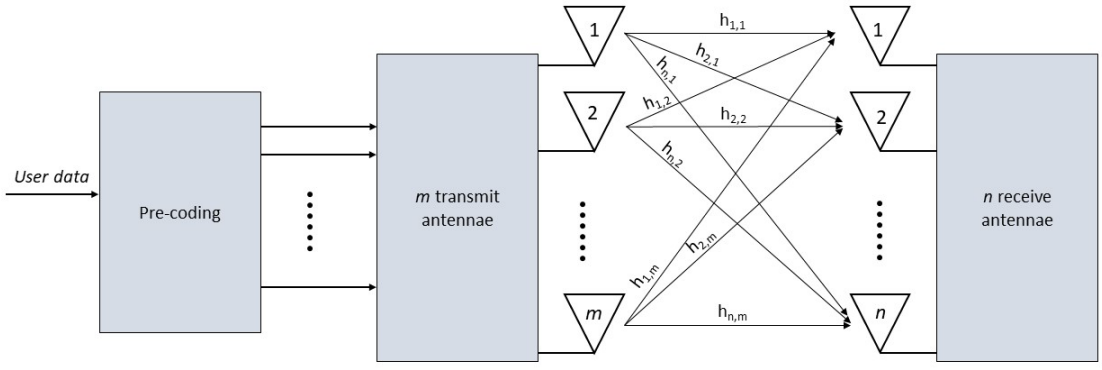

Figure 1. A basic MIMO downlink

The channel between the user receiver antennas and the base station transmitter antennas is described by the channel matrix **H**. The relationship between **H**, the user, and base station is modeled by the equation to describe a standard MIMO communication link,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where **y** is an $n \times l$ vector representing the received signal, **x** is the transmitted $m \times l$ transmitted signal, and **n** is added noise, usually in the form of additive white Gaussian noise. The number of frames transmitted while the channel is static is represented by l. The channel response at the time of transmission is given by **H**. While the channel is static where the transmitted symbol does not exceed the coherence time of the channel, the channel impulse response (CIR) components of **H** remain unchanged [6]. The coherence time represents the time it takes before the channel information changes. The typical makeup of **H** is shown as

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n,1} & h_{n,2} & \cdots & h_{n,m} \end{bmatrix} \tag{2}$$

where each element $h_{n,m}$ represents the channel response between the transmitter antenna i and the receiver antenna j. Components of this matrix account for any multipath fading, doppler shifting, or scattering that may occur between the transmitters and receivers. Multi-user cases were initially brought to the forefront, but for data streams to be sent out to multiple users there had to be a way to separate the signal of each user. A transmission signal intended for one user would still be received by another user. The problem of user signal interference is illustrated by Figure 2. A method of tackling this issue that has been developed is to pre-code the data before it is transmitted by the transmitter antennas. Pre-coding allows for data to be uniquely coded to be received by a designated user and ignored by others.

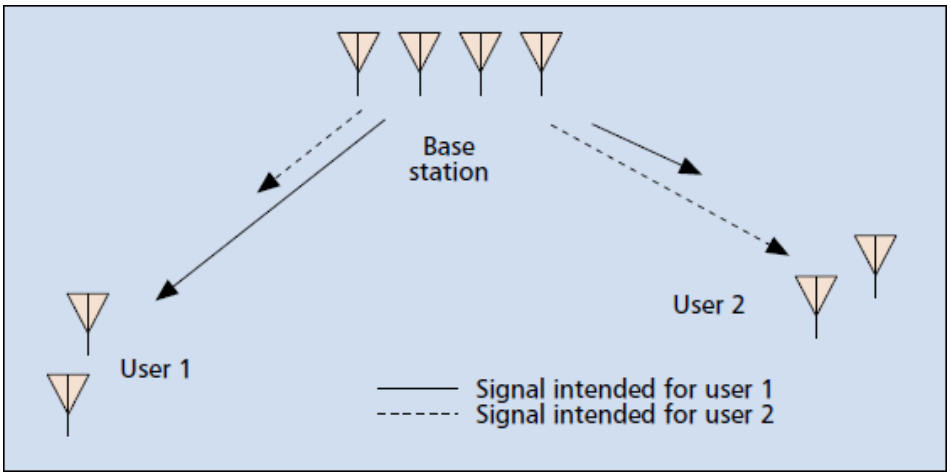

Figure 2. A multi-user MIMO downlink showcasing that signals may arrive at other users that they were not intended for. Source: [1].

One of the key aspects of any communication channel is what happens between the transmitter and the receiver. This is the area we chose to focus on. As previously discussed, the channel is described by **H**. In our survey of previous work, we considered other methods of fingerprinting or using previously stored information as comparison to newly measured data. The fingerprints described are received signal strength (RSS) and CSI-based fingerprinting. The key difference between the two is which part of the communication link they refer to. RSS refers to a magnitude of measured power while CSI may refer to several aspects of the channel, such as phase, power, and fading. For this thesis, CSI refers to the individual complex values that make up the channel matrix, and it is the numerical representation of our channel model. This is what we use for our chosen method of user localization.

B. PREVIOUS WORK AND LITERATURE REVIEW

User localization in communication systems may be provided through a varying range of the methods. The most common method for outdoor localization is the global position systems (GPS). Methods for indoor localization include RSS, CSI-based fingerprinting, and other methods. The authors of [5] provide a brief overview of several localization methods. Our work aims to take some of the previously implemented indoor techniques and apply them to a simulated urban communication scenario within a MATLAB simulated environment.

Additionally, we aim to simulate a massive MIMO scenario using large transmitter antenna arrays operating in the GHz ranges.

The previously applied CSI fingerprinting techniques explored in this literature review have been done indoors rather than in an urban environment. Fingerprinting is a way of taking a measurement or feature of, in this case, the channel and storing it as a "fingerprint" to be used as a comparison metric for later iterations of the channel or features. Our simulations are set up to simulate the fading channels corresponding to a crowded suburban street or a city. Our goal is to model larger areas and establish CSI fingerprinting using some of the concepts associated with massive MIMO. In this section, we summarize the methods described by the authors of [5], and specifically describe some previously explored methods for CSI-based fingerprinting given in [2]–[4].

In [5], the authors describe indoor localization as difficult because of low probability for a LOS path in a multipath environment. There is no single strong path to use to analyze a received signal. Instead, there are several signals arriving with different delays. The same argument may be made for an outdoor urban environment. The method of interest to this thesis described in the survey is scene analysis. The section in [5] on scene analysis describes the use of fingerprints obtained from the area of transmission, or the scene. The fingerprint described has characteristics that depends on the location of the transmitter or receiver. This method employs two stages like those described in [2], [3]; an offline stage for environmental survey where offline fingerprints are collected, and an online stage where observed signals are compared to that of the stored information [5]. In [5], the authors describe using signal strength for fingerprinting purposes. Later in this chapter, we describe studies that were done using CSI rather than signal strength, but they applied algorithms like those described in this survey paper. Of the five positioning algorithms described in the survey done by Liu et. al., the *k*-Nearest Neighbor (*k*NN) method draws the most interest for the work done in this thesis. The *k*NN algorithm described in [5] involves using RSS from the online phase to find the match with the lowest root-mean square error between the offline fingerprint and the online fingerprint. The *k*NN method focuses on RSS rather than CSI, but the authors of [2], [3] applied probabilistic and deep learning methods to using CSI for fingerprinting rather than RSS.

As stated earlier, a CSI-fingerprinting method for indoor localization has been explored for indoor WiFi environments in [2], [3]. We provide an overview of the work done by both studies in this section. The work by the authors in [2] was done to provide accurate CSI-based localization in an indoor WiFi environment that employed orthogonal frequency-division multiplexing (OFDM) MIMO with 30 subcarriers. The authors collected CSI information for 19 different locations within the same floor to use for their fingerprinting technique [2]. The methodology behind their fingerprinting was based on two vectors: an amplitude and a phase difference vector [2]. They collected information from these 19 locations and compiled the channel information for each subcarrier into a single-value magnitude and a single-value phase of the difference of each of the 30 subcarrier channel averages. Each subsequent channel average was subtracted from the one before it to create two 1×29 vectors. The last channel of the subcarrier average was dropped because it was implemented in the difference to the subsequent carrier.

The authors in [2] used a *k*NN algorithm and a probabilistic Bayes' rule for localization. The *k*NN approach focused on using a Euclidean distance calculation and a look at the lowest matching distance as similar to the survey method in [5]. The probabilistic approach was used to find which location would maximize the probability that the point is at that location given the testing CSI. The highest probability corresponded to the location estimate [2]. The authors had a limited scope of their experiment due to the limitations of a stationary controlled laboratory environment. They had completely stationary computers with limited variability of environmental parameters such as size and distance between UE and BS. The advantage of using simulations means we can vary our environmental parameters and area size to see how those conditions might affect the accuracy of our fingerprinting technique. Additionally, we are collecting CSI for a multi-user scenario. A base station is usually dealing with several users in a single cell for an urban environment. Our method is similar to that of the work done in [2] in that we look at matrix norms as an estimation for how close our channel realizations are to stored information from an off-line training phase. If we compare the pre-coding matrix of each new channel realization to multiple matrices, we can make estimations from the lowest value of the norm.

The authors of [3] also looked at using CSI for localization. Their study differs from the work done in [2] with the methodology they used for fingerprinting from the CSI; they used a deep-learning based approach for indoor fingerprinting. They dubbed this approach DeepFi [3]. To give a brief overview, the authors of [3] take in CSI in a training phase, analyze said fingerprints through the use of a four-layer deep network. The goal of the deep network was to reduce training complexity [3]. The authors of [3] make a note that CSI amplitudes are more stable than RSS values when information is collected at a stationary location, which provides context for why one might choose CSI as a fingerprinting reference over RSS. The authors conducted their experiments in two environments: a living room and a laboratory. Like the authors of [2], the scope of their experiment is limited to indoor environments, and the sizes of their simulation environments are static.

A study conducted in [4] was focused on using CSI fingerprinting in an urban environment with a massive MIMO scenario. An offline phase was proposed where fingerprints were collected from a grid and stored by the BS to be matched later using *k*NN for fingerprint matching. The authors of [4] conducted their simulation in a sector with a range of 500 m. They also varied the interval at which they were fingerprinting within their simulated environment. The authors had a fixed simulation size that did not vary. Our goal was to do similar work to the authors in [4] but to vary the environment size and explore different user configurations similar to how these authors randomized their user locations. Varied environments allow us to look at how accuracy might be affected for small or large cells being serviced by a single BS. We also based our fingerprinting on the existence of strong LOS and NLOS components.

Some of the work done in looking at MM wave CSI-based fingerprinting has been done in [7] where the authors use an intermediate channel measurement of spatial beam signal-to-noise ratios for their fingerprints. These authors conducted their measurements in an indoor environment like the authors in [2]–[4]. They use the beam SNR measurements as their fingerprinting method and use deep neural networks to generate a location estimation. Another method for MM wave fingerprinting was explored in [8]. These authors look at RSSI and the direction of arrival of the signal from access points to determine their location estimates. Their scheme is also done at an indoor location with various scattered paths. The

authors of [8] use references to establish a known database of RSSI and direction of arrival information and then match online RSSI and directions of arrival to the stored information using a *k*NN approach to determine the location estimate. The authors of [7], [8] conduct their data collection in an indoor environment.

The work done in this thesis expands upon previous CSI fingerprinting models by moving the indoor localization concept to an outdoor urban environment setting and exploring wireless communications concepts employed in 5G technology. We look at CSI for a small city setting where there still exists one LOS between the user equipment (UE) and the BS with an additional strong NLOS reflected path. The advantage of using simulations means we can vary our environmental parameters and area size. We will go further into depth on our results in future chapters. Additionally, our simulations consider multiple users rather than just one with a massive MIMO scenario. As we will discuss later, massive MIMO is an integral part of future communication technologies such as 5G, and we employ aspects of a massive MIMO and MM wave for our CSI fingerprinting scheme and simulations. Some work in the area of fingerprinting has been done for MM wave communications in [7], [8] using RSSI and direction-of-arrival and spatial beam SNR, respectively. We also look at the pre-coding matrices generated from CSI in addition to the CSI itself. The pre-coding matrices act to diagonalize the channel matrix and for users to receive the signal intended for them rather than one intended for another user. Data collection and processing are conducted in a simulation environment to evaluate a variety of environmental scenarios with the employed 5G technology aspects.

C. MASSIVE MIMO AND ITS CURRENT RELEVANCY

Several companies have begun the rollout of 5G, or 5th generation wireless networks. The technology promises to have faster download times and communication speeds [9]. One of the new technologies associated with 5G communication systems is massive MIMO. Erik Larsson and several other authors give an overview of the advantages and disadvantages of massive MIMO technology in [10].

The focus of massive MIMO is to go larger through the employment of large volume of transmission antenna. Massive MIMO relies on scaling up the number of antennas at the

BS. The BS will have a significantly higher number of antennas compared to the number of UE antennas. According to an overview in [10], massive MIMO has the potential to increase capacity to possibly ten times more than traditional MIMO systems. The systems have the ability to focus a large amount of energy at one location through coherent superposition of waves being sent to one location [10]. The waves combine destructively at all other locations. Massive MIMO has further benefits in low latency times and the ability to be built with inexpensive components [10].

In terms of relevancy to millimeter wave (MM wave) technology employed by 5G systems, massive MIMO presents an effective way of employing 5G technologies. Technology that employs MM wave refers to the high frequency, millimeter-sized wavelengths employed for future 5G communication systems. The small wavelength of MM wave technology allows for closely spaced antennas at the BS, designated by the condition that they have a spacing of $\lambda/2$ where λ is the carrier wavelength. When the number of transmit antennas increases, the product of two channel estimates approaches zero or becomes orthogonal. This behavior is called channel hardening and occurs with the increase in the number of transmit antennas at the BS. It is one of the key advantages of massive MIMO [10]. Massive MIMO employs the increase in the number of transmit antennas at the BS and has a close relationship with future 5G networks, which is why we chose to scale up the number of BS antennas for the later stages of our simulations.

D. CHANNEL FADING MODELS

Our ability to accurately model realistic channels is presented as one of the main aspects of this thesis. We examine two of the possibilities for channel modelling and channel characterizations in this section. Channels follow random patterns modeled by a probability density function (PDF). The probability models themselves are predictable. By understanding the behavior of wireless channels, we can simulate our own channel coefficients for MIMO channels in an environment such as MATLAB. In this section, we describe exactly what these channel models are before giving a description of how they are implemented.

A channel between the transmitter and receiver may consist of multiple paths over which the signal can reach the receiver due to reflections, scattering, other signals, etc. Thus,

many channels are multipath channels. Multipath fading is modeled as *small-scale fading*. The channel is time-variant and changes with slight movements occurring along the propagation path from cars, people, atmospheric conditions, etc. The channel also experiences changes due to movements by the transmitter or receiver as well. The impulse response

$$h(t,\tau) = \sum_{i=0}^{I} a_i(t)\delta(t-\tau_i) \tag{3}$$

which represents the summation of multiple taps along a channel delay line, may be used to describe the channel behavior given multiple paths. The path gain of the *i*th path is given by a_i, I is the total number of paths between the transmitter and receiver, τ_i is the propagation delay along path *i*.

With this impulse response in mind, we needed a way to model $a_i(t)$. The method of modelling the complex gain depended on whether we were dealing with a line-of-sight component or not. We explored two statistical models for our channels described in the subsequent sections: Rayleigh and Rician fading.

1. Rayleigh Fading Channels

When no line-of-sight path exists between the transmitter and receiver, the magnitude of the fading process can be described by a Rayleigh fading model. The model has a PDF described by

$$f_R(r) = \frac{r}{\sigma^2} e^{-r^2/(2\sigma^2)} \tag{4}$$

where σ is the standard deviation, and r is the random variable representing the magnitude of the path gain $a_i(t)$ [11]. The CIR produced by the fading model appear as a zero-mean complex Gaussian process in nature.

2. Rician Fading Channels

The other fading channel primarily used in our modeling and simulation is the Rician model. The implication of Rician fading is that the magnitude of each channel tap has a non-zero mean and is modeled by the PDF [11]

$$f_R(r) = \frac{r}{\sigma^2} I_0\left[\frac{Ar}{\sigma^2}\right] e^{-(r^2+A^2)/(2\sigma^2)}, \tag{5}$$

where A is the specular power of $h(t,\tau)$, i.e., the power of the LOS or the strong reflected path and I_0 is a zeroth-order modified Bessel function.

E. PRE-CODING METHODS

One of the key ideas MIMO systems employ to send users their intended data is pre-coding. If User 1 and User 2 are both receiving information from the same base station, pre-coding is the method by which the BS avoids sending User 1 the information of User 2 and vice versa. Rather than directly transmitting user data streams. The streams are run through a pre-coding algorithm before being transmitted by the BS on the downlink. Pre-coding may be done through linear or non-linear algorithms. Non-linear algorithms require greater complexity and computational power than linear. A linear method was chosen for this thesis. Linear pre-coding methods are relatively simpler and easier to implement. They require less data processing than non-linear methods such as dirty paper coding [1]. The linear pre-coding method explored for our simulation environment was zero-forcing (ZF). The ZF pre-coding matrix can be used to compute how close each new channel realization is to the stored training values.

1. Zero-Forcing

ZF presents a simple way of ensuring user signals do not interfere at the receiver in the downlink. If the transmitter has access to full CSI from the receivers and each user has only one antenna, then the ZF pre-coding matrix is generated for the case where, in an $n \times m$ MIMO scenario, $n \leq m$ [12]. The pre-coding matrix $\mathbf{P}$ is given by the pseudo-inverse of $\mathbf{H}$ shown as

$$\mathbf{P} = \mathbf{H}^{\dagger}. \tag{6}$$

If the columns of the channel matrix $\mathbf{H}$ are linearly independent, the pseudo-inverse is calculated by

$$\mathbf{H}^{\dagger} = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}, \tag{7}$$

and if the rows are linearly independent,

$$\mathbf{H}^{\dagger} = \mathbf{H}^{\mathrm{H}}(\mathbf{H}\mathbf{H}^{\mathrm{H}})^{-1}. \tag{8}$$

If both the rows and columns of **H** are linearly independent, then the pseudo-inverse is just the inverse of **H**.

The pseudo-inverse accounts for situations where **H** is not a square matrix and is, therefore, not invertible. When **H** is $n \times m$ and **P** is $m \times n$, then

$$\mathbf{H}^{\dagger}\mathbf{P} = \mathbf{I} \tag{9}$$

where **I** is the identity matrix. From this relationship, the channel becomes diagonalized, and the received signals that are not meant for each user are cancelled out. This pre-coding must be done by the BS because the users are not coordinating with each other. The transmitted signal with pre-coding becomes

$$\mathbf{x} = \mathbf{P}\mathbf{s} \tag{10}$$

where **s** is the original data of the users or user original data. One of the key advantages of ZF pre-coding is that it is easy to implement and requires little computational power. With perfect CSI it is as easy as inverting the channel matrix formed from said CSI. ZF is not without its disadvantages though. One of the major disadvantages described in [1] include that for a low signal-to-noise ratio (SNR) situation, ZF may be sub-optimal to overcome interference at the receivers [1]. ZF requires more power to invert channel matrices that are not properly conditioned. Because of its easy implementation, we used ZF as the primary pre-coding scheme for our simulations. Our simulations are based on generating channel information over an off-line period, then generating pre-coding matrices based on the number of user receivers the BS is communicating with and comparing that stored information with newly generated channel realizations. We expand upon how our pre-coding scheme was simulated and run in Chapter III.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SIMULATION SETUP

Our simulation methodology is divided into three sections. First, we look at how we modeled the fading environment and establish CSI. Second, we describe our offline training phase where we take CSI from each transmitter to each reference point (RP) and put together our fingerprint training database. Finally, we execute an online phase where we match new channel realization pre-coding matrices to stored fingerprints.

A. CHANNEL MODEL SETUP

The first step in creating our simulation environment was to establish a working channel model and confirm that that channel model followed the predicted behavior as described in [11]. We had to establish what determines an accurate fading channel. Fading channels may be modeled as complex Gaussian random processes or as the sum of multiple sinusoidal components [11], [13]. In our simulations, we modeled the fading channels as Gaussian random processes. The sum of sinusoids method proved to be more computationally taxing and would achieve results similar to the Gaussian modeling method. To save computational power for CSI fingerprint matching at later stages of simulation, we used the complex Gaussian model.

For the work done in this thesis, fading was modeled by generation of Rician and Rayleigh taps with additive white Gaussian noise (AWGN). We assumed each user would receive a signal from NLOS multipath components, a LOS path, and a strong NLOS path. The NLOS reflected path is a building or structure within the environment that still produces strong signal power when the signal bounces off it. Therefore, it was modeled as a Rician tap. We had knowledge of the exact distance for the LOS and NLOS strong reflected path for each reference point (RP) on the grid. With this, the BS could estimate the phase of the signal arriving at the LOS path and the phase of the NLOS strong reflected path of the Rician fading taps. The phase for each tap was calculated from

$$\theta = \frac{2\pi \times d \times f_c}{c} \tag{11}$$

where d is the path length in meters of the LOS and NLOS strong reflected path, respectively, f_c is the carrier frequency, 30 GHz in this case, and c is the speed of light, 3×10^8 m/s. The free-space path-loss equation allowed us to generate the non-zero mean for the two Rician channel taps. The normal path loss model is given by,

$$P_r = P_t \times L_0 \left(\frac{d_0}{d} \right)^{\alpha} \tag{12}$$

where P_r is received power, P_t is the transmitted power, L_0 is the free-space path loss at the reference distance d_0, d is the distance between the transmitter and receiver, and α is the path loss exponent. In our case, α is 3 for an urban environment and our reference distance d_0 was one meter. The free-space path loss L_0 is calculated from

$$L_0 = \frac{G \times \lambda^2}{(4\pi \times d_0)^2} \tag{13}$$

where G is the antenna gain. The antenna gain G is given by

$$G = \frac{4\pi \times A_e}{\lambda^2} \tag{14}$$

where A_e is the effective aperture area. In our case, we set this to one. Substituting (14) into (13) we get the path loss at the reference given by

$$L_0 = \left(\frac{1}{4\pi \times d_0^{\,2}} \right). \tag{15}$$

With our reference distance set as one meter and the path loss exponent of three, the equation for P_r becomes

$$P_r = P_t \left(\frac{1}{4\pi} \right) \left(\frac{1}{d} \right)^3 \tag{16}$$

Given the received power, the variance is derived from the signal-to-noise ratio (SNR) as

$$\text{SNR} = \frac{P_r}{P_n} \tag{17}$$

where P_n is the noise power. To determine the SNR for the LOS path, we established a baseline distance in the grid as the central point of the grid to the location of the transmitter for each given grid size. For the baseline of the established grid, the SNR was set as a standard 20 dB. The location of the RP in the training phase was compared to the baseline location. If the distance was greater than the baseline distance, the SNR was scaled to a lower value. If the distance was lower than the baseline distance, the SNR was scaled to a higher value. Our final Rician tap estimation was given by

$$h_{rician} = \sqrt{P_r} e^{j\theta} + \mathcal{CN}(0, P_n). \tag{18}$$

This principle was what allowed us to generate a theoretical fading coefficient for each possible channel. Approximately 30 channel taps were generated for the multipath channel taps, and a collection of those were used for each channel estimation. Because of the zero-mean nature of the Rayleigh fading channel, we did not need to include all 30 channel taps in the channel model. A single channel tap was generated for the LOS and NLOS reflected path through the summation of the generated Rician coefficients while the small, scattered Rayleigh taps were generated from an exponential decay applied to a gaussian random number generator to scale the taps to have a lower magnitude for a larger delay τ. Ultimately, the Rayleigh taps sum together to create a zero mean. The first step of our exponential decay model was to establish the number of taps we needed for the channel estimate. This was determined by randomly choosing a value from a Poisson distribution with a rate of occurrences set to an arbitrary value of four. The specific location of these taps along the delay line was chosen at random from the 30 possible channel taps.

An exponential scale was used with a decay constant of one to significantly lower the magnitude of the tap located at the furthest delay. The scale was applied to the generated complex gaussian values. The closest value had a real and complex value of one, so the taps were further scaled using a generated variance from the Rician *K* factor. The phase of the Rayleigh channel taps was uniformly distributed. The variance of the Rayleigh taps was the magnitude of the combined LOS and strong NLOS specular path divided by a designated *K* value. The variance was generated by scaling

$$K = \frac{P_{LOS,NLOS}^2}{P_{scattered}^2} \tag{19}$$

where $P_{LOS,NLOS}$ is the power associated with the Rician components of the fading channels, in order to have the variance equal the noise power that would result from a certain *K* value. The variance for the taps generated was associated with a *K* value to represent the ratio of the power of Rician taps to the power of the scattered multipaths. Given the power from the scattered multipath components $P_{scattered}$ and an arbitrarily chosen *K* value, we generated a standard deviation. We used $K = 3$ for our simulations. The *K* value was chosen based on the shape of the curve produced by the Rician pdf. Our decision for this value was driven by the ability of a Rician pdf with this value to be modeled as a gaussian random process. A value too large would be unrealistic for an urban environment, and a value too small would make the fading channel purely Rayleigh, defeating the purpose of modeling a Rician fading channel in the first place. Therefore, $K = 3$ was used for the simulations.

The overall channel estimate was generated from the sum of the NLOS reflected path, the LOS path, and the scattered multipath taps given by

$$h_{nm} = h_{LOS} + h_{NLOS} + h_{scattered}. \tag{20}$$

Our check to ensure we had correctly generated a proper working channel was to observe that the phase from several channel realizations averaged together matched the theoretical phase difference between the NLOS and LOS paths. As described earlier, the channel taps for a Rayleigh fading channel produces zero mean and looks like a gaussian random process over several realizations. Combined with the taps for the NLOS strong reflector and the LOS path generated by Rician elements, the overall average for multiple realizations should match the amplitude and phase of several combined generated LOS and NLOS strong reflector paths. These results are confirmed in Figure 3 and Figure 4. With confirmation that the channels followed predicted behavior, we confirmed that they would be valid for use within the simulation. With a working channel model, we take results and compare them to conduct our analysis of how accurate our user location estimation scheme

was and what our results said about the ability to estimate user location given CSI based pre-coding matrices. Our results and analyses are discussed in the next chapter.

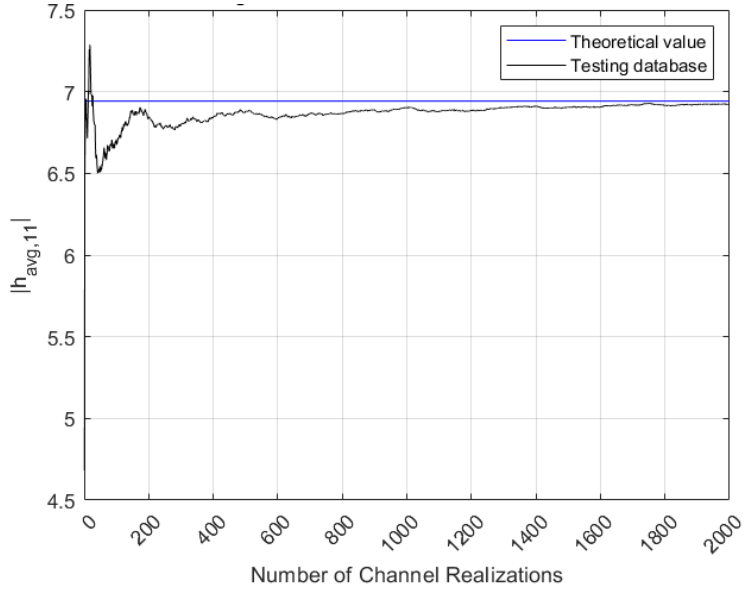

Figure 3. Running average of the magnitude of several realizations of the training database compared to the magnitude of the Rician path components.

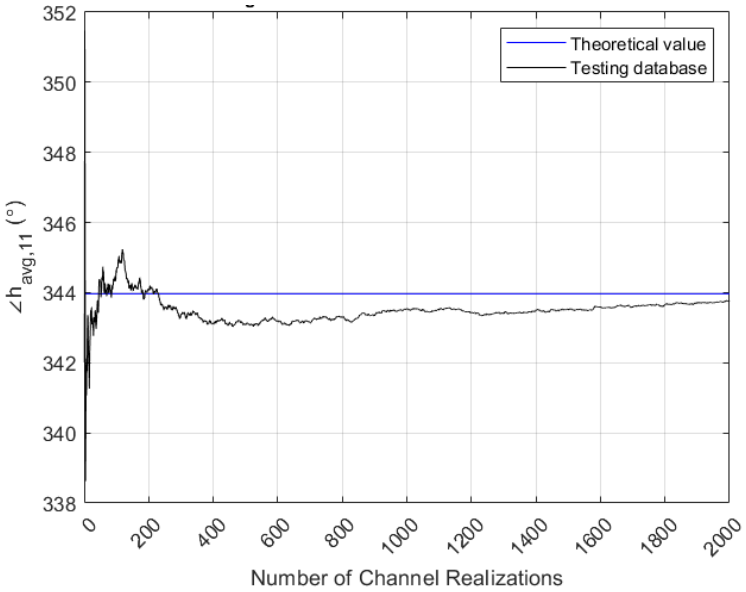

Figure 4. Running average of the phase of several realizations of the training database compared to the theoretical phase calculated as the difference between the LOS and the NLOS path.

B. SIMULATION ENVIRONMENT

With channel models, we needed to establish the cell for the BS. The cell of the BS in our model consisted of an $X \times X$ grid with a strong reflector at a fixed location and either one or multiple UE randomly dispersed throughout the grid. We designated the same reflector location for all simulations conducted to keep a consistent model.

The first set-up in our simulations was a 4 × 4 grid, where each square was either 1 m^2 or 9 m^2. We wanted to create smaller grid to simulate a micro-cell. The two square sizes were used to see if and how distance between training points affected accuracy. Other grid sizes included 5 × 5 and 7 × 7. The total number of square locations within each grid size was designated by G. We explored various changes in the parameters that made up the simulation environment. These parameters included the number of grid squares, the size of the grid squares, number of transmitters, and total number of users. Simulation complexity increases drastically with an increase in the number of users, transmitters, and number of grid squares. This was also a factor that determined the sizing of our grids. We made the determination of whether to run larger grid sizes based on computational requirements and whether there would be more gain in doing so. The total number of channel models to setup the training database $N_{Channels,\ training}$ is given by

$$N_{Channels,training} = N_{Tx} \times N_{rp} \tag{21}$$

where N_{Tx} is the total number of transmitters and N_{rp} is the total number of reference points. Given that the number of reference points used equates to the total number of grid squares not including the transmitter, the total number of channels is

$$N_{Channels} = N_{Tx} \times (G-1), \tag{22}$$

where the total number of squares in the grid G is

$$G = X \times X. \tag{23}$$

The basic setup for the grid is shown in Figure 5.

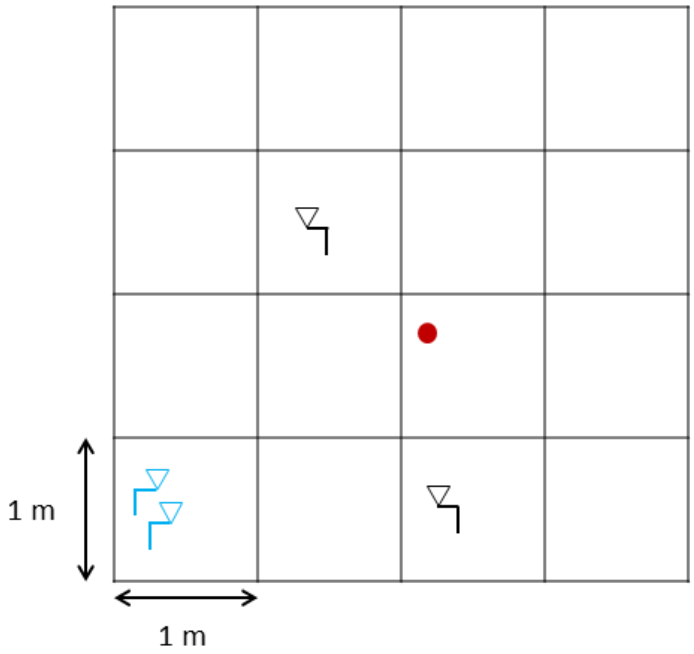

Figure 5. A generic example of the simulation grid for a 2 × 2 MIMO scenario. Transmission antennas are shown in blue, two users are shown in black, and the strong reflector is the red dot.

The channel paths are further included in Figure 6, where the LOS and the strong reflected paths are indicated.

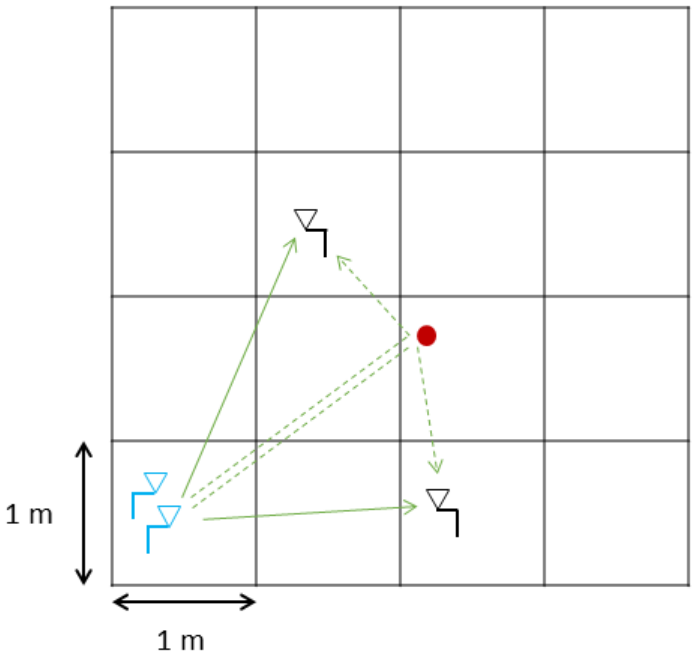

Figure 6. A generic example of a 2 × 2 MIMO setup scenario with the LOS and strong NLOS path indicated for one the of the transmission antennas. The transmit antennas are shown in blue, and the strong reflector is designated as red. Each square in this model is 1 m^2, but sizes were adjusted in simulation.

While the grids displayed in Figure 5 and Figure 6 show 1 m^2 for each square in the grid, multiple square sizes we used. The transmitter antennas were always located in the same grid location for each simulation, as was the reflector. The only location changes occurred with the UE. UE locations were randomized for each simulation.

C. OFFLINE TRAINING STAGE

The next step of simulation was to establish a stored database of all offline measurements in what we refer to as the training sample database. The measurements were taken at the center of each grid square. As an example the simulation training locations are shown in Figure 7. The locations are fit to a 1-m^2 square size, but the size was varied for the simulations. The grid square sizes used in simulation were 1 m^2 and 9 m^2. Each grid square was assigned a numeric value ranging from $g = 1,\ldots,G$ to be stored in the training

database. The training sample database consisted of $G-1$ values for the channel coefficient corresponding to each transmitter-RP pair.

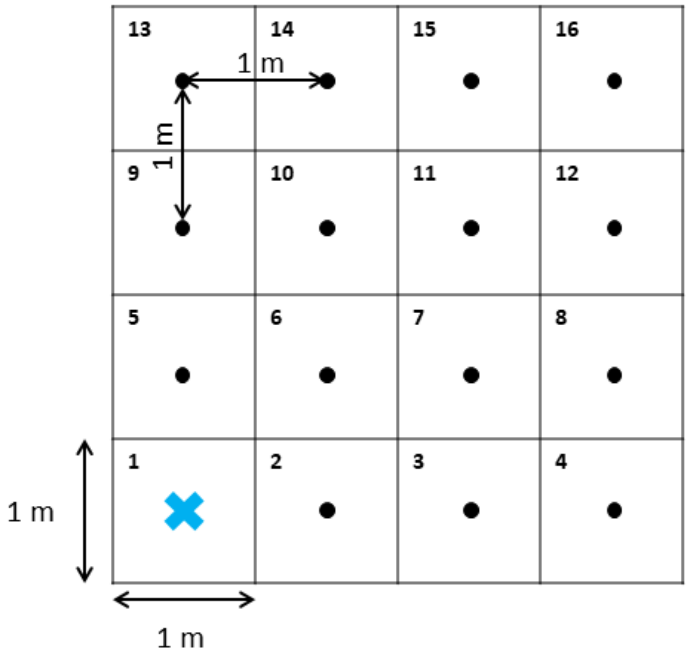

Figure 7. Grid simulation environment showing RPs for the training database calculation during the offline phase. In this case the transmitter is located at $g = 1$, designated by the blue "X."

For each simulation run, the training sample database contained all 10 000 training samples of CSI between each transmitter and each possible location on the grid for a user location. The averages for the training samples settled out after within the first 300–400 samples so 10,000 provides a stable average to use for comparison. This is similar to the offline stages described in [2]–[4]. These averages were used to compile the entries in the training channel matrix database $\mathbf{H}_{\text{training}}$. The training database matrix $\mathbf{H}_{\text{training}}$ was used for as a comparison metric for the location estimate. Simulations were run for k users where k was 1, 2, and 5. For designated k users, the database contained every possible combination of users on the grid given by the binomial coefficient

$$\binom{G-1}{k} = \frac{(G-1)!}{k!((G-1)-k)!}. \tag{24}$$

With each possible user combination, the training channel matrix database also contained the corresponding channel matrix $\mathbf{H}_{\text{training}}$. For example, for the 4 × 4 grid with two users, the training channel matrix database has $\binom{15}{2}$ possible user combinations, equating to 105 combinations. For each location combination, there was a stored matrix of size 2 × m, where m is the number of transmitter antennas. As the grid size increases, the training channel matrix database contains more entries. As the number of users or transmitter antennas increases, each entry has a larger sized matrix. The simulation complexity had a massive increase as these elements were scaled up.

D. ONLINE EXECUTION

After the offline stage, an online stage is run to take actual realizations of the channel to be compared to the training database. The basic algorithm for the online phase is as follows. User locations were first randomly generated. These generated locations were used for multiple channel realizations, where new channel estimates were calculated. With each new channel estimate, a pre-coding matrix was established. That matrix was compared to each entry in our training database using a norm to establish a distance between the matrices. The location entry with the corresponding lowest norm was used as the location estimate of the users.

The new data from the online stage deals with generating individual new channel realizations rather than generating a running average of channel realizations. After establishing the offline channel matrix training database, actual grid square locations for the users were randomly generated. From there, specific locations within the squares were established as though the users were randomly located somewhere within the square. Each new online channel realization was designated as $\mathbf{H}_{\text{test}}$. From $\mathbf{H}_{\text{test}}$, the zero-forcing pre-coding matrix $\mathbf{P}_{\text{test}}$ was established by taking the pseudo-inverse. For each new channel realization, the Frobenius norm was calculated to measure the distance between $\mathbf{P}_{\text{test}}$ and the pseudo-inverse of each stored training matrix designated as $\mathbf{P}_{\text{training}}$. We used the

Frobenius norm because it allows us to look at the similarity between each individual matrix element. Each matrix element described the behavior of the channel between each transmitter and user. The Frobenius norm allows us to gauge the overall closeness between the two matrices based on elementwise differences. Two matrices that are almost equal or have similar channel behavior theoretically have a smaller norm. The Frobenius norm calculation to find the distance between the two matrices is given by

$$\left\| \mathbf{P}_{test} - \mathbf{P}_{training} \right\| = \sqrt{\sum_{n} \sum_{m} \left(p_{test,mn} - p_{train,mn} \right)^2}, \tag{25}$$

where p_{mn} describes the element of each matrix. The location(s) corresponding to the smallest calculated norm was stored as the location estimate(s) for the user(s). The distance error for each user linking with the base station was determined from

$$d_{error,n} = \sqrt{(x_{actual} - x_{rp})^2 + (y_{actual} - y_{rp})^2} \tag{26}$$

where x_{actual} and y_{actual} are the actual coordinate locations of the user. The RP locations corresponding to the grid square location determined via the minimum norm calculated in (25) are represented by x_{rp} and y_{rp}. The distance error is calculated for each new channel realization. For a multi-user case, the location estimate used for each user corresponded to the matching order of the reference points for that fingerprint. For example, say two user locations were [2,5], (i.e., one user at square #2 and one user at square #5), and they matched to the fingerprint stored for locations [3,6]. The algorithm calculates the distance from the user at square #2 to the reference point in square #3 and calculates the distance from the user at square #5 to the reference point in square #6. The algorithm considers strictly ordered user locations. This allowed us to keep a consistent condition for location estimation but restricted the possibility for having a more accurate system by incorporating the distance from #3 to #5 or #2 to #6.

A running average was kept for the distance error of each new channel realization to establish a performance metric. The running average of the error for each new channel realization stabilized after a few hundred realizations for each environmental setup. Because of this, we ran 250, 500, or 1,000 realizations for each randomized user location.

The 500 realizations correspond to the case where we have a larger number of users: 5 and 10. Because the complexity of the simulation increases significantly with the number of channel models we had to generate, we had to run fewer simulations. For each grid size and square size, we ran 500, 1,000 or 2,000 simulations. Fewer total simulations were run with more users and larger grids to save computational time. For clarification, the total number of simulations accounts for the total number of randomly generated user location(s), and the total number of realizations was the number of newly generated precoding matrices compared to the stored fingerprints. The empirical CDF of the mean distance errors and the number of times the users were within a given radius from the correct reference point were used as performance metrics for the system.

IV. RESULTS AND ANALYSIS

To measure the success of our results, we looked at the number of times the fingerprinting algorithm deduced that the users are located within a certain radius of the location estimate. Because the location estimates are not exact, we did not achieve a distance error of zero. Our ability to estimate the location of the user was based on the mean-distance error discussed in Chapter III. We established a radius around the RP that we consider to be an accurate location estimate for the user. This was a way to say the user falls within the correct square. We used 0.5 m for the 1-m and 1.5 m for the 3-m RP spacings. We also analyzed the cumulative distribution function (CDF) based off the empirical data for the mean error distances given the different number of transmitter antennas.

A. INITIAL FINDINGS AND OBSERVATIONS

We observe the behavior of the accuracy of the fingerprinting model for 1, 2, and 5 user scenarios. Our initial observations are based on the empirical CDFs for each grid size and each square size.

Our first simulation scenario looks at a single user to establish a baseline for what results look like when the transmitter is only focusing its energy on one user within the grid.

1. Single User Scenario

The database for the single user contains only single-user fingerprints. These prints are vectors consisting of the channel information for each transmitter-to-user pair. In Figure 8 and Figure 9, we see the start of a trend that follows through for all our scenarios. For the case where the RPs are spaced apart by 1 m (i.e., the 1-m^2 square size), we see that there is a large difference in accuracy between the case of two and 20 transmitter antennas and the case of 128 and 256 transmitter antennas. Specifically, about 15% of the generated mean error distances for the two transmitters fall within an accuracy of 0.5 m, and 12% for the 20 transmitter antennas fall within that accuracy marker. There is a significant drop for the

128 and 256 transmitter antennas. Both cases have only 10% of their mean error distances falling below the accuracy marker. The reverse happens for the 9-m^2 square size where the RPs are spaced 3 m apart. We see the mean error distances for the 20, 128, and 256 antennas has approximately 15% of the mean error distances falling below the accuracy marker, and the two transmitter antennas falls below 10%. We start to see a trend in this area as we change the size of the grids and vary the number of users. The percentages of mean error distances that fall within the accuracy marker are given in Table 1. Figure 16, Figure 17, Figure 18, and Figure 19 for the 4×4 grid scenario are in the Appendix. Analysis of the results given are provided in later in this chapter.

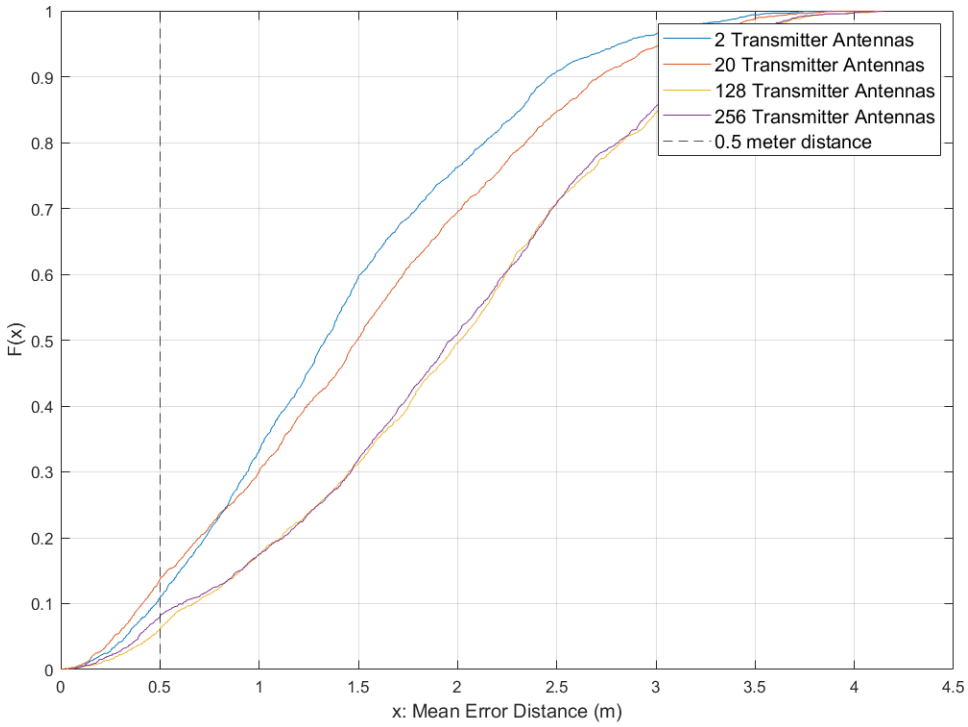

Figure 8. Empirical CDF for single user in a 4×4, 1 m^2 grid.

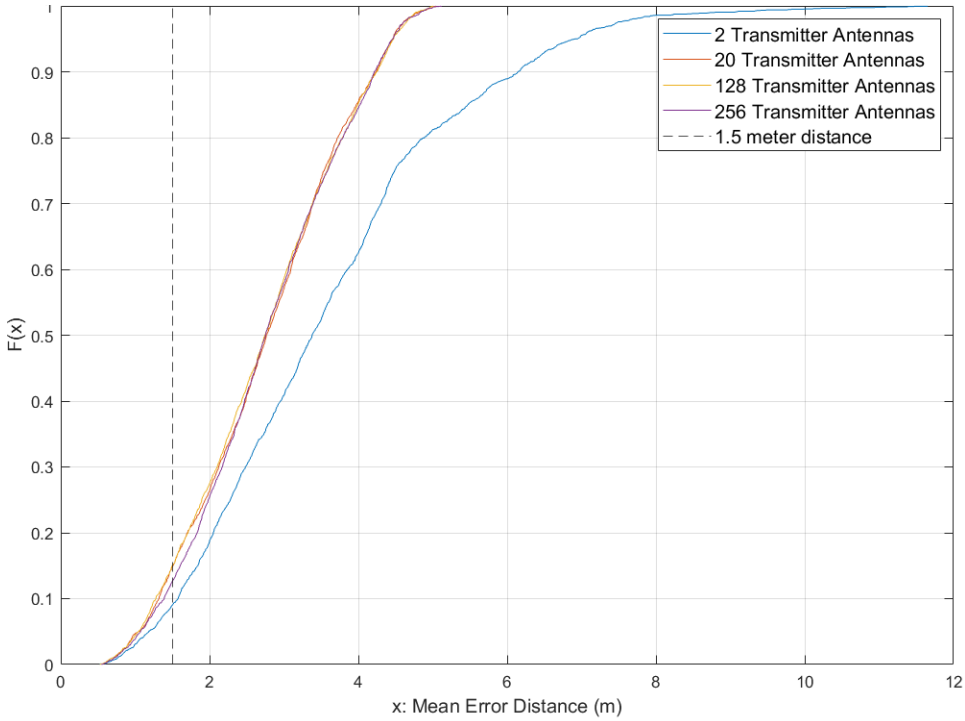

Figure 9. Empirical CDF for single user in a 4×4, 9 m^2 grid.

Table 1. Single User Accuracies

		Percentages based on Number of Transmitter Antennas			
Grid	**Square**	2	20	128	256
4×4	**1 m^2**	10.95	13.7	7.9	6.15
	9 m^2	9.0	14.9	15.05	12.65
5×5	**1 m^2**	8.9	12.5	5.75	4.75
	9 m^2	13.0	20.15	21.25	22.85
7×7	**1 m^2**	5.25	11.9	4.5	3.8
	9 m^2	10.55	18.1	16.9	16.75

2. Two-Users Scenario

We scale our simulations to include two single antenna users. For each simulation the user locations are randomly generated for multiple realizations of the channel. As we did for the single user case, we observe a 4×4, 5×5, and a 7×7 grid. Two square sizes are run for each grid: 1 m^2 and 9 m^2. We see a significant decrease in accuracy from the single user case to the two-user case. This is probably attributed to the total number of possible combinations of user locations that grows significantly as we increase the total number of users. The percentage of results that fall within the accuracy markers of 0.5 m and 1.5 m are shown in Table 2. The trends we see with these percentages closely follow what we found for the single user.

Table 2. Accuracies for two users

		Percentages Based on Number of Transmitter Antennas			
Grid	**Square**	2	20	128	256
4×4	**1 m^2**	4.6	5.4	1.5	1.5
	9 m^2	5.35	9.25	10.1	8.75
5×5	**1 m^2**	3.05	4.3	0.06	0.085
	9 m^2	8.5	19.75	19.75	19.9
7×7	**1 m^2**	1.2	2.75	0.045	0.05
	9 m^2	5.15	11.85	12.8	11.5

For comparing these results to our baseline single user results, we look at the empirical CDFs for our two-user cases. In Figure 10 and Figure 11, the results closely match the single-user case in terms of general trend for the CDF shape. Attached in the Appendix, Figure 20, Figure 21, Figure 22, and Figure 23, are plots of the CDF for the other two-user cases and reinforce the data shown in Table 2. The ordering of the curves for the two-user cases matches that of the single-user cases. We see a significant reduction in how many of the mean error distances fall below the accuracy marker. For the single-

user case 4×4, 1 m^2, we saw those percentages for each case fall between 7% and 12%. For the two-user case, we see these percentages cut by more than half, falling between 1% and 6%.

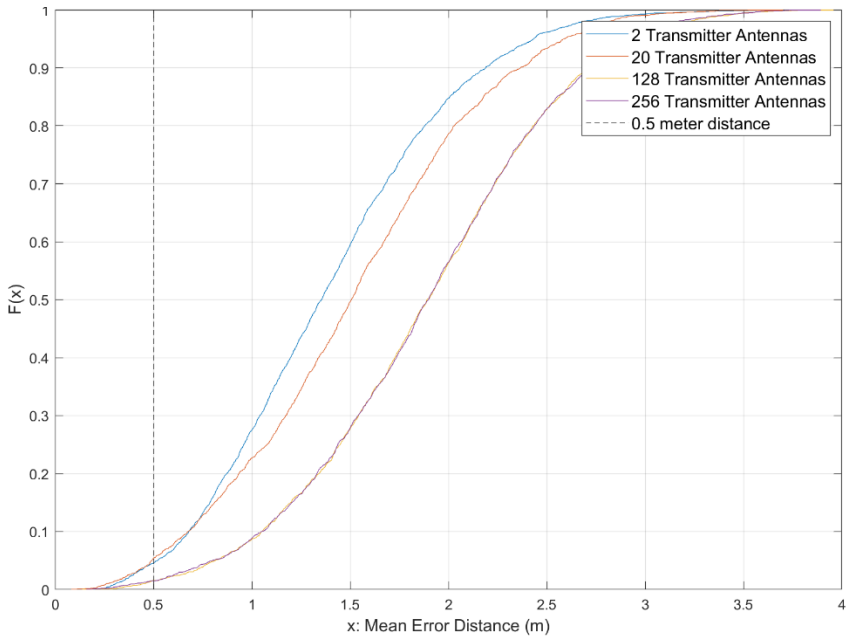

Figure 10. Empirical CDF for two users in a 4×4, 1 m^2 grid.

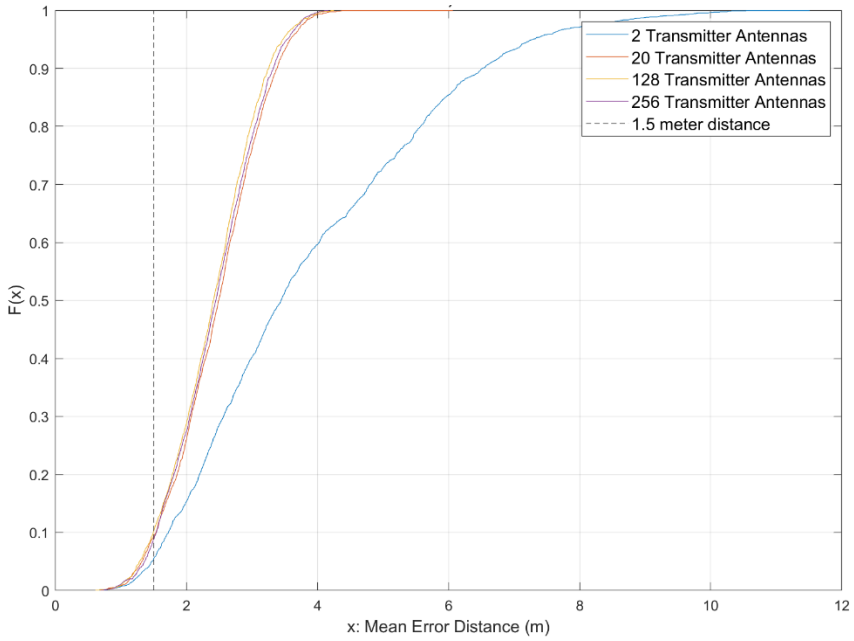

Figure 11. Empirical CDF for two users in a 4×4, 9 m^2 grid.

3. Five-Users Scenario

In our five-users case, we see a clear degradation in the percentage of users that fall within the reasonable error margin. The spread of the mean error distances is shown in Figure 12 and Figure 13. Figure 24 and Figure 25 in the appendix are CDF plots that reinforce the degradation of accuracy seen in the five-user case. We see a change in the trend that we saw from our previous cases. The percentage of mean distance errors no longer improves from changing the RP spacing to 3 m. None of the distance errors in Figure 13 fall within the "acceptable" accuracy margin of 1.5 m, while only approximately 1% to 2% of the cases for the 1-m^2 square size fall within the 1.0-m marker. What we can gather from these results is that complexity plays a large role in the ability of the system to accurately predict multiple user locations. Due to the computation requirements of running simulations for five users, we chose to forego larger grid sizes. We can clearly see a degradation in accuracy for having five users. The system has too many possibilities to consider.

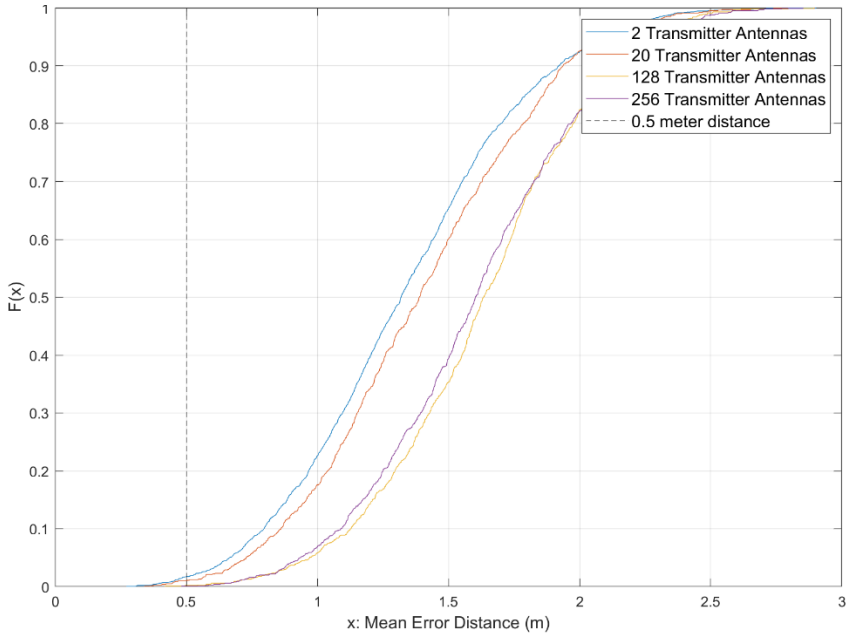

Figure 12. Empirical CDF for five Users in a 4×4, 1 m^2 grid.

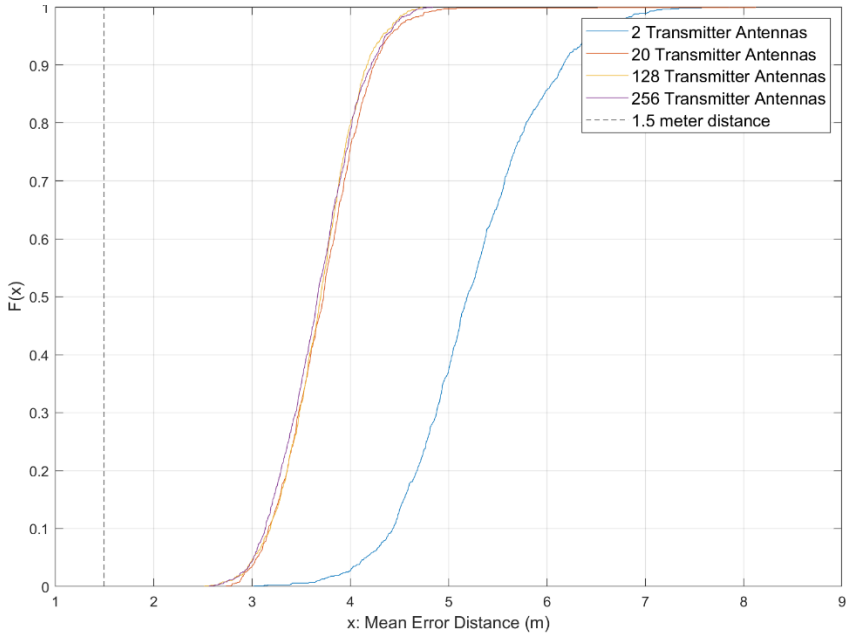

Figure 13. Empirical CDF for five Users in a 4×4, 9 m^2 grid.

B. ANALYSIS OF FREQUENCY EFFECT ON PHASE

In our results, we notice an initial sense of randomness associated with the CSI. In other words, we notice that the channel information differs widely based on the location of the users. The difference in CSI from a user location is explained by the nature of how the CSI is generated based on distance from the transmitters and phase, but it was unexpected to find that something as small as a millimeter change would cause such drastic CSI changes. In this section, we discuss how the CSI may change with the slightest movement of the UE and attribute some of our findings to that movement. When we set up our simulations, we had the UE locations generated at random. Thus, they would never align exactly with the RP location. Our initial thought process was that the path would be similar enough to the path to and from the RP. The CSI for the UE would have a magnitude and phase that fell within a close range of the magnitude and phase of the combined NLOS strong reflected path and the LOS path.

Based on our findings, there are certain locations where the location of UE is predicted correctly every time and other locations where the correct location of the UE correct location is never determined. There are also a few scenarios where we found accurate RP prediction for some cases but not for others with the same user location. We attribute this disparity to the phase changes that occur with small variations in location within the grid. The cause of many location estimation errors is due to large changes in phase because of small changes in distance from the RP. We also observe that the 9-m^2 square sized simulations have a considerably higher accuracy rate where the RP correctly matches to the square that the user is in. This may indicate that if this methodology was used for actual deployed Massive MIMO technology, then fewer measurements could be taken while maintaining above average performance.

We look at two communication examples in Figure 14 and Figure 15 to describe how the change in location affects the change in phase.

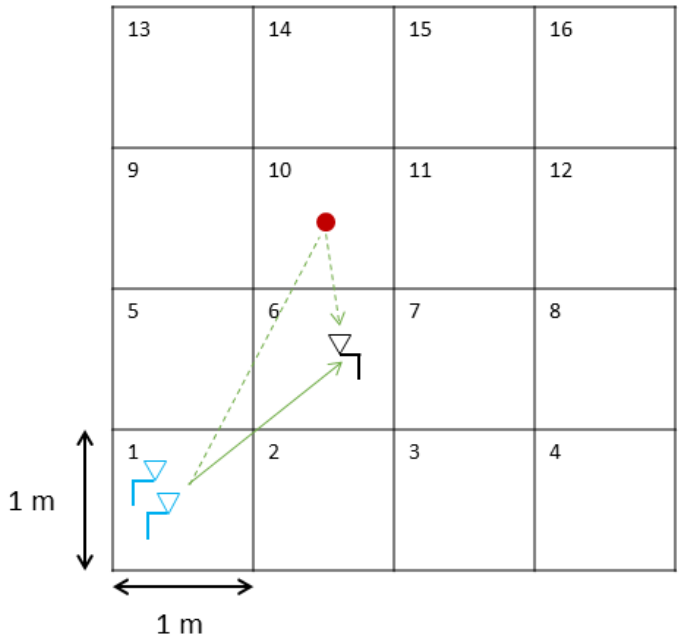

Figure 14. Visual display of LOS path and NLOS strong reflected path for a user located in square #6.

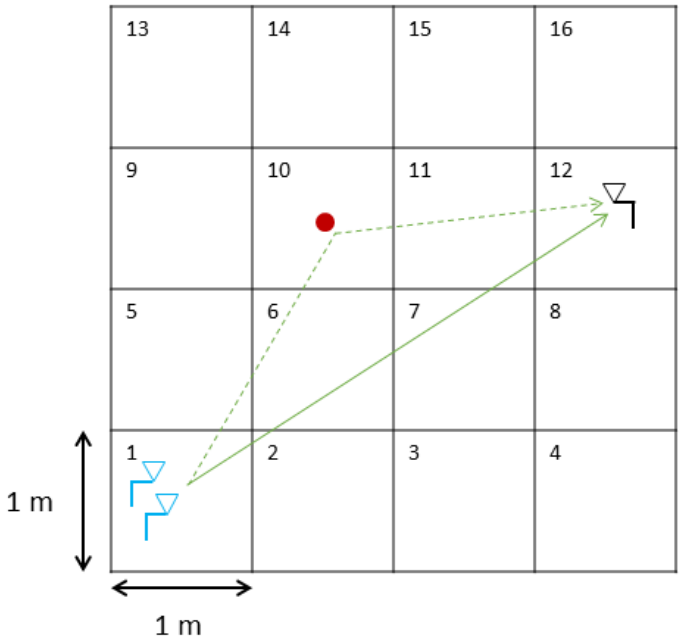

Figure 15. Visual display of LOS path and NLOS strong reflected path for a user located in square #12.

The initial purpose of these two scenarios was to observe how the angle of the bounce path might be affecting accuracy results. Instead of angle of the bounce path, we find that frequency plays a large role in how well the system performs. This is the result of how the channel behaves when the frequency approaches the MM wave realm. For our examples, we have a situation where the user is in square #6 and another where the user is in square #12. We calculate the phase of the bounce and direct paths to the RPs in these respective squares. We have two transmitter antennas so there are two individual values for each bounce and direct path making up a total of four phase values. We use (11) to calculate our phase for each path. For the purposes of this explanation, we focus on the phase values between one of the transmitters and the RP or UE rather than both transmitters. The phase difference between the transmitters is negligible. In Table 3, we see the difference in phase calculated for the UE and the RP. For each of these cases, we set the UE to be located 3 mm to the right of the RP to show the difference in phase. We can see in the table the difference caused by moving the UE just 3 mm from the RP. The phases are not within an acceptable range to accurately predict the square in which the UE is located.

Table 3. Phases for 30-GHz Carrier Frequency, 1-m^2 square

Square Number	RP/UE	LOS Phase (°)	NLOS Phase (°)
6	RP	317.47	76.3137
	UE	34.0811	100.9
12	RP	327.91	134.96
	UE	55.8478	237.43

To further investigate our claim that frequency is causing large user location error, we investigate what happens when we lower the frequency to 1-GHz. We look at the same situation where the UE is in squares 6 and 12 as shown in Figure 12 and Figure 13, respectively. These findings are recorded in Table 4. We clearly see that the phase information for the UE located only 3 mm from the RP still closely matches the phase information that makes up the recorded fingerprint in the training database.

Table 4. Phases for 1-GHz Carrier Frequency, 1-m^2 square

Square Number	RP/UE	LOS Phase (°)	NLOS Phase (°)
6	RP	297.2332	358.7159
	UE	299.8715	357.5805
12	RP	48.4698	358.7159
	UE	51.4416	2.1315

These findings bring us to the conclusion that our chosen frequency of 30 GHz has an effect on accuracy in this system and brings up the question of what qualifies as an acceptable frequency for use in this specific fingerprinting methodology

While the phases match much more closely with the 1 GHz carrier frequency, MM wave systems operate at higher frequencies than 1 GHz. The phases closely match between the RP and UE up to about 4 GHz as shown in Table 5 for the UE located in squares #6 and #12. What we can gather from this is that as the carrier frequency increases, the UE is too far away from the RP for the phase to match closely enough. This is for a 4×4 grid scenario at 1 m^2 sized reference squares. In future work it may be worthwhile to explore whether increasing point density allows for better location estimation. If those points are closer together, then the database contains more information for the user to match to. The radius of accuracy would be smaller.

Table 5. Phases for 4-GHz Carrier Frequency, 1-m^2 square

Square Number	RP/UE	LOS Phase (°)	NLOS Phase (°)
6	RP	12.3789	310.3547
	UE	23.016	305.8133
12	RP	51.8618	310.3547
	UE	63.6232	324.017

We also explore how the frequency influences the phase for RPs placed in 9-m^2 sized grid squares. As with the 1-m^2 squares, we also look at two different squares within the grid to paint a full picture of our results.

Table 6. Phases for 30-GHz Carrier Frequency, 9-m^2 square

Square Number	RP/UE	LOS Phase (°)	NLOS Phase (°)
6	RP	52.6148	291.097
	UE	129.025	23.7205
12	RP	51.5964	10.8755
	UE	139.4979	106.3283

As was the case with the 1-m^2 squares, the phase data in Table 6 also shows that a carrier frequency of 30 GHz produces training and testing phases that are less likely to match if the UE is just 3 mm away from the RP.

We run the same test with 1 GHz as the carrier frequency. We see a clear improvement in Table 7, just as in Table 4. Additionally, we find that increasing the carrier frequency to 4 GHz also produces phases close to what is calculated in the RP.

Table 7. Phases for 1-GHz Carrier Frequency, 9-m^2 square

Square Number	RP/UE	LOS Phase (°)	NLOS Phase (°)
6	RP	357.2899	159.9204
	UE	349.8661	163.0079
12	RP	290.5540	282.5797
	UE	293.4976	285.7615

Table 8. Phases for 4-GHz Carrier Frequency, 9-m^2 square

Square Number	RP/UE	LOS Phase (°)	NLOS Phase (°)
6	RP	217.149	235.1729
	UE	227.36	247.5227
12	RP	302.8491	5.8101
	UE	314.5814	18.5371

With our data, we see a clear improvement in location accuracy with the 9 m^2 square sizes. Companies may not have to take channel measurements at as many points within a communications cell to still accurately predict where the UE is, but frequency must be lowered from the typical MM wave spectrum. Part of this phenomenon is in how we have defined accuracy for each scenario. By allowing 1.5 m rather than 0.5 m from the reference points, we have created a greater radius of accuracy. More points that are further away can fall within a range that are considered to be acceptably close to the RP. Having the RPs spaced further than 1 meter apart allows for a more accurate system because more points are included. They are still in the same square as the RP but can be further from it and still be considered “acceptable.”

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. SUMMARY OF RESEARCH AND FINDINGS

In our work, we generated a communications MIMO grid between UE and transmission antennae. Our goal was to assess the accuracy of a massive MIMO communications location identification based on CSI. We chose to conduct simulations for single user, two user, and five user scenarios. Using a non-zero mean gaussian distribution, we generated a training phase of CSI based on RP to store in an information database. We then set up a testing phase where UE would randomly be placed on the grid. The goal was to assess how well the estimated location of the UE matched the closest RP and where the reason for errors lay.

We found that our errors were due to our chosen frequency and the drastic millimeter changes seen in CSI. In our setup, we went on an assumption that wherever the UE ended up in the grid square, it would reflect a fairly accurate picture of CSI compared to that of the RP. As we discovered, this was not the case. Even the 1-m^2 spacing proved to be too close in point density. Due to the computational weight of the simulations we performed, we needed to limit the scope of how many grid sizes and spacings we were going to simulate. We are unable to come to a solid conclusion as to whether or not a middle ground grid size would have followed the same trend that we saw, where having the points spaced closer together was actually a detriment to our system "accuracy".

B. FUTURE WORK

There are several directions future continuation of this research could take to further explore MM wave communications and technology. A continuation of this work presents opportunities to explore other methods of channel comparison and to improve upon channel simulation as well. We made discrete decisions on certain parameters for the sake of simulation complexity and timeliness. For our data, we applied a Frobenius norm to analyze the "closeness" of the channel matrices. Additionally, the option of comparing the elements of the testing matrices could be directly compared to that of the training matrices.

Another direction for this work would be to change the density of the CSI fingerprints for the training database. In an area as dense as a city street or block, having points closer together may allow for better separation of channel information. Future research could be conducted on how exactly distance specifically affects the information itself. In this work, we only observed data for two spacings between points: 0.5 m and 1.5 m. There are an infinitesimal number of possibilities for point spacings that have yet to be explored. The density of the fingerprint points could allow for the use of higher frequencies closer to that of real-world 5G frequencies.

In our work, we only looked at one K value in our channel modeling. We specifically chose $K = 3$ to simulate a Rician fading channel. There cannot be too much variance in K because, as stated earlier, a value too large is unrealistic and a value too small makes the channels more Rayleigh in nature. Some finetuning could be done to observe how changes in K affect the results and accuracy of the Frobenius norm comparison. Additionally, other methods of channel modeling could be employed to make a more realistic city block. We wanted to use a baseline simplistic model to get an initial idea of how the location estimation would perform. Our method of pre-coding also went into how closely the channel information would compare for the training and testing databases.

We only looked at zero-forcing due to its ease of modeling and simplicity. As with the norm we decided to use, there are other methods of digital pre-coding available that may be worth looking into in future research. Other areas where this research could be taken include implementation of machine learning algorithms for finetuning the training data and obtaining more accurate location estimates. Different channel models can be used to create the channel databases.

This work has several directions it can be taken. We focused on taking a relatively simple channel design and running a large number of simulations. Future work could include modifying the channel design and running a smaller number of simulations. Additionally, rather than focusing on grid size and number of antennae, future research can be done to instead focus on the channel design. Our model is just one attempt at implementing a fingerprinting style localization method for massive MIMO systems.

APPENDIX A. SINGLE-USER CDF

This appendix contains additional CDF graphs for single-user scenarios to show the trends described in Chapter IV.

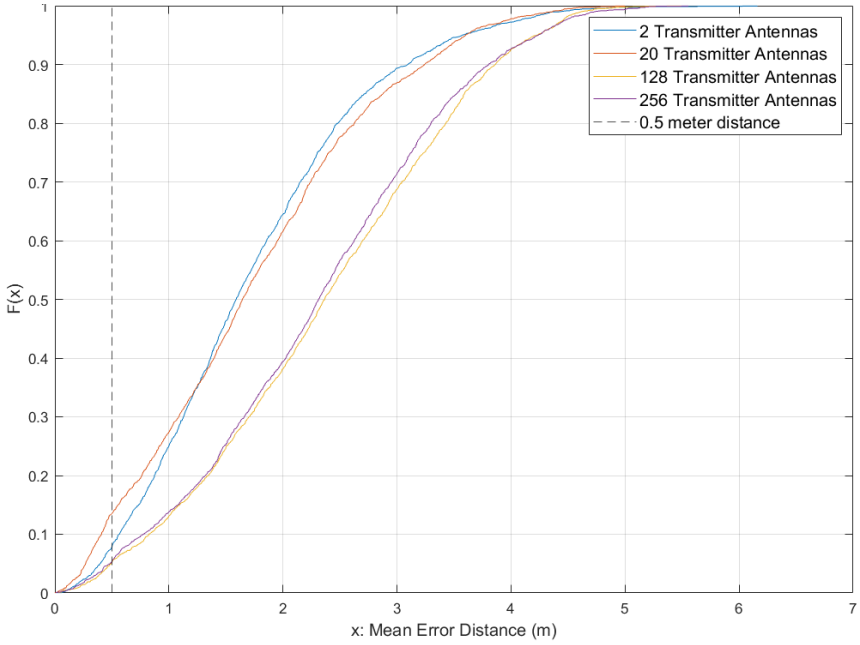

Figure 16. Empirical CDF for one user in a 5×5, 1-m^2 grid.

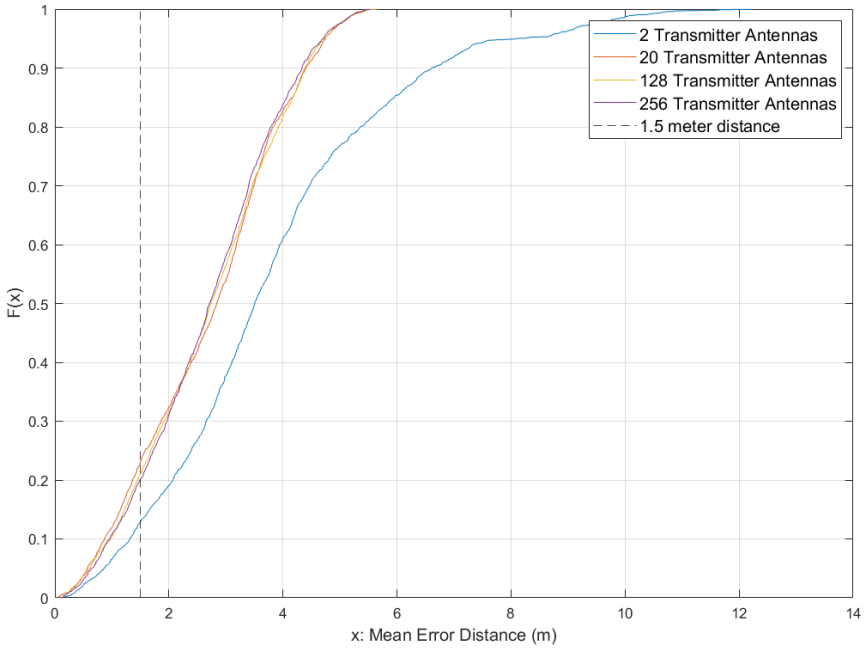

Figure 17. Empirical CDF for one user in a 5×5, 9-m^2 grid.

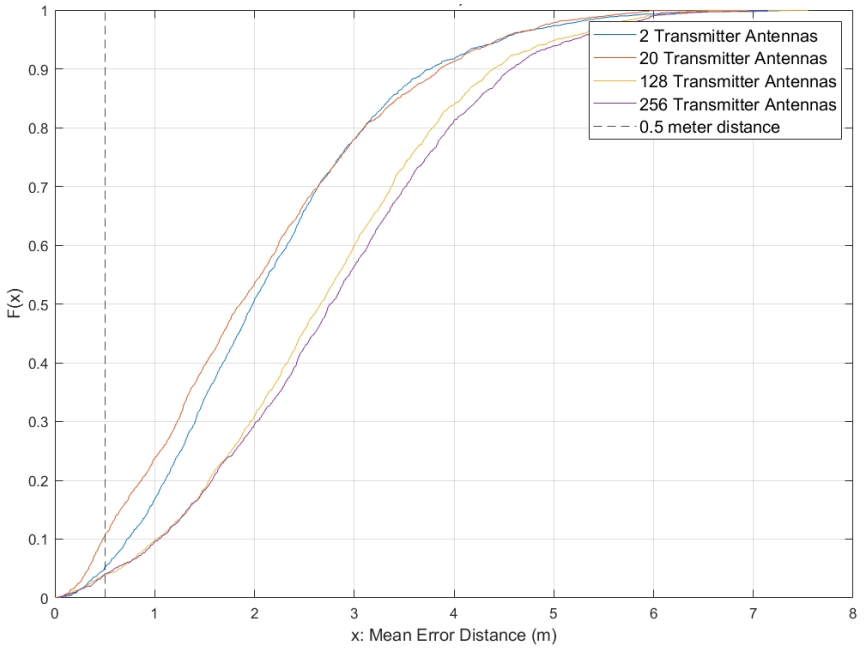

Figure 18. Empirical CDF for one user in a 7×7, 1-m^2 grid.

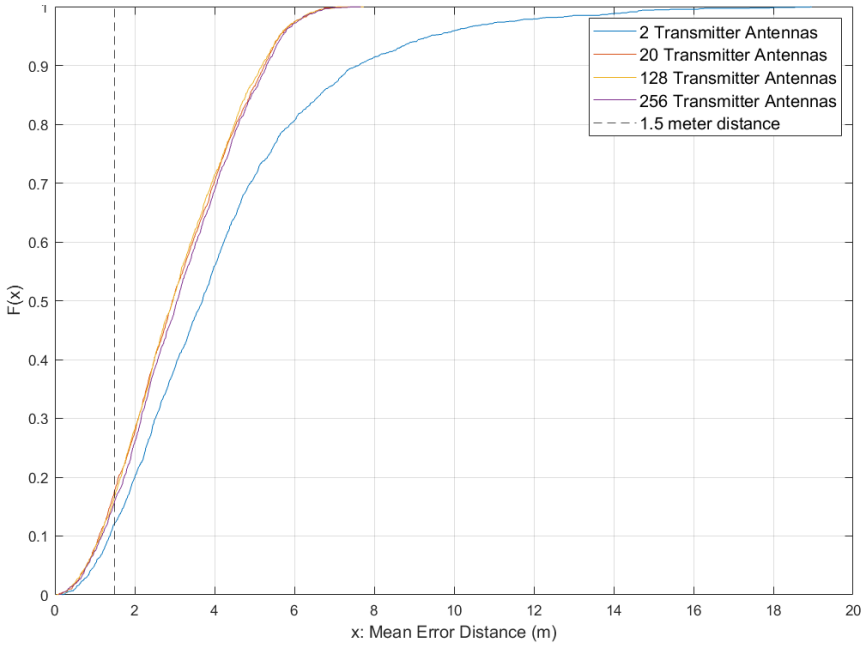

Figure 19. Empirical CDF for one user in a 7×7, 9-m^2 grid.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. TWO-USERS CDF

This appendix contains additional CDF graphs for two-users scenarios to show the trends described in Chapter IV.

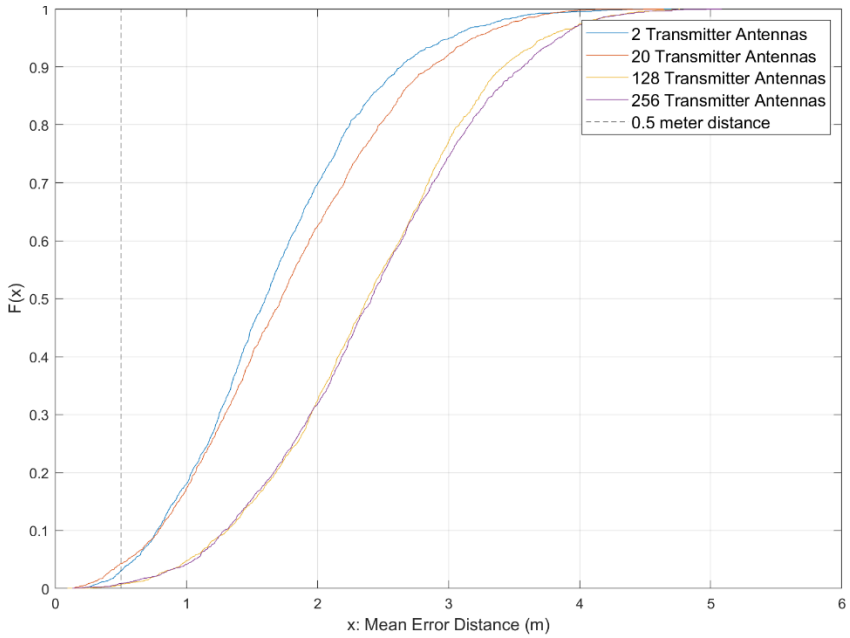

Figure 20. Empirical CDF for two users in a 5×5, 1-m^2 grid.

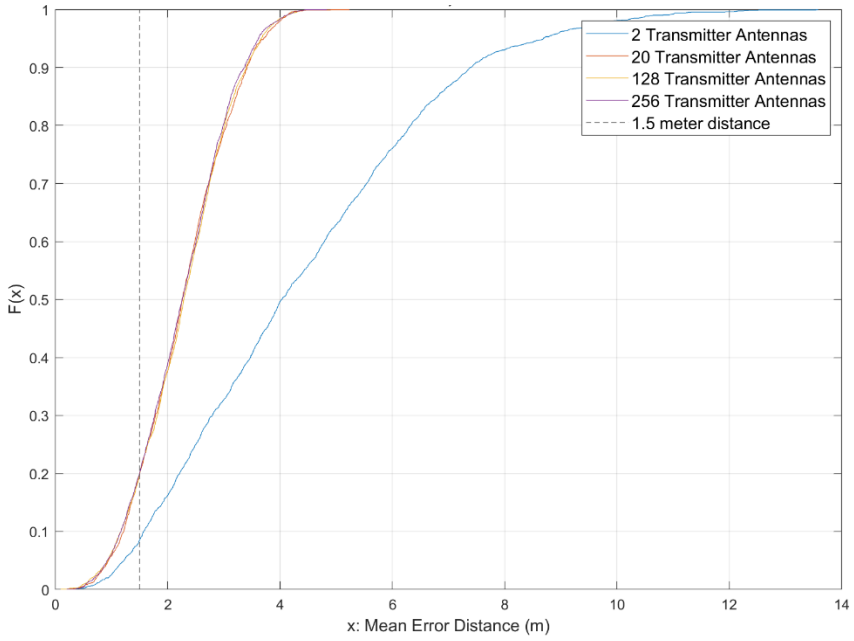

Figure 21. Empirical CDF for two users in a 5×5, 9-m^2 grid.

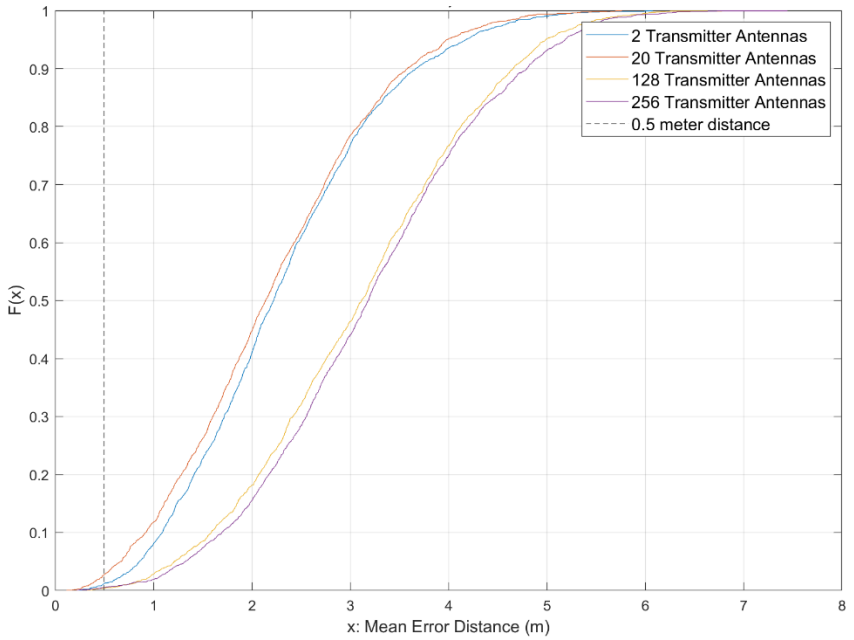

Figure 22. Empirical CDF for 2 Users in a 7×7, 1-m^2 grid.

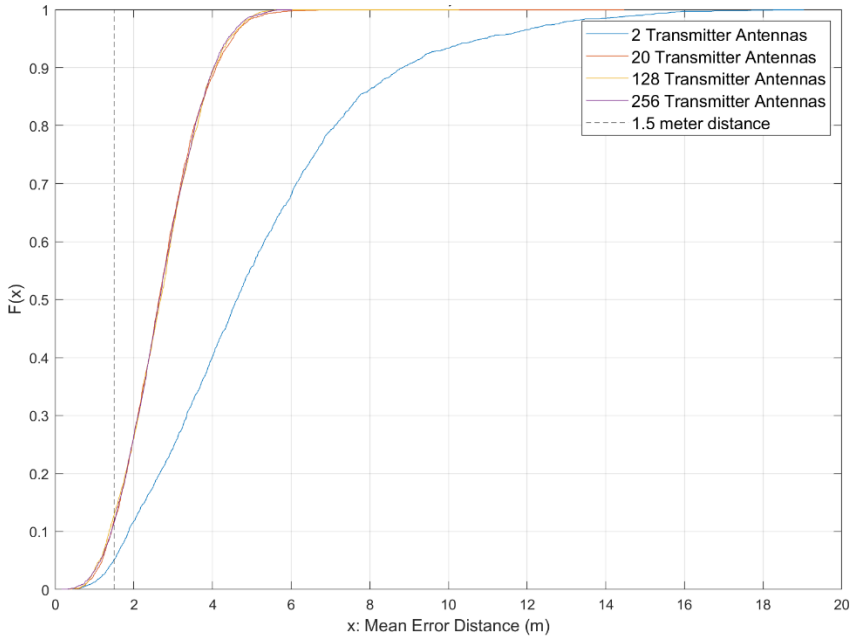

Figure 23. Empirical CDF for two users in a 7×7, 9-m^2 grid.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. FIVE-USERS CDF

This appendix contains additional CDF graphs to show the trends described in Chapter IV.

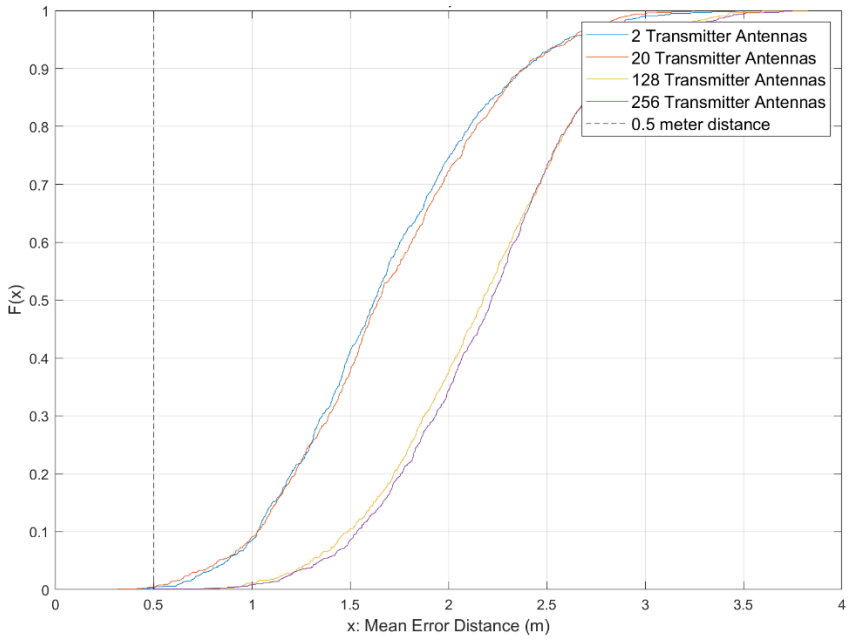

Figure 24. Empirical CDF for five users in a 5×5, 1-m^2 grid.

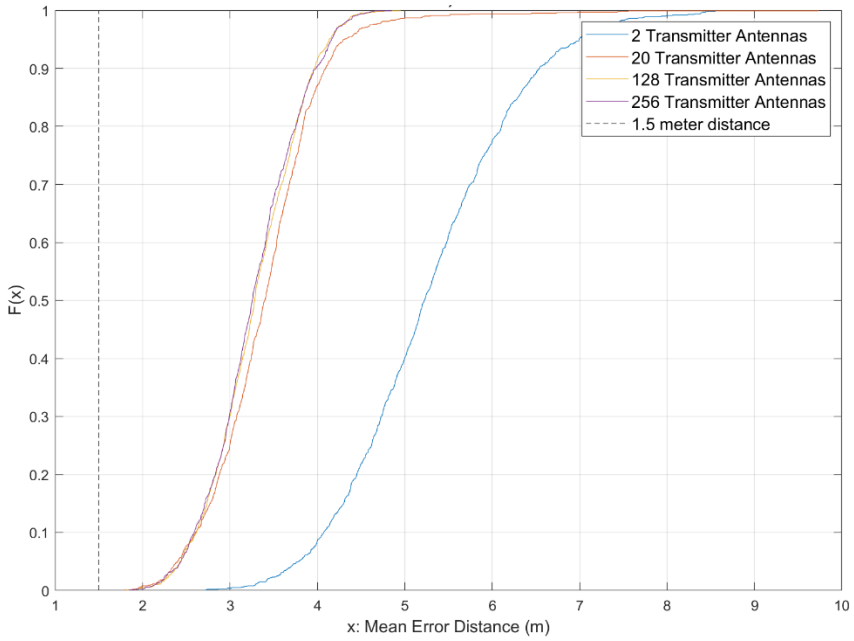

Figure 25. Empirical CDF for five users in a 5×5, 9-m^2 grid.

APPENDIX D. MATLAB CODE

This appendix contains the MATLAB code written to generate the training database, generate the actual testing databased, and compare the two using a Frobenius norm. The code also contains function to generate Rician channel taps that were used to populate **H**. The map function was written to generate individual number values to assign to each grid square.

```
%% System Specifications (Parameters)
%Grid Information
tic
parpool(32);
gridX = 7; gridY = 7;
rng('shuffle');
squareSize = 1;
gridSize = gridX*gridY;
map = Map([gridX,gridY],squareSize); %Create a map of
single values to the grid locations
%and assign actual locations of where the users are.

reflectorX = 2; reflectorY = 3;

reflectorLocation =
findLocation(map,reflectorX,reflectorY);

nTx = 2;
txX = 1;
txY = 1;
txLocation = [txX,txY]; %Grid number for where the
transmitter is located
txLocationActual = zeros(nTx,2); %store the specific
locations of the
nUsersMax = nTx;

K = 3;
txPower = 1;
fc = 30e9; %LTE Carrier freq.
lambda = 3e8/fc;

%transmission antennae, taking into a ccount their spacing
hypotenuse = (nTx-1)*lambda/2;
a = hypotenuse/sqrt(2); %intersect points along x and y
for t = 1:nTx
```

```
  %intersect of first antennna will be (0,a) then continue
down from
  %there
  txLocationActual(t,1) = (t-1)*a/nTx;
  txLocationActual(t,2) = a - (t-1)*a/nTx;
end

%Calculate distance to middle of cell for each transmitter
medianXY = median(map(2:gridSize,4:5));
medianDistance = zeros(nTx,1);
for t = 1:nTx
  medianDistance(t) = sqrt((medianXY(1)-
txLocationActual(t,1))^2 +...
      (medianXY(2)-txLocationActual(t,2))^2);
end
clear medianXY;
%% Set up training database
nIterations = 10000; %DATABASE SIZE (HOW MANY TIMES WE DO A
CHANNEL RESET)
trainingDatabase = cell(1,nIterations); %database contains
all possible number of users
lambda = 4;
p = poissrnd(lambda,1,1e6);
txTrack = findLocation(map,txX,txY);
parfor iteration = 1:nIterations
  idx = 1;
  for rxLoc = 1:gridSize %Go through a realization of the
channel
      % for each antenna... differences
      if rxLoc ~= txTrack
          rayleighH = cell(nTx,1);
          ricianH = zeros(nTx,1);
          phaseout = ricianH;
          channelEstimate = zeros(nTx,1);
          for i = 1:nTx
              [ricianH(i), ~,L,~] =
RicianTap([reflectorX,reflectorY],...
                  txLocationActual(i,:),
map(rxLoc,4:5),medianDistance(i),txPower);
              rayleighH{i} =  (L^2/K)*RayleighTaps(30,p);
              channelEstimate(i) =
sum(cell2mat(rayleighH(i)))+...
                  ricianH(i);
          end
          %grid location
```

```
            trainingDatabase{1,iteration}{idx,1} =
map(rxLoc,1);
            trainingDatabase{1,iteration}{idx,2} =
channelEstimate;
            %channel estimate for each individualized channel
            idx = idx + 1;
        end
    end
end

vars_clear = {'idx','iteration'};
clear(vars_clear{:});
trainingAvgH = cell(gridSize-1,2);
for i = 1:gridSize-1
    temp1 = zeros(nTx,1); %average rayleigh for each
transmitter
    for j = 1:size(trainingDatabase,2)
        for k = 1:nTx
            new1 = trainingDatabase{j}{i,2}(k);%channel
estimate
            temp1(k) = temp1(k) + new1;
        end
    end
    trainingAvgH{i,1} = cell2mat(trainingDatabase{j}(i,1));
%locations
    trainingAvgH{i,2} = (temp1)/size(trainingDatabase,2);
%averages
end
clear temp1; clear temp2; clear new1; clear new2;
%% set up training H matrices, based on number of users
nUsers = 5;
% if nUsers > nTx
%     error('Number of User antennae must be less than
Transmitter');
% end
[C, nCombos] =
LocationCombos(txLocation,nUsers,[gridX,gridY],squareSize);
temp = trainingDatabase{1,1}; clear trainingDatabase;
databaseLocations = cell2mat(temp(:,1));
trainingH = cell(nCombos,2); %1: location, 2: H matrix
for combo = 1:nCombos
    %set the user combination
    uLocations = C(combo,:);
    trainingH{combo,1} = uLocations;
    %combination
    H = zeros(nUsers,nTx);
```

```
  % Do training Database first
  for j = 1:nUsers
      location = uLocations(j);
      index = find(location == databaseLocations);
      H(j,:) = cell2mat(trainingAvgH(index,2));
  end
  trainingH{combo, 2} = H;
end
clear trainingAvgH; clear combo;
fprintf('Database Success!\n')
%% set up actual user locations (randomly chosen)
%Graph information
totalSimulations = 500;
Saved_Mean_Averages = cell(totalSimulations,1); %1-nUsers:
user locations
Success_Count = cell(totalSimulations,1);
% waitstring = ['running Simulations'];
% wait = waitbar(0,waitstring);
parfor simulation = 1:totalSimulations
  userLocations = zeros(nUsers,3); %1: map location, 2-3:
actual x and y
  %1: Get random user location(S)
  for user = 1:nUsers
      userLocations(user,1) = randi([2,gridSize]);
      %check that users are not in same location
      for m = 1:nUsers-1
          %skip yourself
          if m == user
              continue
          end
          %check that users aren't in same locations
          while userLocations(user,1) == userLocations(m,1)
              userLocations(user,1) = randi([2,gridSize]);
          end
      end

      %Get x and y locations
      userLocations(user,2) = map(userLocations(user,1),2)-
1 +...
          randi(99)/100;
      userLocations(user,3) = map(userLocations(user,1),3)-
1 +...
          randi(99)/100;
  end
  [~,I] = sort(userLocations(:,1),'ascend');
  userLocationsSorted = userLocations(I,:);
```

```
  userLocations = userLocationsSorted;

  %2: Generate new realizations
  nRealizations = 250;
  ChannelRealizations = cell(1,nRealizations); %store each
new H realization
  ErrorDistance = zeros(nRealizations,nUsers); %row:
realizations, column: user #
  sumError = zeros(1,nUsers);
  runningAverage = zeros(nRealizations,nUsers); %row is the
average mean error, column is the user number
  temp_success_count = 0;

  for r = 1:nRealizations
      H = zeros(nUsers,nTx);
      Norms = zeros(nCombos,1); %store norm for each
      %1: generate channel realization
      for user = 1:nUsers
          %channel coefficients from each transmitter to
each user
          rayleighH = cell(nTx,1);
          ricianH = zeros(nTx,1);
          phaseout = ricianH;
          channelEstimate = zeros(nTx,1);
          for tx = 1:nTx
              [ricianH(tx), ~,L,~] =
RicianTap([reflectorX,reflectorY],...
                  txLocationActual(tx,:),
userLocations(user,2:3), medianDistance(tx),txPower);
              channelEstimate(tx) =
sum(cell2mat(rayleighH(tx)))+...
                  ricianH(tx);
              rayleighH{tx} = (L^2/K)*RayleighTaps(30,p);
          end
          H(user,:) = channelEstimate;
      end
      ChannelRealizations{r} = H;

      %2: compare realization to each entry in training
database
      for c = 1:nCombos
          trainH = cell2mat(trainingH(c,2));
          Norms(c) = frobenius(pinv(trainH),pinv(H));
      end
      %3: Find location combo H with lowest norm
      locationIdx = find(Norms == min(Norms));
```

```
        %4: Get actual locations from each
        LocationEstimate = C(locationIdx,:);

        %5: calculate error distance for each user
        for n = 1:nUsers
            userActual = userLocations(n,2:3);
            estimateActual = map(LocationEstimate(n),4:5);
            ErrorDistance(r,n) = sqrt((userActual(1)-
estimateActual(1))^2 +...
                (userActual(2) - estimateActual(2))^2);
            sumError(n) = sumError(n) + ErrorDistance(r,n);
        end

        if mean(ErrorDistance(r,:)) < squareSize*sqrt(2)/2
            temp_success_count = temp_success_count+1;
        end

        %6: keep a running average of error distances
        runningAverage(r,:) = sumError/r;
    end
    Success_Count{simulation} = temp_success_count;
    Saved_Locations{simulation} = userLocations;
    Saved_Mean_Averages{simulation} =
mean(runningAverage(nRealizations/2:nRealizations,:));
end
clear trainingH;
fprintf('Took %0.0f seconds. \n',toc)
save('Results_7x7_1m2_2Tx_5User.mat');

function [H,phase,L,equalFlag] =
RicianTap(reflectorLocation,txLocation,...
    userLocation,medianDistance,txPower)
%All location inputs are in terms of x and y
fc = 30e9;
c = 3e8;
wavelength = c/fc;
lossExp = 3;
equalFlag = 0;
refPower = (1/(4*pi));
medianSNR = 100; %based off of 20 dB for median
txPower = txPower;
%Calculate distances of the bounce path and the direct line
of sight
if (reflectorLocation(1) ~= userLocation(1)) ||
(reflectorLocation(2) ~= userLocation(2))
```

```
  %Acquire distances
  distTx2Reflector = sqrt((txLocation(1) -
reflectorLocation(1))^2 +...
      (txLocation(2) - reflectorLocation(2))^2);
  distReflector2User = sqrt((userLocation(1)-
reflectorLocation(1))^2 +...
      (userLocation(2) - reflectorLocation(2))^2);
  bouncePath = distTx2Reflector + distReflector2User;
  directPath = sqrt((userLocation(1) - txLocation(1))^2
+...
      (userLocation(2) - txLocation(2))^2);
  %Acquire SNR relative to median distance
  if directPath == medianDistance
      SNR = medianSNR;
  elseif directPath < medianDistance
      SNRScale = ((medianDistance-
directPath)/directPath)+1;
      SNR = SNRScale*medianSNR;
  else
      SNRScale = ((directPath-medianDistance)/directPath);
      SNR = SNRScale*medianSNR;
  end

  directPhase = 2*pi*(directPath/wavelength -
floor(directPath/wavelength));
  bouncePhase = 2*pi*(bouncePath/wavelength -
floor(bouncePath/wavelength));

  Ldirect = refPower * (1/directPath)^lossExp;
  Lbounce = refPower * (1/bouncePath)^lossExp;

  rxPowerDirect = txPower * Ldirect;
  rxPowerBounce = txPower * Lbounce;

  noisePower = rxPowerDirect/SNR; %get the variance
  var = noisePower;
  %     Lbounce = 2;
  %     Ldirect = 1;
  meanDirect = sqrt(rxPowerDirect)*exp(1i*directPhase);
  meanBounce = sqrt(rxPowerBounce)*exp(1i*bouncePhase);

  HDirect = var*((randn + 1i*randn))+ meanDirect;
  HBounce = var*((randn + 1i*randn))+ meanBounce;
  H = HDirect + HBounce;

  pathDiff = bouncePath - directPath;
```

```
  %     phaseDiff = 2*pi*(pathDiff/wavelength -
floor(pathDiff/wavelength));
  L = abs(meanBounce+meanDirect);
  phase = angle(meanBounce+meanDirect);
  if pathDiff == 0
      equalFlag = 1;
  end
else
  directPath = sqrt((userLocation(1) - txLocation(1))^2
+...
      (userLocation(2) - txLocation(2))^2);

  %Acquire SNR relative to median distance
  if directPath == medianDistance
      SNR = medianSNR;
  elseif directPath < medianDistance
      SNRScale = ((medianDistance-
directPath)/directPath)+1;
      SNR = SNRScale*medianSNR;
  else
      SNRScale = ((directPath-medianDistance)/directPath);
      SNR = SNRScale*medianSNR;
  end

  directPhase = 2*pi*(directPath/wavelength-
floor(directPath/wavelength));
  Ldirect = refPower * (1/directPath)^lossExp;
  rxPowerDirect = txPower * Ldirect;
  noisePower = rxPowerDirect/SNR; %get the variance
  var = noisePower;
  meanDirect = rxPowerDirect*exp(1i*directPhase);
  L = rxPowerDirect;
  HDirect = var*((randn + 1i*randn))+ meanDirect;
  H = HDirect;
  phase = directPhase;
end
end

function [M] = Map(Grid, square)
%Take in Grid x and y size
%generate single value to match each x and y location
GridSize = Grid(1)*Grid(2);
M = zeros(GridSize,5); %Map locations to grid
counter = 1;
for y = 0:Grid(2)-1
  for x = 0:Grid(1)-1
```

```
        xactual = x*square + 1/2*square;
        yactual = y*square + 1/2*square;
        M(counter,:) = [counter,x+1,y+1,xactual,yactual];
        counter = counter +1;
    end
end
end
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Q. H. Spencer, C. B. Peel, A. L. Swindlehurst, and M. Haardt, "An introduction to the multi-user MIMO downlink," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 60–67, Oct. 2004, https://doi.org/10.1109/MCOM.2004.1341262.

[2] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," *39th Annual IEEE Conference on Local Computer Networks*, Sep. 2014, pp. 202–209. https://doi.org/10.1109/LCN.2014.6925773.

[3] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," in *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017, https://doi.org/10.1109/TVT.2016.2545523.

[4] X. Sun, X. Gao, G. Y. Li, and W. Han, "Fingerprint Based Single-Site Localization for Massive MIMO-OFDM Systems," *IEEE Global Communications Conference*, Dec. 2017, pp. 1–7. https://doi.org/10.1109/GLOCOM.2017.8254873.

[5] H. Liu, H. Darabi, P. Banerjee and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," in I*EEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007, https://doi.org/10.1109/TSMCC.2007.905750.

[6] R. L. Haupt, *Wireless Communications Systems: An Introduction*. Newark, NJ, USA: John Wiley & Sons, Incorporated, 2019.

[7] T. Koike-Akino, P. Wang, M. Pajovic, H. Sun and P. V. Orlik, "Fingerprinting-Based Indoor Localization With Commercial MMWave WiFi: A Deep Learning Approach," in *IEEE Access*, vol. 8, pp. 84879-84892, 2020, https://doi.org/10.1109/ACCESS.2020.2991129.

[8] Z. Wei, Y. Zhao, X. Liu and Z. Feng, "DoA-LF: A Location Fingerprint Positioning Algorithm With Millimeter-Wave," in *IEEE Access*, vol. 5, pp. 22678-22688, 2017, https://doi.org/10.1109/ACCESS.2017.2753781.

[9] P. 27 J. 2017 | 19:00 GMT, "Everything You Need to Know About 5G - IEEE Spectrum," *IEEE Spectrum: Technology, Engineering, and Science News*. https://spectrum.ieee.org/video/telecom/wireless/everything-you-need-to-know-about-5g.

[10] E. G. Larsson, O. Edfors, F. Tufvesson and T. L. Marzetta, "Massive MIMO for next generation wireless systems," in *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, February 2014, https://doi.org/10.1109/MCOM.2014.6736761.

[11] M. C. Jeruchim, *Simulation of communication systems*. New York, NY, USA: Plenum, 1992.

[12] Q. H. Spencer, A. L. Swindlehurst and M. Haardt, “Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels,” in *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 461–471, Feb. 2004, https://doi.org/10.1109/TSP.2003.821107.

[13] W. C. Jakes, Ed., *Microwave mobile communications*, Nachdr. New York, NY: IEEE Press, 1995.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California