THE PENNSYLANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ELECTRICAL ENGINEERING

MIMO WIRELESS AD-HOC COMMUNICATION NETWORKS: EVALUATION OF PERFORMANCE

MICHAEL WANG
Spring 2011

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Electrical Engineering and Physics
with honors in Electrical Engineering

Reviewed and approved* by the following:

Aylin Yener
Professor of Electrical Engineering
Thesis Supervisor

John D. Mitchell
Professor of Electrical Engineering
Honors Advisor

* Signatures are on file in the Schreyer Honors College.

Abstract

The relatively limited resources, e.g. frequency and power, of wireless communication, necessitate the development of techniques designed to optimize the use of resources for the existing demand. Employing multiple antennas has recently emerged as a viable design choice, as this has the potential to improve the communication rates without having to use more frequency resources. The resulting Multiple-Input-Multiple-Output (MIMO) wireless communication networks need to account for many factors such as shared resources and power constraints, which complicate the design of such networks. Many physical-layer techniques have been developed to counter the effects of interference and noise and to account for power limitations, in order to optimize the performance of MIMO wireless networks under certain conditions.

In this thesis, we explore the application of several physical-layer techniques to improve the performance of MIMO wireless ad-hoc communication networks under realistic operating parameters. A computer model is developed in MATLAB to generate an arbitrary system topology and simulate its performance under a variety of optimization techniques. Performance results are compared for different optimization techniques.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Professor Aylin Yener, my thesis advisor and instructor for several classes, for taking me on as an undergraduate member of the Wireless Communications and Networking Laboratory at the Pennsylvania State University (WCAN@PSU) in the spring of 2010, and deciding that I was worth keeping. Thank you for the initial inspiration to begin this project, and for the continued support to see it through. I appreciate all of the time you have invested into me, and I hope I can demonstrate that it was time well-spent.

I would like to thank the members of WCAN@PSU, for their help in the development of this project. Thank you for listening to my ideas and for helping to point me in the right direction.

I would also like to give special thanks to Semih Serbetli, whose previous graduate work on designing linear transmitters for MIMO systems provided the foundation for this project.

I would like to thank Professor John D. Mitchell, of the Department of Electrical Engineering at Penn State, my honors advisor for this thesis and teacher of my first course in this major. Thank you for your guidance and your instruction.

I would also like to thank Professor Richard W. Robinett, of the Department of Physics at Penn State, my honors advisor and instructor for PHYS 444. Thank you for the advice, guidance, support and inspiration you have given me over these past four years at Penn State.

Finally, I would like to thank my parents, Shiehlie and Hsiu-Mei Wang, for making this work possible. Thank you for your continued support and encouragement throughout my education.

1. Introduction

1.1 Motivation

The wireless “vision” for the future consists of the wireless networks such as cellular telephony, wireless Local Area Networks, wide area wireless data systems, satellite systems, and ad hoc wireless networks, and the applications that they run [1]. In this concept, future users can use wireless networks to transmit and receive their desired data, from any location they choose—all from a small portable device. Many of these wireless systems already exist today: cellular networks enable wireless voice communication, wireless LANs allow expedient file transfers from one user to another, and satellite systems even allow for high-definition multimedia to be beamed around the world.

However, due to the sharply rising demand for mobile computing and multimedia applications, future wireless systems need to be more efficient and outperform current wireless networks. This need is complicated by the many technical challenges that face wireless system development: challenges such as the need for end-user access terminals with low power consumption, the need to operate within limited spectrum bands, and the intrinsic non-deterministic nature of the wireless channel itself.

Many wireless network architectures have been designed to counter these technical challenges. Several network architectures, such as cellular telephony, have shifted the majority of the power-intensive processing work to centralized base stations, in order to increase the portability and lifetime of end-user access terminals.

A major issue that arises from these base-station-centric architectures is that their structure must be predetermined and relatively static. This centralized architecture is fragile: base stations are not guaranteed to remain functional after major disasters, require much time and effort to construct and deploy, and may be too cost-inefficient to be used in providing coverage to sparsely populated regions.

The alternatives to wireless communications and to cellular are wireless ad-hoc networks. These architectures are ad-hoc: all end-user access terminals dynamically configure themselves into a decentralized wireless network, capable of compensating for unexpected outages and dynamic changes in the network topology, or structure. Furthermore, equipping each of the terminals in the networks with multiple antennas allows each transceiver to exploit wireless channel properties to increase network performance. Although this alternative holds much promise, relatively little research has been done on the performance of MIMO wireless networks, primarily due to the innately arbitrary topology of the network.

1.2 Objective

The purpose of this work is to apply optimization techniques to improve the performance of MIMO wireless ad-hoc networks. We first determine the performance of a baseline MIMO wireless ad-hoc network through simulation. We then introduce a variety of techniques used in MIMO wireless communication networks into the network architecture to alter the network performance. After that, we determine the performance of each optimization technique, and any gain in performance due to its application. Finally, we discuss possible techniques to be attempted in the future.

1.3 Thesis Organization

The thesis is organized as follows. Chapter Two focuses on the literature review. We first discuss the elements of a communication system, and the types of distortions that may arise in the channel model that can alter our transmitted signal. We then briefly discuss the Single-Input-Single-Output (SISO) model, and its vulnerability to symbol corruption. Next, we discuss the MIMO model, and its advantages and disadvantages over the SISO model. We then elaborate on the concept of multiple access channels in the MIMO domain, and discuss various methods used to optimize MIMO multiple access communication networks.

Chapter Three focuses on the system design for this project. We describe the main objective of the project: to improve the performance of MIMO wireless ad-hoc communication networks. We then mention the six main assumptions that we make in describing our system model, and their consequences. We then present the basic model of our system, and describe the performance metric we use to compare optimization techniques. Finally, we present mathematical descriptions of how the system model functions with and without various optimization techniques.

Chapter Four qualitatively compares the performance results of the various MIMO wireless ad-hoc communication network implementations—both with and without the use of optimization techniques. The performance is analyzed for both boundary values of the log-distance path loss exponent to simulate network operation in the extreme cases. Sample data from the simulations is given.

Chapter Five forms several conclusions about our work, and explores potential avenues for future research in this direction.

2. Background and Literature Review

In this chapter, we will define the basic components of a communication system, detail the Single-Input-Single-Output and Multiple-Input-Multiple-Output models, and discuss several optimization techniques for multiple access MIMO networks.

2.1 The Communication System

A communication system is an arrangement of individual transmission systems, communication networks, relay stations and receiver equipment that combine to form an integrated and coordinated structure that allows information to be quickly and accurately transferred from an individual or group of users to another individual or group of users.

In this thesis, we will consider the communication system to be as shown in Fig. 1. A communication system consists of these four major components: the information source, the transmitter, the channel, and the receiver.

1. The *source coder* is the component that generates the data to be transferred, known as the message. This data can be composed of many things: it can be voice communication in the case of a cellular phone, computer files in the case of file transfers, or even multimedia in the case of cable or satellite television.
2. The *transmitter* takes in the message, and maps it to a form suitable to be transmitted over the channel.
3. The *communication channel* is the physical layer or medium (such as telephone wires, optical cables, and radio frequency bands) over which the signal is sent.
4. The *receiver* takes in the received signal and processes it, in order to obtain the message estimate, or best guess at what the original message was.

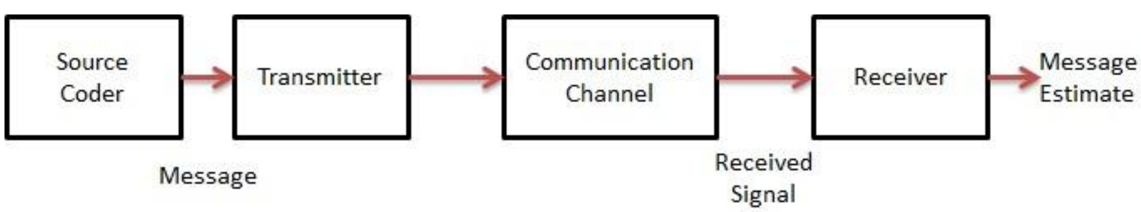

Fig. 1—Schematic diagram of a general communication network [2]

In this thesis, we will be focusing on the transmitter, communication channel and receiver elements in this model.

Over the course of transmitting the message, distortions may be introduced into the signal. These distortions may alter the received signal such that the inversion operation performed by the receiver on the signal results in a received message that differs from the original message, resulting in an error in communication. These distortions come in three major categories: channel distortion, noise and interference.

1. *Channel distortion* is a passive, physical property of the communication channel. As the signal passes through the channel, it is deformed by the non-ideal nature of the channel: frequency-dependent attenuation, multipath effects and Doppler shifts all fall into this category.
2. *Noise* is a passive, physical phenomenon that occurs at the receiver. Due to thermodynamic perturbations in charge carriers and conductors in the receiver, chaotic electromagnetic effects arise and are added to the received signal.
3. *Interference* is an active, physical phenomenon that occurs when external sources introduce randomness into the channel and alter the message. Interference can result from other communication systems transmitting signals over nearby channels, induction effects from power lines, and even natural events such as nearby thunderstorms and solar radiation.

In digital communications, the transmitted signal is said to be composed of an ordered sequence of symbols, which are each chosen from a finite set of possible symbols. When the signal is transmitted, depending on the system configuration, symbols are either sent individually, or in vectors over the channel. The received symbols are then reconfigured into their original order to form the received signal, which is then used to reconstruct the original message [1].

Most communication systems have some type of feedback, to allow the transmitter and receiver to determine the Channel State Information, or CSI. CSI, or the knowledge of channel properties such as gain, multipath effects, and path loss, allows the transmitter to adapt the sent signal to the channel, increasing the performance of the communication system [1].

2.2 The Single-Input-Single-Output Model

The simplest example of a communication system is the Single-Input-Single-Output (SISO) model. This model is characterized by the fact that both the transmitter and the receiver each have only one antenna or one wire connection, resulting in a communication link with a single input and a single output of a single symbol per transmission period through the channel, as shown in Fig. 2.

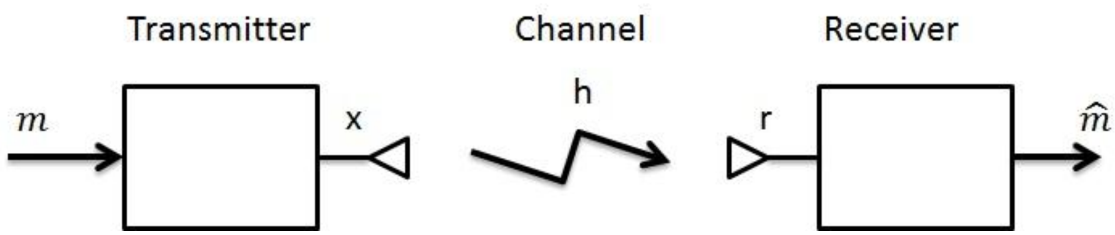

Fig. 2—Single-Input-Single-Output communication system model

It is relatively simple to analyze the performance of a SISO system. First, let us assume that the channel is slowly varying and exhibits flat fading. Slow varying assumes that the coherence time of the channel is large compared to the time to transmit and receive the message, so the channel is roughly constant over the entire transmission. We also assume flat fading: that the coherence bandwidth of the channel is large enough so that all signal components, even those at different frequencies, will encounter the same magnitude of fading [1].

Let x represent a single symbol sent from the transmitter and n be the noise at the receiver. h is the complex channel gain due to channel distortion, and we assume no interference for simplicity. The resulting received symbol r is

$$r = hx + n. \tag{1}$$

By reusing the SISO system repeatedly to transmit consecutive symbols of a given message, the receiver can reconstruct a close representation of the original message by calculating and concatenating the received symbols.

SISO systems, however, are not immune to noise and system outages—if the noise randomly spikes, or if the channel happens to reduce the signal amplitude such that the noise becomes the dominant component in (1), the reconstructed message would differ from the original, resulting in a corrupted symbol. SISO systems are especially vulnerable to sudden changes in the channel because it only sends data over a single link each transmission period.

2.3 The Multiple-Input-Multiple-Output Model

A more complex scheme for communication systems is the Multiple-Input-Multiple-Output (MIMO) model. The primary difference between the MIMO and the SISO model is the additional antennas that each MIMO transmitter or receiver has, as shown in Fig. 3.

Although the presence of additional antennas at both ends of the communication system increases the computational complexity of (1) and requires the parsing of the original message into several equal-length signals to be transmitted from each antenna, these antennas give rise to several significant advantages over SISO systems.

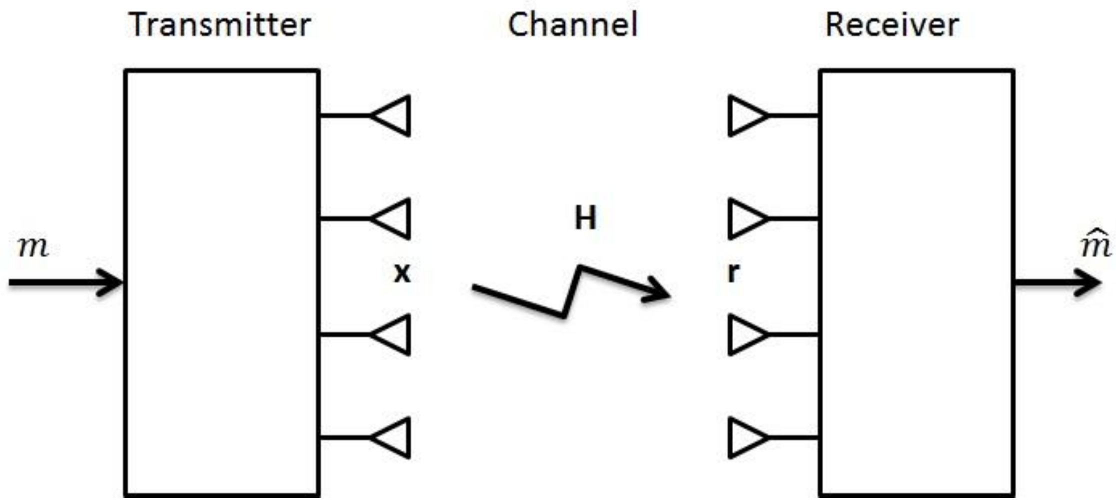

Fig. 3—Multiple-Input-Multiple-Output communication system model

MIMO systems compensate for the susceptibility of SISO systems to noise and outages by introducing the use of multiple antennas for both transmission and reception of signals. By placing the individual antennas at least half of a wavelength of the message transmitted over each antenna, the parallel channels that form between the transmitter and receiver antennas can be considered virtually independent of each other [1]. These channels can then be exploited by MIMO networks to improve overall system performance.

Using transmitters with multiple antennas allow the system to take advantage of three different transmission techniques called spatial multiplexing, transmitter precoding, and diversity coding, which improves the ability of the MIMO system to convey information [1].

Spatial multiplexing is a technique that takes advantage of multiple antennas at both the transmitter and the receiver to improve the resilience of the MIMO system to noise and channel distortions. In spatial multiplexing, a single signal at a high bit-rate is parsed into several lower bit-rate signals, which are then separately transmitted from each of the transmit antennas [3]. Each of these signals can then be considered to have passed through independent channels by the receiver, resulting in higher information capacity [4] at higher signal-to-noise ratios.

Transmitter *precoding* is another technique that improves the MIMO system performance compared to SISO, by taking advantage of multiple antennas for beamforming and spatial processing. By making use of the known CSI at the transmitter, the transmitter is able to process and transmit the signal in such a way that the signal power is maximized at the receiver, which results in reduced multipath fading and increased received signal power seen in [5].

Diversity coding is a MIMO technique used when there is little or nothing known about the CSI at the transmitter end. In order to optimize transmission in the absence of information about channel properties, the signal is sent from all of the transmit antennas with some form of orthogonal coding as in [6], with the most famous examples being Alamouti coding [7] for two antennas, Bell Labs' V-BLAST [8] and Papadias and Foschini's four transmit antenna scheme [9]. Diversity coding takes advantage of the independent nature of the parallel channels to improve system resilience to noise, channel distortions and imperfections, and enhance signal diversity at the receiver [6], resulting in improved system performance.

2.3.1. The Multiple Access MIMO Model

In wireless ad-hoc networks, there exist many different end-user access terminals, each with their own messages to send over the same medium. It is clear that transmitting only one message at a time from one user to another is inefficient and requires more complexity in determining which user can use the channel at any one time, and involves needless delay. It is easy to see that the solution would involve the sharing of channel resources, with many users simultaneously transmitting their messages over the channel when they want.

Since the receiver receives the sum of all transmitted messages scaled by the gain matrix from each transmitter to the receiver plus noise, the received signal at the receiver changes from (1) to the following. Let M be the total number of transmitters, N_t be the number of antennas per transmitter, N_r be the number of antennas at the receiver, $\boldsymbol{H_i}$ be a N_r-by-N_t matrix describing the channel gain between the i^{th} transmitter and the receiver, $\boldsymbol{x_i}$ be the N_t-by-1 symbol vector being transmitted from the antennas of the i^{th} transmitter during the current transmission cycle, $\boldsymbol{n}$ be the N_r-by-1 noise vector at the receiver. The received signal at the receiver, $\boldsymbol{r}$, is found using the following equation:

$$\boldsymbol{r} = \sum_{i=1}^{M}(\boldsymbol{H_i}\boldsymbol{x_i}) + \boldsymbol{n} \tag{2}$$

Although the individual transmitters and receiver are identical to those of Fig. 3, it is the presence of multiple transmitters that characterize the Multiple Access MIMO model. Fig. 4 shows a cellular phone network, a type of multiple access MIMO network. Each user sends voice communication (the messages m_i encoded in the associated $\boldsymbol{x_i}$ vector) across wireless channels (channel gain matrices $\boldsymbol{H_i}$), to the cell phone tower (the base station, or receiver), which then can generate the message estimates $\widehat{m}_1, \widehat{m}_2, \dots, \widehat{m}_M$ based on the total received signal, $\boldsymbol{r}$.

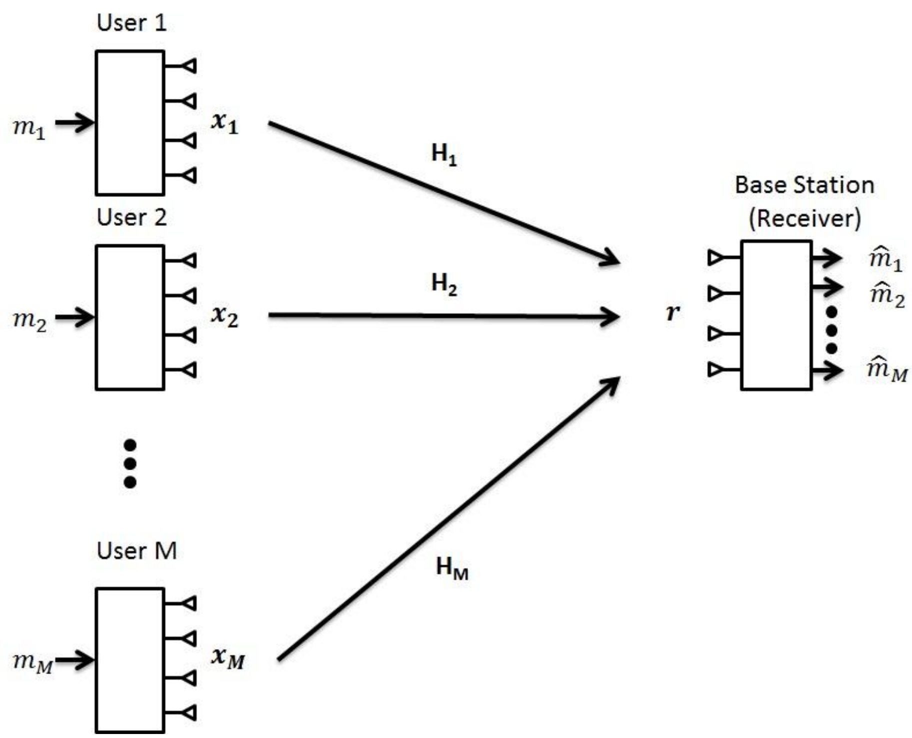

Fig. 4—Cellular phone multiple access MIMO model

2.3.2 The Ad-Hoc MIMO Model

Of particular interest is the ad-hoc MIMO communication network model. In this model, there exists no centralized architecture for users to rely upon to facilitate message transmission, nor is there a preset topology for all transmitting and receiving users. Instead, these systems possess the ability to self-organize into a network, optimize their performance, and reliably transmit and receive messages all in an unknown transmitter/receiver topology.

There are several unique issues that arise with the ad-hoc MIMO model due to the lack of predetermined or fixed transmitters or receivers. One major issue is that there is a significant transmission overhead associated with the setup, maintenance and determination of CSI for the channel between each pair of transmitters and receivers due to the lack of information about the network at the start. Since this transmission overhead is so large, it is simpler to treat all other signals at the receiver as noise, rather than using classical approaches that require interference cancellation [10]. In addition, optimization of the transmitters or receivers to reduce the system-wide mean-squared error is not equivalent to the common technique of optimization of individual mean-squared errors, seen in [11].

2.4. Multiple Access MIMO Channel Optimization

There exist many techniques to optimize multiple access MIMO channels. In this paper, we will focus on two primary techniques used to optimize MIMO systems with multiple transmitters and receivers: power allocation (water filling) and linear transceiver optimization to minimize mean-squared error.

2.4.1 Power Allocation: Water Filling

In a MIMO wireless system, the channel matrices (the set of all $\{\mathbf{H}\}$ in (2)) can be modeled with independent and identically distributed (i.i.d.) complex random variables. It is clear that we cannot expect that all channels to have equal channel gains over the entire transmission (recall that the antennas are placed at least half a wavelength apart to ensure independent channels), which begs the question of how we should allocate power to each antenna to optimize our transmission.

Furthermore, it is also clear that during some fraction of transmission cycles, a particular transmitter antenna may experience an unusually high loss of signal power due to natural and man-made fluctuations in the communication channel, making it highly unlikely that the receiver would be able to decode the message sent over that channel correctly. (For example, the case where the channel attenuated the signal to such a degree that the noise magnitude of the receiver is greater than or equal to that of the attenuated signal.) We are then faced with the question of how much power we should allocate to the antenna in question to render the signal understandable at the receiver end.

The solution to both of these questions is found in the technique of water-filling, studied in depth by Yu et. al. [12]. Water-filling is a technique that uses CSI at the transmitter to determine how to allocate power to the transmit antennas based on the transmitter's information on the state of the channel. The name "water-filling" comes from the analogy of pouring water to a constant level in a bowl, where the deeper portions represent good received signal quality and the shallower portions represent attenuated signals. The power allocated to each antenna (and each signal) is proportional to the height of the water in the bowl. The idea is to exploit those channels with little attenuation by allocating more power and transmitting at a higher bit-rate through them, while reducing power used by poorly performing antennas, possibly cutting power completely to an antenna if it has an extremely poor channel over that transmission period.

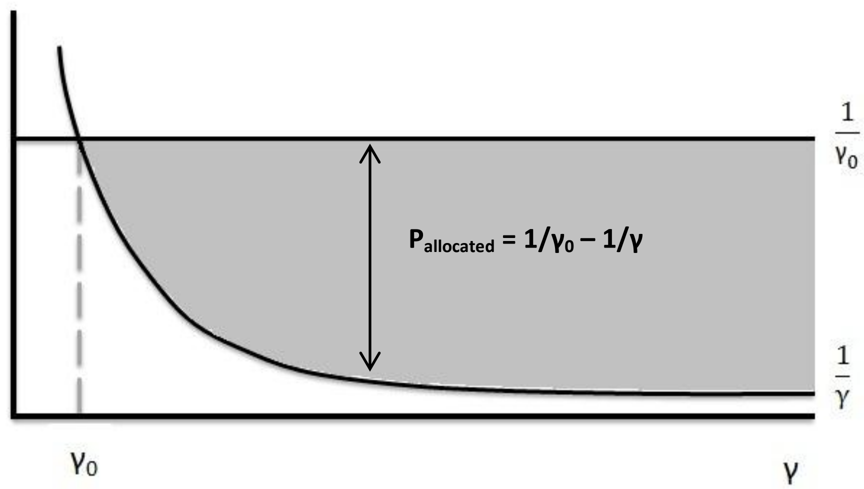

Fig. 5—Water-Filling technique for power allocation [1]

Fig. 5 illustrates the concept: by using CSI at the transmitter, we can determine a minimum signal-to-noise ratio (SNR) γ_0—as long as the noise-to-signal power ratio ($1/\gamma$) associated with a channel does not exceed a certain threshold, we will allocate power to that antenna (illustrated by the grey region). If the noise-to-signal power ratio exceeds our threshold ($1/\gamma_0$), no power will be allocated to that antenna. The exact equation for power allocation is given below [1]:

$$Power = \begin{cases} C\left(\frac{1}{\gamma_0} - \frac{1}{\gamma}\right) & for\, \gamma > \gamma_0 \\ 0 & for\, \gamma < \gamma_0 \end{cases} \tag{3}$$

where C is a normalization constant to ensure that the total power being allocated to all transmit antennas sum to the total transmit power of the transmitter.

2.4.2 Linear Transceiver Optimization

Another approach to optimizing multiple access MIMO systems is to improve performance by jointly optimizing the transceivers in the system, known as linear transceiver optimization. As shown in [11], we can do this by introducing linear transmitters (precoders) and receivers (decoders) into the communication system such that the mean-squared error (MSE) of the entire system is minimized: this is equivalent to optimizing the system to err less, both in magnitude and number of occurrences.

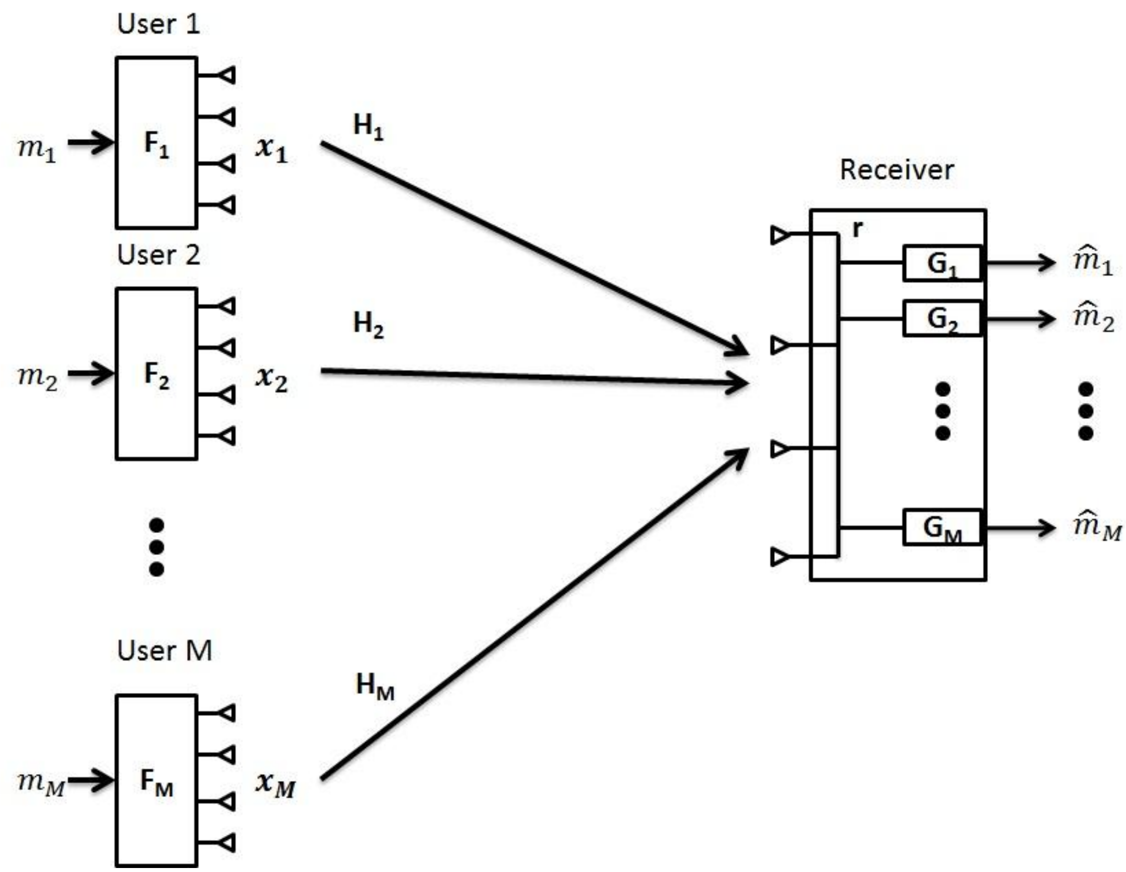

Fig. 6—Multiple access MIMO system with linear transceivers [11]

The multiple access MIMO system model for linear transceiver optimization is shown in Fig. 6, for a single receiver that outputs messages from each user. Each user or transmitter has a linear precoder $\boldsymbol{F_i}$, matched to a corresponding linear decoder $\boldsymbol{G_i}$. The set of all $\{\boldsymbol{F_i}, \boldsymbol{G_i}\}$ is then jointly optimized to minimize the MSE of the entire system.

The actual equation for the received $\boldsymbol{r}$ is similar to the expression developed in (2), with the added change of the linear precoder $\boldsymbol{F_i}$:

$$\boldsymbol{r} = \sum_{i=1}^{M}(\boldsymbol{H_i F_i x_i}) + \boldsymbol{n} \tag{4}$$

The message estimates $\hat{x}_i$ would be found by applying the linear decoder $\boldsymbol{G_i}$ to the received $\boldsymbol{r}$, and a decision function f, which maps the vector $\boldsymbol{G_i}\{\boldsymbol{r}\}$ to a scalar estimate of the original message:

$$\hat{m}_i = f(\boldsymbol{G_i}\{\boldsymbol{r}\}) = f\{\boldsymbol{G_i}\{\textstyle\sum_{i=1}^{M}(\boldsymbol{H_i F_i x_i}) + \boldsymbol{n}\}\} \tag{5}$$

For an ad-hoc MIMO system model, there are no explicit algebraic expressions that we can solve for the set of $\{\boldsymbol{F_i}^*, \boldsymbol{G_i}^*\}$, the set of optimized linear precoders and decoders that result in system-wide MSE minimization due to the variable topology of the ad-hoc system. However, iterative approaches have been developed [11] that allow the system to approach system-wide minimum MSE.

In order to implement linear transceiver optimization to improve system performance, we require reliable feedback from the receivers to the transmitters. The feedback would provide CSI at the transmitters, allowing them and the receivers to jointly optimize to a configuration to minimize mean-squared error.

3. System Design

In this chapter, we will discuss the basic MIMO ad-hoc wireless communication system model and two improved model variants using water-filling and linear transceiver optimization techniques.

3.1 Model Overview

The objective in this thesis was to adapt methods in optimizing non-ad-hoc MIMO wireless systems for use in improving the performance of MIMO wireless ad-hoc networks. In order to do that, we first needed to develop a system model that would accurately reflect the physical operating conditions of such a network. The following assumptions were used in the following model.

1. All transceivers have the same maximum power constraint P_{max} while transmitting.
2. There exists a rich scattering environment such that each channel matrix $\boldsymbol{H}$ is composed of i.i.d. complex Gaussian random variables with zero mean and variance σ^2.
3. The system can be treated as exhibiting a log-distance path loss model with path loss exponent γ, and is independent of the channel matrix $\boldsymbol{H}$.
4. The noise at the receiver is circularly complex Gaussian noise with covariance matrix $\sigma^2\mathbf{I}$, where σ^2 is the noise power and $\mathbf{I}$ is the identity matrix.
5. Channel State Information (CSI) is available to all transmitters and receivers in the network.
6. The number of users and the channel matrices are fixed over the course of the transmission.

The system model is as follows, shown below in Fig. 7.

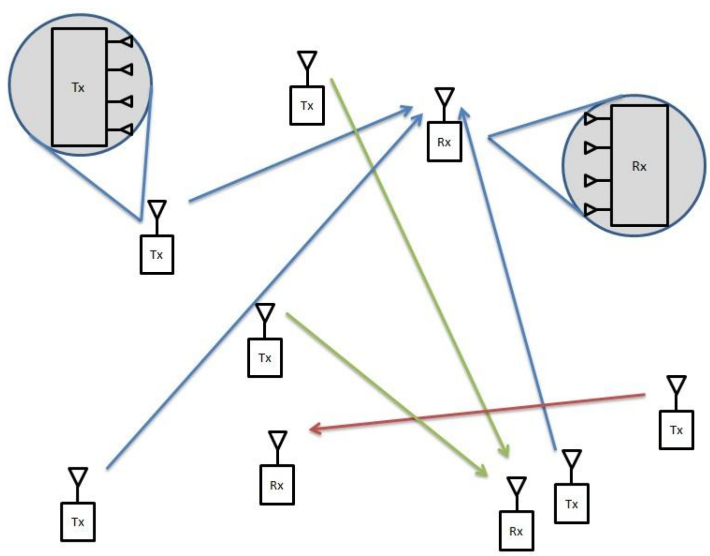

Fig. 7—MIMO wireless ad-hoc system model

The performance metric we use in this work to compare different optimization techniques is the bit-error rate vs. signal-to-noise ratio (BER vs SNR). Ideally we want to obtain a method to minimize system-wide BER vs SNR for all values of SNR, but because of the ad-hoc nature of the network topology, the scope of this work focuses on developing techniques to improve performance in terms of decreasing, rather than minimizing, BER vs SNR.

The idea behind using this model is to set up a randomly-generated topology of MIMO transceivers to model an ad-hoc MIMO multiple access network, and iteratively transmit single bits until a message of desired length has been transmitted. From the number of erroneous received messages and the total number of bits transmitted, we can then derive the BER for a particular SNR.

We first simulate the performance of a baseline scenario—that is, a MIMO wireless ad-hoc network without any optimization techniques. We then simulate and compare the performance of MIMO wireless ad-hoc networks with various optimization techniques intended for use with MIMO non-ad-hoc networks, and look for an improvement in the network performance, as measured by BER vs SNR, compared to the baseline scenario.

3.2 The Baseline Scenario: No Optimization

The computation of the BER for a specific SNR value using the standard baseline model proceeds as follows. We assume a fixed number of users and slow fading to enable us to reuse the channel over multiple transmission iterations. First, we set up the environment with the number of transceivers (*M*) by placing transceivers in a square two-dimensional area according to a uniform random distribution to model ad-hoc network topology. We then generate the set of all N_R-by-N_T channel matrices $\{\boldsymbol{H_{i,j}}\}$ and the set of all path loss coefficients associated each pair of transceivers.

To calculate the BER, we iterate over the following loop. First, we independently generate a message bit m_i from a Bernoulli distribution with equal probability for all of the non-receiving transceivers in the environment, and encode those bits into a signal bit b_i using Binary Phase Shift Keying (BPSK) as shown in (6):

$$b_i = \begin{cases} +1 & if\ m_i = 1 \\ -1 & if\ m_i = 0 \end{cases} \tag{6}$$

Next, we convert the signal bit b_i into a N_T-by-1 signal vector $\boldsymbol{x_i}$ to be transmitted by the N_T transmit antennas by the i^{th} transmitter, with equal power allocated to each antenna:

$$\boldsymbol{x_i} = \frac{b_i}{P_{max}} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} {b_i}/{P_{max}} \\ \vdots \\ {b_i}/{P_{max}} \end{bmatrix} \tag{7}$$

where P_{max} is the total transmit power allocated to the transmitter.

Next, we obtain the received signal $\boldsymbol{r_j}$ at the j^{th} receiver using the following:

$$\boldsymbol{r_j} = \sum_{i=1}^{M}\left[(l_{i,j})\boldsymbol{H_{i,j}x_i}\right] + \boldsymbol{n} \tag{8}$$

where $(l_{i,j})$ represents the log-distance path loss and $\boldsymbol{H_{i,j}}$ is the channel matrix between the i^{th} and j^{th} transceivers, and $\boldsymbol{n}$ is the noise vector at the receiver.

At the j^{th} receiver, a decoding function designed to decode the message from the i^{th} transmitter, $\boldsymbol{G_i}(\bullet)$**,** is applied to $\boldsymbol{r_j}$, which gives us the estimated vector transmitted by the i^{th} user, $\boldsymbol{\hat{x}_i}$. In the baseline case, we use the Hermitian of the channel matrix between the i^{th} and j^{th} transceivers, $\boldsymbol{H_{i,j}^{\dagger}}$:

$$\boldsymbol{\hat{x}_{i,j}} = \boldsymbol{G_i}(\boldsymbol{r_j}) = \boldsymbol{H_{i,j}^{\dagger}}(\boldsymbol{r_j}) = \boldsymbol{H_{i,j}^{\dagger}}\left(\sum_{i=1}^{M}\left[(l_{i,j})\boldsymbol{H_{i,j}x_i}\right] + \boldsymbol{n}\right) \tag{9}$$

$$= (l_{i,j})\left(\boldsymbol{H_{i,j}^{\dagger}H_{i,j}}\right)\boldsymbol{x_i} + \boldsymbol{z} \tag{10}$$

where $\bullet^{\dagger}$ represents the Hermitian operator and we have replaced the last two terms in (9) with the vector variable $\boldsymbol{z}$, which can be interpreted to mean we consider all other received transmissions as noise. Since matrix operations are linear, and $\boldsymbol{H_{i,j}}$, $\boldsymbol{H_{i,j}^{\dagger}}$ and $(l_{i,j})$ are deterministic over the duration of the simulation, we can treat $E[\boldsymbol{z}] = \boldsymbol{0}$: this means that, on average, the received signal $\boldsymbol{\hat{x}_i}$ at the j^{th} receiver consists of the first term, which preserves the sign of $\boldsymbol{x_i}$, ideally representing the same bit as the original message.

To obtain the message estimate, $\hat{m}_i$, from the i^{th} transmitter received by the j^{th} receiver, we look at the real part of the received signal vector $\boldsymbol{\hat{x}_{i,j}}$, and use a hard decision threshold at zero to determine what unit polar value each entry corresponds to:

$$\boldsymbol{\hat{b}_{i,j}}(k) = \begin{cases} +1 & if\ \boldsymbol{\hat{x}_{i,j}}(k) \geq 0 \\ -1 & if\ \boldsymbol{\hat{x}_{i,j}}(k) < 0 \end{cases}, \qquad k = 1,2, \dots N_T \tag{11}$$

where k is the index of the k^{th} entry in the vector. The actual message estimate $\hat{m}_i$ is determined by the majority of the BPSK-decoded entries:

$$\hat{m}_i = \begin{cases} 1 & if\ more\ positive\ entries\ of\ \boldsymbol{\hat{b}_{i,j}}\ than\ negative\ entries \\ 0 & if\ more\ negative\ entries\ of\ \boldsymbol{\hat{b}_{i,j}}\ than\ positive\ entries \\ \left(\begin{cases} 1\ if\ \overline{\boldsymbol{\hat{b}_{i,j}}} \geq 0 \\ 0\ if\ \overline{\boldsymbol{\hat{b}_{i,j}}} < 0 \end{cases}\right) & if\ same\ number\ of\ positive, negative\ entries\ in\ \boldsymbol{\hat{b}_{i,j}} \end{cases} \tag{12}$$

where the overbar ($\overline{\ \ }$) indicates the mean value of the entries of the vector. The last case is included to cover situations where the number of receiver antennas N_R is even and there are an equal number of positive and negative entries.

At the end of the loop, we do two things: first, we update the error counter by incrementing it every iteration through the loop in the manner described in (13). If the received message does not match the transmitted message:

$$numError = \begin{cases} (numError + 1) & if\ \widehat{m}_i \neq m_i \\ (numError) & if\ \widehat{m}_i \neq m_i \end{cases} \tag{13}$$

After we update the error counter, we then update the running average of the signal-to-noise ratio (SNR) value:

$$\boldsymbol{snr}(iterationIndex) = \frac{(iterationIndex-1)*\boldsymbol{snr}(iterationIndex-1)+\gamma}{iterationIndex} \tag{14}$$

where γ is the SNR of the current iteration.

After iterating through the loop a number of times equal to the number of bits in the message, the BER can be determined by dividing the total number of errors by the total number of bits transmitted—that is, dividing by the total message length:

$$BER = \frac{number\ of\ bit\ errors}{total\ message\ length} \tag{15}$$

We can then plot the BER as a function of SNR for this baseline scenario for MIMO ad-hoc wireless networks.

3.3 Water-filling Approach

One technique we applied to the MIMO wireless ad-hoc network in an attempt to improve the system performance is the water-filling power allocation scheme. The idea is that, rather than equally distributing power across all N_T transmitter antennas, we use the water-filling approach to allocate more power to the best antennas with the best channels, and reduce power to antennas with worse channels, thereby increasing the likelihood that the decoded message is equivalent to the original. The method to obtain the BER vs SNR for a MIMO wireless ad-hoc network using a water-filling power allocation scheme is very similar to the process outlined in Section 3.2, with one major exception.

In particular, equation (7) is changed in the following way. Instead of multiplying b_i by an N_T-by-1 column vector of ones, we multiply it by the amplitude vector $\sqrt{\boldsymbol{p_{i,j}}}$ for the channels between the i^{th} transmitter and the j^{th} receiver:

$$\boldsymbol{x_i} = b_i\sqrt{\boldsymbol{p_{i,j}}} \tag{16}$$

where $\boldsymbol{p_{i,j}}$ is the power allocation vector a N_T-by-1 column vector determined using the following algorithm.

In order to generate the power allocation vector $\boldsymbol{p_{i,j}}$, we first determine the set of eigenvalues $\{e_{i,j}\}$ of the matrix $\boldsymbol{H_{i,j}H_{i,j}^{\dagger}}$. We then iterate through the following loop starting with an iteration index (representing the number of used channels) $N_{channel} = N_T$.

First, we determine the reference value for power allocation, u:

$$u = \frac{P_{max}}{N_{channel}} + \frac{1}{N_{channel}} \sum_{all\ k} \frac{1}{e_{i,j}(k)} \quad (17)$$

where P_{max} is the total transmit power allotted to the transmitter's antennas, and $e_{i,j}(k)$ is the k^{th} eigenvalue of the matrix $\boldsymbol{H_{i,j}H_{i,j}^{\dagger}}$. We then assign a proportion of that power to each channel according to the water-filling scheme, and store it in the k^{th} entry of the $\boldsymbol{p_{i,j}}$ vector as shown below:

$$\boldsymbol{p_{i,j}}(k) = u - \frac{1}{e_{i,j}(k)} \quad (18)$$

We then check to ensure that, for all entries of $\boldsymbol{p_{i,j}}$, there are no negative values of power assigned to an antenna. If there is a negative-valued entry, we restart the loop, this time ignoring the antenna associated with that entry—that is, the iteration index decreases from $(N_{channel})$ to $(N_{channel} - 1)$ while ignoring the $e_{i,j}$ associated with that antenna, and setting $\boldsymbol{p_{i,j}} = 0$ for that antenna index.

The loop terminates once we are able to obtain a power allocation vector $\boldsymbol{p_{i,j}}$ with only nonnegative entries—this vector is the $\boldsymbol{p_{i,j}}$ used in (16) to modify the original (7) in our model.

The result of this is that the signal strength from the transmit antennas with power allocated to them is much stronger than the noise compared to signals sent using the system model in Section 3.2. That is, when we make our decision in (12), it is easier for the receiver to discriminate the original signal from the noise at the receiver and other transmissions.

3.4 Linear Transceiver Optimization Approach

Another technique we apply to the MIMO wireless ad-hoc model was linear transceiver optimization, as detailed in [11]. The aim of this approach is to use channel information to generate the optimum transmitters (precoders) and the optimum receivers (decoders) to improve the system performance.

To implement this approach, we do the following. First, we generate the environment in the same way as we did in Section 3.2. Next, we randomly partition the set of transceivers into two subsets, where we choose the receiver in each subset randomly. Each subset then uses the linear transceiver optimization approach to optimize their precoders and decoders independently of the other subset. The idea here is the receiver in each subset only cares about signals being transmitted within its own subset and not the other, so it only optimizes its decoder matrices for the transmitters within its own subset.

An illustration for this approach is shown in Fig. 8, where transceivers involved in blue transmissions are in one subset and transceivers involved in green transmissions are in the other subset:

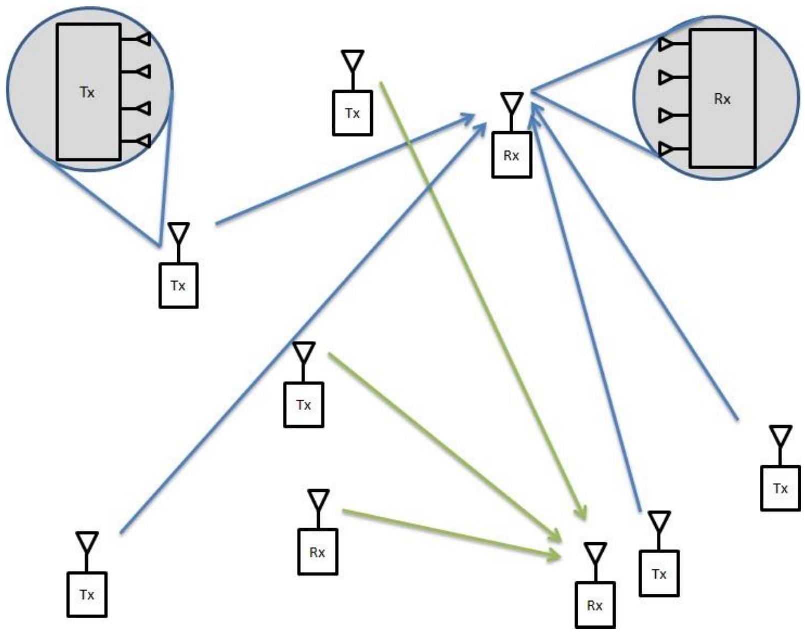

Fig. 8—Linear transceiver optimization implementation

In order to model system operation under linear transceiver optimization, we alter the received signal in (8) to include the transmitter precoder matrix, $\boldsymbol{F_i}$:

$$\boldsymbol{r_j} = \sum_{i=1}^{M}\left[(l_{i,j})\boldsymbol{H_{i,j}}\boldsymbol{F_i}\boldsymbol{x_i}\right] + \boldsymbol{n} \qquad j = 1,2 \qquad (19)$$

where $\boldsymbol{r_j}$ is the signal at the receiver in the j^{th} subset, $\boldsymbol{H_{i,j}}$ is the channel matrix between the i^{th} transmitter and the receiver of subset j, $\boldsymbol{x_i}$ is transmitted signal vector associated with the i^{th} transmitter, and $\boldsymbol{n}$ is the noise at the receiver.

The decoding step is also altered from (9). Instead of using $\boldsymbol{G_i} = \boldsymbol{H}_{i,j}^{\dagger}$, we use a decoder matrix $\boldsymbol{G_i}$ obtained from joint optimization with the precoder matrices to find the estimated signal vector $\boldsymbol{\hat{x}_i}$:

$$\boldsymbol{\hat{x}_i} = \boldsymbol{G_i}(\boldsymbol{r_j}) = \boldsymbol{G_i}\left(\sum_{i=1}^{M}\left[(l_{i,j})\boldsymbol{H_{i,j}}\boldsymbol{F_i}\boldsymbol{x_i}\right] + \boldsymbol{n}\right) \qquad j = 1,2; i \in j^{th}\ subset \quad (20)$$

In order to generate the set of $\{\boldsymbol{F_i}, \boldsymbol{G_i}\}$ for each subset, we follow the second algorithm outlined by Serbetli and Yener in [11] to solve the following minimization problem:

$$\min_{\{\boldsymbol{F_i},\boldsymbol{G_i}\}} \mathbf{MSE} \qquad i \in j^{th}\ subset \quad (21)$$

$$\text{s.t.} \quad \operatorname{tr}\left(\boldsymbol{F}_{\boldsymbol{i}}^{\dagger}\boldsymbol{F_i}\right) \le p_i \qquad i \in j^{th}\ subset \quad (22)$$

where **MSE** is the mean-squared error, p_i is the maximum power constraint for the *i*th transmitter, and the minimization is done over the set of transmitters {*i*} that belong to the *j*th subset.

However, in order to derive the Lagrangian multipliers $\{\mu_i\}$ used in the algorithm, we need to solve the following expression for each transmitter:

$$\sum_{k=1}^{M} \frac{c_{k,k}}{\{\mu_i + d_{k,k}\}^2} = p_i \qquad i \in j^{th}\ subset \quad (23)$$

where $\{c_{k,k}\}$ and $\{d_{k,k}\}$ are the (k,k) entries of the matrices $\boldsymbol{C_i}$ and $\boldsymbol{D_i}$ used in the algorithm, and p_i is the maximum power constraint of the *i*th transmitter. We solve this for the multipliers by using several properties [11]:

1. The maximum power constraint p_i is nonnegative.
2. The elements $\{c_{k,k}\}$ and $\{d_{k,k}\}$ are all nonnegative due to the positive semi-definiteness of matrices $\boldsymbol{C_i}$ and $\boldsymbol{D_i}$.
3. The Lagrangian multipliers $\{\mu_i\}$ are nonnegative.
4. $z(\mu_i)$, the sum on the left side of (23), is shown to be monotonically decreasing, and there exists only one nonnegative real value of μ_i that satisfies (23).
5. If no such solution $\mu_i \ge 0$ exists for (23) then $\mu_i = 0$.

From these properties and (23), we can write a new function of μ_i:

$$Z(\mu_i) = z(\mu_i) - p_i = \sum_{k=1}^{M} \frac{c_{k,k}}{\{\mu_i + d_{k,k}\}^2} - p_i = 0 \qquad (24)$$

We note that, since there is only one positive solution μ_i for which $Z(\mu_i)$, a monotonically decreasing function, equals zero, we use a binary search algorithm to quickly determine the solution to $Z(\mu_i) = 0$, and input it into the algorithm described in the paper. The performance of the approach is determined in the same way as in Section 3.2, by calculating the BER as a function of SNR.

4. Numerical Results

In this chapter, we discuss the performance results of our simulations, and display sample data from our simulations.

4.1 Performance Results

The three implementations of the MIMO ad-hoc wireless network detailed in Sections 3.2-4 were realized by using the MathWorks MATLAB© computing software (see Appendix A for MATLAB source code).

The parameters for all iterations for each MIMO ad-hoc wireless network implementation were as follows. There existed M = 10 transceivers in the environment, each with a maximum power constraint $P_{transmit}$ = 1, and N_T = N_T = 4 transmit/receive antennas. The total length of the message transmitted over each iteration was L = 10^5 bits in length. The log-distance path loss exponent γ was varied between 2 and 4, representing the two extremes of very lossy environments and free space propagation. Noise power was determined by the transmitter SNR specified for that iteration.

Sample network topologies for N = 10 transceivers can be seen in Fig. 9, located within a 100m-by-100m square environment. In Fig. 9a, the transmitter is located at node 7, and the receiver is located at node 10; in Fig. 9b, the transmitter is at node 1, and the receiver is at node 8.

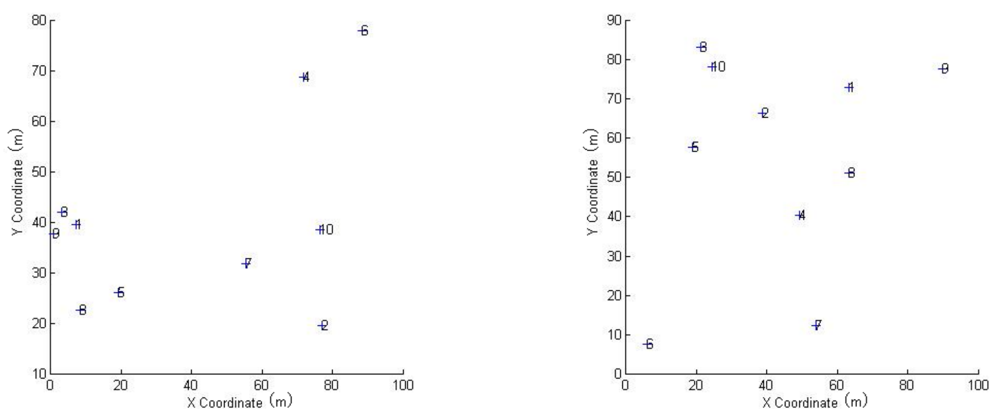

Fig. 9—Sample network topologies for N=10 transceiver nodes. (a) Transmitter: Node 7, Receiver: Node 10, (b) Transmitter: Node 1, Receiver: Node 8.

The complex gain matrix **H** between the four transmit and receive antennas for the sample topologies in Fig. 9 are given in Table 1, where **H**(7,10) corresponds to the topology in Fig. 9a, and **H**(1,8) corresponds to the topology in Fig. 9b. The i^{th}-by-j^{th} entries $\mathbf{H}_{i,j}$ correspond to the complex channel gains between the i^{th} transmit antenna and the j^{th} receive antenna.

Table 1: Sample complex gain matrices between transmitter/receiver nodes

$$\boldsymbol{H}(7{,}10) = \begin{bmatrix} +0.2300 - 0.3326i & -0.2239 + 0.3813i & -0.6856 - 0.9433i & -0.0748 - 0.4256i \\ +0.3897 - 1.0524i & -0.5217 - 1.1166i & -0.0343 - 0.1463i & -0.6359 - 0.6140i \\ +0.0701 + 0.7273i & -0.6403 - 0.0543i & -0.1870 - 0.6436i & +0.9554 - 1.1363i \\ -0.6181 + 0.4134i & -0.3638 - 0.2913i & -0.7574 + 0.1052i & -0.2952 + 0.8277i \end{bmatrix}$$

$$\boldsymbol{H}(1{,}8) = \begin{bmatrix} +0.6016 + 0.6690i & +0.0719 - 0.4659i & +0.0395 + 0.3626i & -0.1821 - 0.5125i \\ -0.4248 + 0.2864i & +0.6621 - 0.5275i & +0.0444 + 0.1979i & +0.9793 + 1.1924i \\ +0.4670 - 0.4885i & +0.2488 - 0.3037i & -0.1803 - 0.6156i & +0.5187 + 0.7050i \\ -0.9040 + 0.0829i & -0.2815 - 0.7200i & -0.0460 + 0.2395i & -0.4606 + 1.2972i \end{bmatrix}$$

A sample log-distance path loss matrix $(PL)_{7,10}$ is given in Table 2, corresponding to the network topology shown in Fig. 9a. The log-distance path loss values are given in dB for ease of access. The i^{th},j^{th} entry in the matrix corresponds to the path loss experienced between the i^{th} and j^{th} transceivers in the network. Due to the symmetry of the matrix (path loss between transceiver pairs (i,j) and (j,i) are identical) only the upper-triangular half of the values are stored, as shown.

Table 2: Sample power log-distance path loss matrix (in dB)

$$\begin{bmatrix} 0 & 74.4101 & 27.5226 & 74.0034 & 50.2104 & 78.1316 & 67.4734 & 49.0571 & 32.9177 & 73.6314 \\ 0 & 0 & 75.5031 & 67.7869 & 70.5758 & 70.9815 & 55.8275 & 73.4577 & 75.7291 & 51.1086 \\ 0 & 0 & 0 & 74.6856 & 54.2475 & 78.6557 & 69.0244 & 52.1020 & 27.7258 & 74.6610 \\ 0 & 0 & 0 & 0 & 73.2385 & 51.1841 & 64.3244 & 75.7559 & 75.5487 & 59.4654 \\ 0 & 0 & 0 & 0 & 0 & 77.4909 & 62.5044 & 42.0860 & 53.4587 & 70.7536 \\ 0 & 0 & 0 & 0 & 0 & 0 & 70.2006 & 79.5133 & 79.3584 & 64.5876 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 67.1467 & 69.5229 & 53.9784 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 48.9628 & 73.7964 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 75.1633 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 10 shows the evolution of mean BER for a specific network realization as a function of transmitter SNR. Over each iteration, the average BER of the overall system decreases exponentially until a minimal value is reached. The simulation results shown in Fig. 10 used a transmit power constraint $p_i \le 1$, and noise variance $\sigma^2 = 0.8$. The minimum mean BER value of 0.000043 was reached on iteration 15, well before the maximum iteration size of 20 that was used in the code.

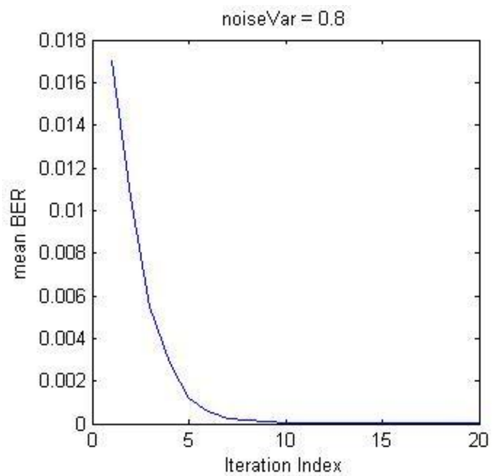

Fig. 10—Evolution of mean BER with transceiver optimization updates for unity transmit power and σ^2=0.8

The mean BER of each implementation was determined by repeating the BER measurements for 80 complex channel realizations and averaging them to obtain the mean BER vs transmitter SNR curve for that implementation. The data shown in Fig. 11 represent mean BER vs transmitter SNR values for the baseline, water filling and linear transceiver optimization implementations. The linear transceiver optimization implementation used a subset size of five transceivers over all complex channel realizations.

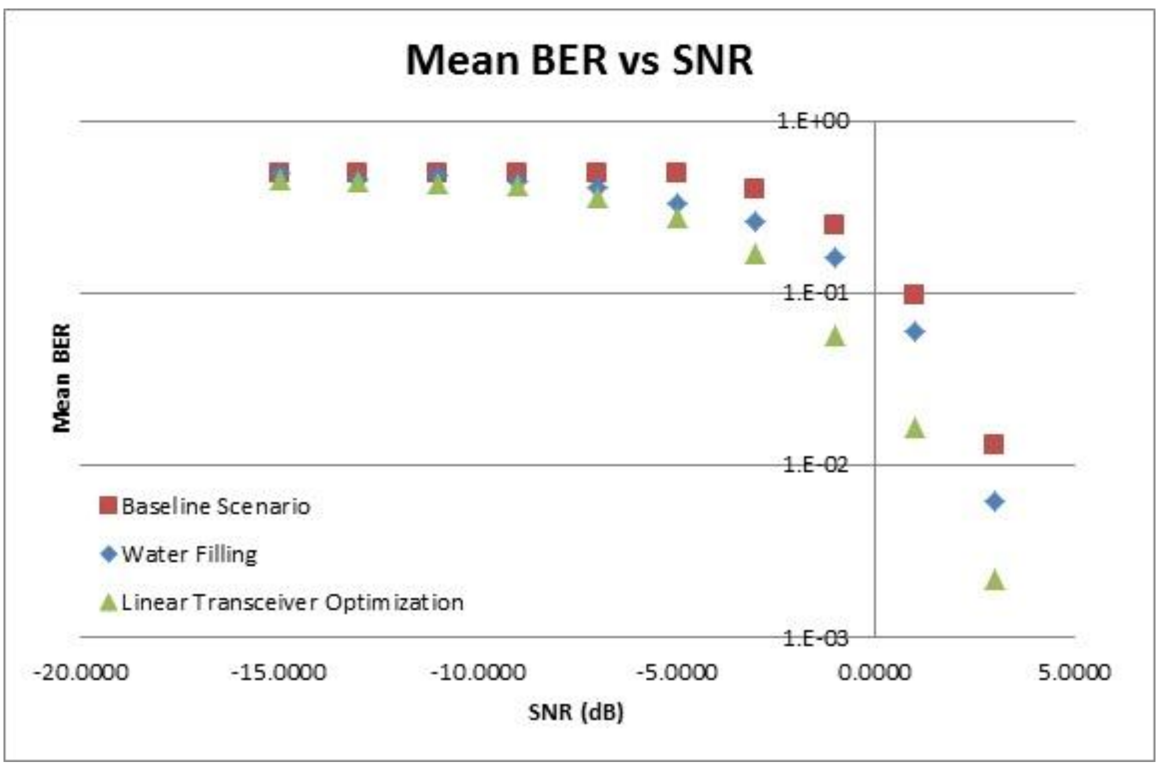

Fig. 11—Mean BER as a function of transmitter SNR for all three implementations

5. Conclusions and Future Work

5.1 Discussion

In this work, we discussed the application of MIMO wireless network optimization techniques to MIMO ad-hoc wireless networks. In particular, we simulated the performance (in terms of BER vs SNR) for both water filling and linear transceiver optimization techniques, and compared them to the baseline performance of baseline MIMO wireless networks.

From our results, we can clearly see that the use of both water filling and linear transceiver optimization techniques to optimize MIMO wireless network performance can extend to the ad-hoc case. Our results show that, similar to the non-ad-hoc case, water filling allows for a moderate improvement in network performance, while linear transceiver optimization results in a significant improvement over the baseline implementation.

The water filling implementation, an approach designed to distribute power to the antennas in such a way to take advantage of the best channels available, was able to begin to reduce the mean BER of the system from 0.5 at around -9dB SNR, while the baseline case started reducing BER at around -5dB SNR, shown in Fig. 11.

The linear transceiver optimization implementation, designed to generate optimal precoders and decoders for a specific subset of transceivers to improve system performance within the subset, started reducing the mean BER from 0.5 at around -12dB SNR, which was a further improvement compared to both baseline and water filling implementations.

5.2 Future Work

This thesis lays the foundation for several future avenues for exploration, both within WCAN and elsewhere. By observing the performance improvement evident in Fig. 10, one naturally wonders if there are alternate methods that can be used to improve system performance even further.

In particular, in our linear transceiver implementation, the precoders and decoders of each subset was optimized independently of the other—we treated the signals sent from the other group as noise. However, it is not hard to imagine that there could be further improvement in system performance by taking the other subset into account and screening out the signals transmitted from it, rather than treating those signals as noise—that is, if it were possible to jointly optimize the precoders and decoders of all subsets.

The MIMO ad-hoc wireless network implementations can also be better modeled by increasing the number of transceivers in the environment. By increasing the total number of users in the environment at any given time, it would be possible to observe the effects of increasing degrees of interference in the different implementations.

Another avenue for exploration is to vary transmitter design: in all of our implementations, the maximum transmit power constraint was fixed at an identical constant value for all of the transceivers in the environment. In reality, many devices may operate on the same frequency bands using variable transmit powers dependent on the device and its applications. In addition to varying transmit power, the number of antennas per transceiver could be relaxed from a constant to a user-defined variable, in order to represent the different devices that could be present in the network.

Appendix A: Source Code

A.1 MIMO Ad-Hoc Wireless Network Code (Baseline Scenario)

```
% Michael Wang
% WCAN@PSU, Spring 2011
% MATLAM script-m file

clear
clc

for repetition = 1:80
    % Create variables
    numNodes     = 10;       % Number of nodes
    numError     = 0;        % Number of errors in transmission
    BER          = 0;        % Bit-Error Rate (BER) of the transmission
    SNR          = 0;        % SNR of the transmission
    SNRcurrent   = 0;        % Helper variable for SNR calculation
    msgLength    = 10^5;     % Length of message in bits (10^5 = 5min)

    % Set up environment
    e = env(numNodes);
    e.plotNodes();           % Display scatter-plot on next figure number
    e.generateH(numNodes);            % Generate H
    e.generatePL(numNodes);           % Generate PL
    list = e.getNodeList();           % Store e.nodeList locally

    % Select primary transmit/receive nodes
    tNode        = ceil(numNodes*rand(1));        % Randomly choose node
    rNode        = ceil(numNodes*rand(1));        % Randomly choose node

    while (rNode == tNode)                        % Different Tx/Rx
        rNode    = ceil(numNodes*rand(1));
    end


    tic;
    % BER Calculation Loop
    for index = 1:msgLength
        % Generate and encode messages in all nodes
        for genIndex = 1:numNodes
            if (genIndex ~= rNode)
                list(genIndex).generateMessage();
                list(genIndex).encodeMessage();
            end
        end

        % Transmit messages from all nodes
        r = zeros(e.getNode(rNode).getAnt(),1);      % Generates empty
(nAnt x 1) array for signal
        for txIndex = 1:numNodes
            if (txIndex ~= rNode)
                r = r + (10^(-
e.getPL(txIndex,rNode)/20))*e.getH(txIndex,rNode)*list(txIndex).getEnco
ded()./4;
```

```
            end
        end
        r = r + list(rNode).generateN();        % Add receiver noise

        % Decode and Check for errors between Tx/Rx nodes
        r_resolved = (e.getH(tNode,rNode))'*r;
        r_real = real(r_resolved);
        s_tx    = e.getNode(tNode).getMessage();
        error = 0;
        for check = 1:4
            if ((r_real(check) > 0) ~= s_tx)
                error = error + 1;
            end
        end

        if (error > 2)
            numError = numError + 1;
        end
        if ((error == 2) && ( (mean(r_real) > 0) ~= s_tx))
            numError = numError + 1;
        end

        % Update SNR
        tMessage = 10^(-
e.getPL(tNode,rNode)/20)*e.getH(tNode,rNode)*list(tNode).getEncoded();
        SNRcurrent = sum(abs(tMessage).^2)/sum(abs(r-tMessage).^2);
        SNR = ((index-1)*SNR + SNRcurrent)/index;

    end
    tEnd = toc;

    % Convert SNR to SNR(dB) (using power ratios)
    SNRdB = 10*log10(SNR);

    % Calculate BER
    BER = numError/msgLength;

    % Write to Excel
    writelocation = ['A',num2str(3+repetition)];
    write = {num2str(repetition), num2str(tNode), num2str(rNode),
num2str(e.distance(tNode,rNode)), num2str(SNR), num2str(SNRdB),
num2str(BER)};
    xlswrite('adHoc_noOpt_data.xls', write, 'Data', writelocation);

end
```

A.2 Water-Filling Code

```
% Michael Wang
% WCAN@PSU, Spring 2011
% MATLAM script-m file

clear
clc
```

```
% WITH WATER FILLING

for repetition = 1:80
    % Create variables
    numNodes    = 10;       % Number of nodes
    numError    = 0;        % Number of errors in transmission
    BER         = 0;        % Bit-Error Rate (BER) of the transmission
    SNR         = 0;        % SNR of the transmission
    SNRcurrent  = 0;        % Helper variable for SNR calculation
    msgLength   = 10^5;     % Length of message in bits (10^5 = 5min)
    pwrAlloc    = ones(4,1,numNodes,numNodes);  % Power Allocation
matrix (upper triangular)

    % Set up environment
    e = env(numNodes);
    e.plotNodes();         % Display scatter-plot on next figure number
    e.generateH(numNodes);          % Generate H
    e.generatePL(numNodes);         % Generate PL
    list = e.getNodeList();         % Store e.nodeList locally

    % Select primary transmit/receive nodes
    tNode       = ceil(numNodes*rand(1));       % Randomly choose node
    rNode       = ceil(numNodes*rand(1));       % Randomly choose node

    while (rNode == tNode)                      % Different Tx/Rx
        rNode   = ceil(numNodes*rand(1));
    end

    % Power Allocation Setup for all nodes
    for index = 1:numNodes
        if (index ~= rNode)
            numChannel = 4;
            WF = 1;

            % Generate power allocation matrix
            P = real(e.getH(index,rNode)*e.getH(index,rNode)');
            eig = real(sort(diag(P),'descend'));
            relation = zeros(1,numChannel);
            minimum = min(index,rNode);
            maximum = max(index,rNode);
            for j = 1:numChannel
                for k = 1:numChannel
                    if (P(k,k) == eig(j))
                        relation(j) = k;    % kth antenna has jth
                                            % largest eigenvalue
                                            % relation = [antenna w/
                                            % largest eigenvalue,
                                            % antenna w/ 2nd largest...
                                            % antenna w/ smallest eig]
                    end
                end
            end

            while(WF && (numChannel>0) )
```

```
                sumInvEig = 0;
                % Sum 1/eigenvalues
                for j=1:numChannel
                    sumInvEig = sumInvEig + 1/(eig(j));
                end
                % Find constant u
                u = 1/numChannel + 1/numChannel*sumInvEig;
                for j=1:numChannel
                    pwrAlloc(j,1,minimum,maximum) = u - 1/eig(j);
                end
                % Check for all positive powers
                if (pwrAlloc(numChannel,1,minimum,maximum)>0)
                    WF = 0;
                else
                    pwrAlloc(numChannel,1,minimum,maximum) = 0;
                    numChannel = numChannel - 1;
                end
            end

            % Save power allocation into pwrAlloc(:,1) for(index,rNode)
            permute=zeros(4,4);
            for j=1:4
                permute(relation(j),j) = 1;
            end
            pwrAlloc(:,1,minimum,maximum) =
                  permute*pwrAlloc(:,1,minimum,maximum);
        end
    end

    tic;
    % BER Calculation Loop
    for index = 1:msgLength
        % Generate and encode messages in all nodes
        for genIndex = 1:numNodes
            if (genIndex ~= rNode)
                list(genIndex).generateMessage();
                list(genIndex).encodeMessage();
            end
        end

        % Transmit messages from all nodes (including power allocation
        r = zeros(e.getNode(rNode).getAnt(),1);     % Generates empty
(nAnt x 1) array for signal
        for txIndex = 1:numNodes
            if (txIndex ~= rNode)
                r = r + (10^(-
e.getPL(txIndex,rNode)/20))*e.getH(txIndex,rNode)*list(txIndex).getEnco
ded().*pwrAlloc(:,1,min(txIndex,rNode),max(txIndex,rNode));
            end
        end
        r = r + list(rNode).generateN();      % Add receiver noise

        % Decode and Check for errors between Tx/Rx nodes
        r_resolved = (e.getH(tNode,rNode))'*r;
        r_real = real(r_resolved);
        s_tx    = e.getNode(tNode).getMessage();
```

```
        error = 0;
        for check = 1:4
            if ((r_real(check) > 0) ~= s_tx)
                error = error + 1;
            end
        end

        if (error > 2)
            numError = numError + 1;
        end
        if ((error == 2) && ( (mean(r_real) > 0) ~= s_tx))
            numError = numError + 1;
        end

        % Update SNR
        tMessage = 10^(-
e.getPL(tNode,rNode)/20)*e.getH(tNode,rNode)*list(tNode).getEncoded();
        SNRcurrent = sum(abs(tMessage).^2)/sum(abs(r-tMessage).^2);
        SNR = ((index-1)*SNR + SNRcurrent)/index;

    end
    tEnd = toc;

    % Convert SNR to SNR(dB) (using power ratios)
    SNRdB = 10*log10(SNR);

    % Calculate BER
    BER = numError/msgLength;

    % Write to Excel
    writelocation = ['Z',num2str(144+repetition)];
    write = {num2str(repetition), num2str(tNode), num2str(rNode),
num2str(e.distance(tNode,rNode)), num2str(SNR), num2str(SNRdB),
num2str(BER)};
    xlswrite('adHoc_noOpt_data.xls', write, 'Data', writelocation);

end
```

A.3 Linear Transceiver Optimization Code

```
% Michael Wang
% WCAN@PSU, Spring 2011
% MATLAM script-m file

clear
clc
% WITH LINEAR TRANSCEIVER OPTIMIZATION

for repetition = 1:80
    % Create variables
    numNodes    = 10;       % Number of nodes
    numError    = 0;        % Number of errors in transmission
    noiseVar    = 10^-13;   % Noise Variance
```

```matlab
BER          = 0;         % Bit-Error Rate (BER) of the transmission
SNR          = 0;         % SNR of the transmission
SNRcurrent   = 0;         % Helper variable for SNR calculation
msgLength    = 10^5;      % Length of message in bits (10^5 = 5min)

% Create variables (Transceiver Optimization)
maxLoop = 20;             % Number of iterations for F/G optimization
F = ones(4,4,numNodes);   % Linear Transmitters
G = ones(4,4,numNodes);   % Linear Receivers
uk = zeros(numNodes,1)    % u_k values
ukTolerance = 0.0001;     % Tolerance for binary search for u_k
distTol = 0.0000001;
A = [];
B = [];
C = [];
D = [];
U = [];
V = [];
search = 1;               % u_k binary search algorithm indicator
u_MIN = 0;                % Lower bound for u_k binary search
u_MAX = 100000;           % Upper bound for u_k binary search

% Set up environment
e = env(numNodes);
e.plotNodes();           % Display scatter-plot on next figure number
e.generateH(numNodes);            % Generate H
e.generatePL(numNodes);           % Generate PL
list = e.getNodeList();           % Store e.nodeList locally

% Select primary transmit/receive nodes
tNode        = ceil(numNodes*rand(1));        % Randomly choose node
rNode        = ceil(numNodes*rand(1));        % Randomly choose node

while (rNode == tNode)                        % Different Tx/Rx
    rNode    = ceil(numNodes*rand(1));
end

% Linear Transceiver Optimization for all Nodes ===================
tic;

% Set up starting linear transmitters F ---------------------------
% Set up starting T matrix value
T = noiseVar*eye(4,4);
for index = 1:numNodes
    if index ~= rNode
        H = e.getH(index,rNode);
        trans = F(:,:,index);
        T = T + H*(trans*trans')*H';
    end
end
% Find starting u_k values
for index = 1:numNodes
    if index ~= rNode
        H = e.getH(index,rNode);
        search = 1;        % Reset search booleans and binary search
        u_low = u_MIN;
```

```
            u_high = u_MAX;
            A = H'*(T^-1 - noiseVar*T^-2)*H;
            B = H'*T^-1*H*F(:,:,index);
            [U,D,V] = svd(A);
            C = real(V'*(B*B')*U);


            % Run Binary Search for u_k=u_(index)  (using starting F)
            % Check zero
            total = -list(index).getPower();
            for i = 1:list(index).getAnt()
                total = total + C(i,i)/(real(D(i,i)))^2;
            end
            if (real(total) < 0)
                search = 0;
                uk(index) = 0;
            end
            % Binary search
            while(search)
                midpoint = 0.5*(u_low + u_high);
                % Check midpoint
                total = -list(index).getPower();
                for i = 1:list(index).getAnt()
                    total = total + C(i,i)/(midpoint + real(D(i,i)))^2;
                end
                if abs(total) < ukTolerance
                    uk(index) = midpoint;
                    search = 0;
                else
                    % Search Lower Half/Upper Half
                    if (real(total) < 0)
                        u_high = midpoint;
                    else
                        u_low = midpoint;
                    end
                end
            end
        end
    end

    % Iterate to find optimal Transmitters/Receivers {F},{G}-----------
    for loop = 1:maxLoop
        % Update Transmitters {F}  (using X)
        for trans = 1:numNodes
            if trans ~= rNode
                H = e.getH(trans,rNode);
                X = (uk(trans)*eye(4,4) + H'*(T^-1 - noiseVar*T^-
2)*H)^-1*H'*T^-1*H*F(:,:,trans);
                %===========% Find u_k for k=index (using X*X')
                for index = 1:numNodes
                    if index ~= rNode
                        H = e.getH(index,rNode);
                        search = 1; % Reset search boolean/binary
search
                        u_low = u_MIN;
                        u_high = u_MAX;
```

```
                        A = X'*(T^-1 - noiseVar*T^-2)*X;
                        B = X'*T^-1*X*F(:,:,index);
                        [U,D,V] = svd(A);
                        C = real(V'*(B*B')*U);

                        % Run Binary Search for u_k=u_(index)
                        % Check zero
                        total = -list(index).getPower();
                        for i = 1:(list(index).getAnt())
                            total = total + C(i,i)/(real(D(i,i)))^2;
                        end
                        if (total < 0)
                            search = 0;
                            uk(index) = 0;
                        end
                        % Binary search
                        while(search)
                            midpoint = 0.5*(u_low + u_high);
                            % Check midpoint
                            total = -list(index).getPower();
                            for i = 1:(list(index).getAnt())
                                total = total + C(i,i)/(midpoint +
                                          real(D(i,i)))^2;
                            end
                            if (abs(real(total)) < ukTolerance)
                                uk(index) = midpoint;
                                search = 0;
                            end
                            if (abs(u_high-u_low) < distTol)
                                uk(index) = midpoint;
                                search = 0;
                            else
                                % Search Lower Half/Upper Half
                                if (real(total) < 0)
                                    u_high = midpoint;
                                else
                                    u_low = midpoint;
                                end
                            end
                        end
                    end
                end
                % ==========
                % Update F(index)
                F(:,:,trans) = (uk(trans)*eye(4,4) + H'*(T^-1 -
noiseVar*T^-2*H)*H')^-1*H'*T^-1*H*F(:,:,trans);
            end
        end

        % Update T
        for index = 1:numNodes
            if index ~= rNode
                H = e.getH(index,rNode);
                trans = F(:,:,index);
                T = T + H*(trans*trans')*H';
            end
```

```
        end

    end

    % Generate the Receivers {G}---------------------------------------
    for index = 1:numNodes
        if index ~= rNode
            G(:,:,index) = F(:,:,index)'*e.getH(index,rNode)'*T^-1;
        end
    end

    % End Linear Transmitter/Receiver Generation ======================
    tEndTrans = toc;



    % BER Calculation Loop
    for index = 1:msgLength
        % Generate and encode messages in all nodes
        for genIndex = 1:numNodes
            if (genIndex ~= rNode)
                list(genIndex).generateMessage();
                list(genIndex).encodeMessage();
            end
        end

        % Transmit messages from all nodes (including power allocation
        r = zeros(e.getNode(rNode).getAnt(),1);      % Generates empty
(nAnt x 1) array for signal
        for txIndex = 1:numNodes
            if (txIndex ~= rNode)
                r = r + (10^(-
e.getPL(txIndex,rNode)/20))*e.getH(txIndex,rNode)*F(:,:,txIndex)*list(t
xIndex).getEncoded();
            end
        end
        r = r + list(rNode).generateN();       % Add receiver noise

        % Decode and Check for errors between Tx/Rx nodes
        r_resolved = G(:,:,tNode)*r;
        r_real = real(r_resolved);
        s_tx    = e.getNode(tNode).getMessage();
        error = 0;
        for check = 1:4
            if ((r_real(check) > 0) ~= s_tx)
                error = error + 1;
            end
        end

        if (error > 2)
            numError = numError + 1;
        end
        if ((error == 2) && ( (mean(r_real) > 0) ~= s_tx))
            numError = numError + 1;
        end
```

```
        % Update SNR
        tMessage = 10^(-
e.getPL(tNode,rNode)/20)*e.getH(tNode,rNode)*list(tNode).getEncoded();
        SNRcurrent = sum(abs(tMessage).^2)/sum(abs(r-tMessage).^2);
        SNR = ((index-1)*SNR + SNRcurrent)/index;

    end
    tEnd = toc;

    % Convert SNR to SNR(dB) (using power ratios)
    SNRdB = 10*log10(SNR);

    % Calculate BER
    BER = numError/msgLength;

    % Write to Excel
    writelocation = ['A',num2str(3+repetition)];
    write = {num2str(repetition), num2str(tNode), num2str(rNode),
num2str(e.distance(tNode,rNode)), num2str(SNR), num2str(SNRdB),
num2str(BER)};
    xlswrite('adHoc_noOpt_data.xls', write, 'Data', writelocation);

end
```

Bibliography

[1] A. Goldsmith, *Wireless Communications*. Cambridge, England: Cambridge Univ. Press, 2005, pp. 4-8, 89, 107-111, 205, 323-329.

[2] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, pp. 379-423,623-656, July, October, 1948.

[3] G. J. Foschini, "Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas," *Bell Labs Technical Journal*, pp. 41-47, Autumn 1996.

[4] E. Telatar, "Capacity of Multi-Antenna Gaussian Channels," *AT&T Bell Labs Memo.*, Murray Hill, NJ, 1995.

[5] S. Serbetli and A. Yener, "Time-Slotted Multiuser MIMO Systems: Beamforming and Scheduling Strategies," *EURASIP Journal on Wireless Communications and Networking*, pp. 286-296, Feb. 2004.

[6] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, Mar 1998.

[7] S. M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Select Areas in Communications*, vol. 16, no. 8, Oct. 1998.

[8] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel," *International Symposium on Signals, Systems, and Electronics, 1998*, pp. 295-300, Sept 1998.

[9] C. B. Papadias and G. J. Foschini, "A Space-Time Coding Approach for Systems Employing Four Transmit Antennas," *IEEE International Conference on Acoustics, Speech, and Signal Processing 2001 Proceedings*, vol. 4, pp. 2481-2484, 2001.

[10] B. Chen and M. J. Gans, "MIMO Communications in Ad Hoc Networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 7, July 2006.

[11] S. Serbetli and A. Yener, "Transceiver Optimzation for Multiuser MIMO Systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 214-226, Jan. 2004.

[12] W. Yu, W. Rhee, S. Boyd, and J. M. Cioffi, "Iterative Water-filling for Gaussian Vector Multiple Access Channels," *IEEE Transactions on Information Theory*, vol. 50, no. 1, pp. 145-152, Jan. 2004.

ACADEMIC VITA

Michael Wang
2910 Arch Road
Plymouth Meeting, PA 19462

EDUCATION:

Pennsylvania State University (University Park) — University Park, PA
Bachelor of Science in Electrical Engineering, with Honors
Bachelor of Science in Physics, Electronics Option
Minor in Mathematics
Degrees Anticipated: May 2011

EXPERIENCE:

Research Assistant — Dec 2009 – April 2011
Wireless Communications and Networking Laboratory, Pennsylvania State University
Professor Aylin Yener, Penn State University

- Designed computer models for MIMO wireless communication networks
- Compared and verified wireless network characterization with published studies

Research Assistant — Feb 2008 – Aug 2008
Superconductivity and Quantum Electronics Materials Laboratory, Penn State University
Professor Ying Liu, Penn State University

- Developed graphene device manufacturing process using SiO substrate
- Built and tested graphene devices via thin-film deposition

PUBLICATIONS:

“MIMO Ad-Hoc Wireless Networks: Evaluating Performance” — Apr 2011
Thesis Advisor: Dr. Aylin Yener, Pennsylvania State University
Honors Advisor: Dr. John D. Mitchell, Pennsylvania State University

“Cool, Clear Water: The Fairmount Water Works” — Dec 2010
<http://pabook.libraries.psu.edu/palitmap/FairmountWW.html>.

SKILLS:

Computing Languages: Matlab w/ Simulink, Mathematica
Programming Languages: C++, Java, HTML
Thin Film Deposition
Microscopy: Atomic Force, Focused Ion Beam, Scanning Electron, Tunneling Electron
Languages: Chinese (Mandarin)

HONORS:

Sigma Pi Sigma Physics Honor Society	2010-2011
Schreyer Honors College	Fall 2007-Spring 2011
Dean’s List	Fall 2007-Spring 2011
Braddock Scholarship	2007-2011
Bert Elsbach Scholarship in Physics	2009-2010
John and Elizabeth Holmes Teas Scholarship	Summer 2008
The President’s Freshman Award	Spring 2008