ECE 6604 Design Project: Performance of STBC Schemes over Rayleigh Block-Fading Channel for 4G MIMO

Abhishek Obla Hema and Klaus Okkelberg

Abstract

Wireless mobile networks are a part of everyday life, but they have limited range and data rate to satisfy the demands of the exponentially increasing number of wireless mobile devices. In particular, high energy consumption and spectrum contraints mean that current 3.9G technology will not be able to satisfy the demands of mobile data traffic in the near future. To overcome these limitations, various diversity schemes are used, which fall into six categories, namely time, space, angle, polarization, frequency, and multipath. One implementation of a diversity scheme is to use MIMO links, which use multiple transmit and receive antennas to increase time and spatial diversity, the result of which is to increase range and data rate without requiring additional transmit power or bandwidth. To this end, this paper investigates the performance of space-time block coding (STBC) schemes for up to eight transmit antennas and up to two receive antennas, in line with current 4G standards for cellular phones. These schemes are analyzed for their spectral efficiency and bit error rates for transmitting data modulated using BPSK, QPSK, 16-QAM, and 64-QAM over a Rayleigh block-fading channel with additive white Gaussian noise. This is accomplished through software simulation in Matlab.

I. Introduction

Current wireless high-speed data communication standards are considered to be 3.9G with 4G technologies deploying in the near future. Of the 3.9G technologies, Long Term Evolution (LTE) developed by the 3rd Generation Partnership Project (3GPP) is far more popular than IEEE 802.16e WiMAX. LTE is commonly marketed as 4G LTE even though it does not comply with ITU-R IMT-Advanced requirements to be considered 4G. Current 4G standards include 3GPP Long Term Evolution Advanced (LTE-A) and IEEE 802.16m WiMAX Release 2, with LTE-A being the more popular of the two currently. This delay in the development and deployment of actual 4G standards means that future 5G standards have a lot of catching up to do in order to meet the exponentially increasing number of mobile devices, each with greater data needs. To this end, this paper explores one possible method of meeting those needs, namely using massive MIMO systems with appropriately chosen space-time block codes.

The paper is organized as follows. The remainder of this section discusses the motivation and implementation of LTE and LTE-A. Section II provides background on diversity-increasing schemes used in current 3.9G and 4G standards, especially MIMO. Section III discusses current STBC schemes. Section IV explains the simulation methodology. The simulation results are presented and analyzed in section V. Finally, section VI summarizes the results and discusses how they can be applied to future 5G technologies. The simulation code used for this paper is included in the appendix.

A. LTE

LTE is a registered trademark owned by ETSI (European Telecommunications Standards Institute) for the wireless data communications technology and a development of the GSM/UMTS standards. LTE is based on the GSM/EDGE and UMTS/HSPA network technologies, increasing the capacity and speed using a different radio interface together with core network improvements. LTE is the natural upgrade path for carriers with both GSM/UMTS networks and CDMA2000 networks. The different LTE frequencies and bands used in different countries mean that only multi-band phones are be able to use LTE in all countries where it is supported.

The goal of LTE was to increase the capacity and speed of wireless data networks using new digital signal processing (DSP) techniques and modulations that were developed around the turn of the millennium. A further goal was the redesign and simplification of the network architecture to an IP-based system with significantly reduced transfer latency compared to the 3G architecture. The LTE wireless interface is incompatible with 2G and 3G networks, so it must be operated on a separate radio spectrum. The LTE specification provides downlink peak rates of 300 Mbit/s, uplink peak rates of 75 Mbit/s and QoS provisions permitting a transfer latency of less than 5ms in the radio access network. LTE has the ability to manage fast-moving mobiles and supports multi-cast and broadcast streams. LTE supports scalable carrier bandwidths, from 1.4 MHz to 20 MHz and supports both frequency division duplexing (FDD) and time-division duplexing (TDD). The IP-based network architecture, called the Evolved Packet Core (EPC) designed to replace the GPRS Core Network, supports seamless handovers for both voice and data to cell towers with older network technology such as GSM, UMTS and CDMA2000. The simpler architecture results in lower operating costs (for example, each E-UTRA cell will support up to four times the data and voice capacity supported by HSPA)

LTE supports deployment on different frequency bandwidths. The current specification outlines the following bandwidth blocks: 1.4MHz, 3MHz, 5MHz, 10MHz, 15MHz, and 20MHz. Frequency bandwidth blocks are essentially the amount of space a network operator dedicates to a network. Depending on the type of LTE being deployed, these bandwidths have slightly different meaning in terms of capacity. That will be covered later, though. An operator may choose to deploy LTE in a smaller bandwidth and grow it to a larger one as it transitions subscribers off of its legacy networks (GSM, CDMA, etc.).

LTE uses two different types of air interfaces (radio links), one for downlink (from tower to device), and one for uplink (from device to tower). By using different types of interfaces for the downlink and uplink, LTE utilizes the optimal way to do wireless connections both ways, which makes a better-optimized network and better battery life on LTE devices.

For the downlink, LTE uses an orthogonal frequency division multiple access (OFDMA) air interface as opposed to the code division multiple access (CDMA) and time division multiple access (TDMA) air interfaces . OFDMA (unlike CDMA and TDMA) mandates that multiple-input, multiple-output (MIMO) is used. Having MIMO means that devices have multiple connections to a single cell, which increases the stability of the connection and reduces latency tremendously. It also increases the total throughput of a connection. MIMO is what lets 802.11n WiFi reach speeds of up to 600Mbps, though most advertise up to 300-400Mbps. There is a significant disadvantage though. MIMO works better the further apart the individual carrier antennae are, in order to increase spatial diversity. On smaller phones, the interference caused by the antennae being so close to each other will cause LTE performance to drop.

For the uplink (from device to tower), LTE uses the discrete Fourier transform spread OFDMA (DFTS-OFDMA) scheme of generating a single carrier frequency division multiple access (SC-FDMA) signal. As opposed to regular OFDMA, SC-FDMA is better for uplink because it has a better peak-to-average power ratio over OFDMA for uplink. LTE-enabled devices, in order to conserve battery life, typically do not have a strong and powerful signal going back to the tower, so a lot of the benefits of normal OFDMA would be lost with a weak signal. Despite the name, SC-FDMA is still a MIMO system. LTE uses a SC-FDMA 1×2 configuration, which means that for every one antenna on the transmitting device, there are two antennae on the base station for receiving.

The LTE technology itself also comes in two flavors: an FDD (frequency division duplex) variant and a TDD (time division duplex) variant. The most common variant being used is the FDD variant. The FDD variant uses separate frequencies for downlink and uplink in the form of a band pair. That means for every band that a phone supports, it actually uses two frequency ranges. These are known as paired frequency bands. For example, Verizons 10MHz network is in FDD, so the bandwidth is allocated for uplink and downlink. This is commonly noted as a 2x10MHz or 10+10 MHz configuration. Some also call it 10x10MHz, but this is mathematically incorrect, but they mean 10+10MHz. Some will also call it a 20MHz network, but this can be ambiguous. The TDD variant uses one single range of frequencies in a frequency band, but that band is segmented to support transmit and receive signals in a single frequency range.

For example, an LTE TDD network deployed on 20MHz of spectrum uses the whole chunk as one large block for frequency allocation purposes. For network bandwidth purposes, a LTE TDD networks spectrum can be further divided to optimize for the type of network traffic (half up and half down, mostly down and a bit up, mostly up and a bit down, and so on).

B. LTE-A

LTE-A is a major enhancement of the LTE standard that meets 4G requirements. It was standarized by 3GPP as 3GPP Release 10. Presently, there are few deployments and almost no capable smartphones.

The main developments of LTE-A over LTE are [1]

- Ability to use up to 8×8 MIMO and 128-QAM in downlink
- Up to 100 MHz of carrier aggregated bandwidth
- Higher spectral efficiency with a maximum of 30 bps/Hz in LTE-A release 10 versus 16 bps/Hz in release 8
- 1 Gbit/s peak data rate per user
- Increased peak data rate of 3 Gbit/s download and 1.5 Gbit/s upload
- Faster switching between power states
- Improved performance at cell edges

Even with these enhancements, LTE-A is backwards compatible with LTE, meaning an LTE mobile station would work in an LTE-A network and vice versa.

C. Impact of LTE on Battery Life

By itself, LTE devices should last roughly as long as their HSPA+ equivalents because of the optimized radios for both downlink and uplink operations. The reason why LTE devices right now eat batteries for breakfast is because the network operators are forcing many of these devices into active dual-mode operation.

For Verizon Wireless, this means that most of their LTE devices connect to both CDMA2000 and LTE simultaneously and stay connected to both. This consumes twice the amount of battery for every minute connected than if the phone were connected only to CDMA2000 or LTE. Additionally, when calls are made on Verizon Wireless LTE phones, the CDMA2000 radio sucks down more power because you are talking. Sending and receiving text messages causes pulses of CDMA2000 activity, which cuts battery life more. Arguably, constantly changing radio states could be worse for battery life than a switch into one mode for a period of time and switching back, so text messages may actually kill the batteries faster.

Also of importance for battery life is handover. Handover is the operation in which a device switches from one network to another or from one tower to another. Handover is the critical component that makes any cellular wireless network possible. Without handover, a user would have to manually select a new tower every time the user leaves the range of a tower. For cellular networks, this is even more critical because the range of a tower is not very predictable due to factors outside of anyones control (like the weather, etc.). LTE supports handover like all other cellular wireless networks, but it improves on it by doing it much faster when handing over to a supported type of network or cell.

II. MIMO AND DIVERSITY

MIMO technology is one of the most important technologies to improve the system performance in capacity, coverage and the user data rates. The performance gains depend on the propagation characteristics of each scenario. Wi-Fi, LTE, and many other radio both RF as well as wireless technologies use the new

MIMO wireless technology to increase the link capacity, spectral efficiency and the link reliability using interference paths.

Rayleigh fading of signals yields a very large performance loss by converting the exponential dependency of bit error probability P_b on the mean received bit energy-to-noise ratio E_b/N_0 into an inverse linear one. Diversity is one remedy that improves the reliability of communication by providing the receiver with multiple independently-faded copies of the same information. These copies are often referred to as diversity branches. Diversity methods can be categorized as those that exploit (1) spatial, (2) angle, (3) polarization, (4) frequency, (5) multipath, and (6) time diversity. At the receiver, the diversity branches are combined together, with the most effective combining method depending, among other things, on the type of additive impairment. For additive white Gaussian noise (AWGN) dominant channels, maximal ratio combining (MRC) is optimal in the maximum likelihood (ML) sense. For co-channel interference (CCI) dominant channels, optimum combining performs better.

The performance of a diversity scheme is measured by its diversity gain, defined as the number of independent receptions of the same signal. The independent copies of the signal increase the signal-to-interference ratio and allow a reduction in transmission power without a performance loss. A MIMO system with N_T transmit antennas and N_R receive antennas has a maximum diversity gain equal to $N_T N_R$.

The remainder of this section discusses the following diversity schemes, as these are the most commonly used.

- *Time diversity*: In time diversity, a message is transmitted at different times, e.g. using different timeslots and channel coding.
- *Frequency diversity*: In this form we use different frequencies. It may be in the form of using different channels, or technologies such as spread spectrum OFDM.
- *Space diversity*: Space diversity is used as the basis for MIMO. It uses the antennas that are located in multiple positions to take advantage of the different radio paths that exist in a typical environment.

A. Time Diversity

In time diversity, the fading of the channel is averaged over time by using channel coding and interleaving of code words. Thus, if a deep fade occurs, parts of many code words are affected rather than all of one code word. In this way, the error burst caused by the fade is spread out among many code words so that the error correcting code can handle the fading error.

B. Frequency Diversity

In frequency diversity, rather than spreading out signals in time, signals are spread out in frequency, which is called spread spectrum. Two common methods for spread spectrum are frequency hop spread-spectrum and direct sequence spread-spectrum.

Frequency hopping is one of two basic modulation techniques used in spread spectrum signal transmission. It is the repeated switching of frequencies during radio transmission, often to minimize the effectiveness of ”electronic warfare” - that is, the unauthorized interception or jamming of telecommunications. It also is known as frequency- hopping code division multiple access (FH-CDMA).

In an FH-CDMA system, a transmitter ”hops” between available frequencies according to a specified algorithm, which can be either random or preplanned. The transmitter operates in synchronization with a receiver, which remains tuned to the same center frequency as the transmitter. A short burst of data is transmitted on a narrowband. Then, the transmitter tunes to another frequency and transmits again. The receiver thus is capable of hopping its frequency over a given bandwidth several times a second, transmitting on one frequency for a certain period of time, then hopping to another frequency and transmitting again. Frequency hopping requires a much wider bandwidth than is needed to transmit the same information using only one carrier frequency.

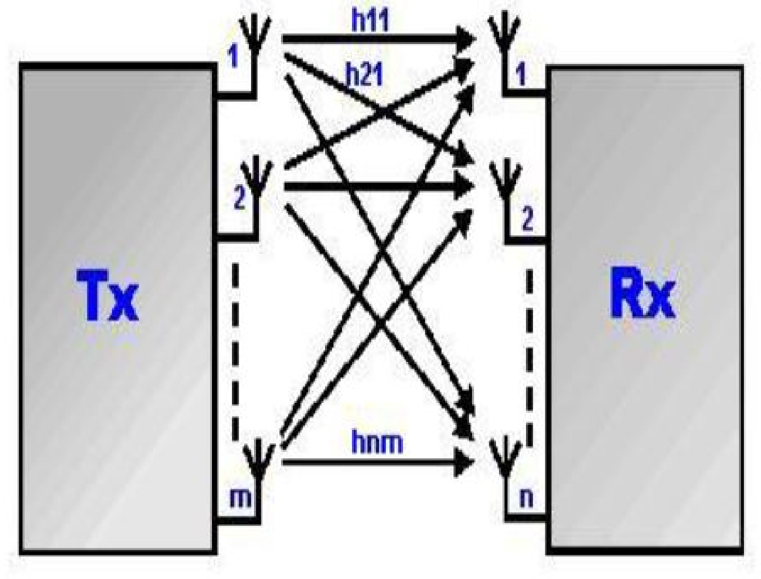

Fig. 1. General MIMO system.

In direct sequence spread spectrum, the stream of information to be transmitted is divided into small pieces, each of which is allocated across to a frequency channel across the spectrum. A data signal at the point of transmission is combined with a higher data-rate bit sequence (also known as a chipping code) that divides the data according to a spreading ratio. The redundant chipping code helps the signal resist interference and also enables the original data to be recovered if data bits are damaged during transmission. In general, frequency-hopping devices use less power and are cheaper, but the performance of DS-CDMA systems is usually better and more reliable.

C. Spatial (Antenna) Diversity

In spatial diversity, many copies of the signal in uncoded transmission or coded versions of multiple signals are sent using multi-antenna systems, so each copy undergoes different fading. This diversity method is commonly referred to as MIMO. See figure II-C for a depiction of a general MIMO system.

It is found that in between a transmitter and receiver, the signal can travel through multiple paths. So moving the antennas by even a small amount will lead to a change. The different number of paths that are available occurs as a result of a number of objects that appear in the direct path between the transmitter and receiver. Before these paths were just used to introduce interference but now using MIMO they can be used as an advantage. The two main formats of MIMO are:

- *Spatial diversity*: This refers to transmit and receive diversity. They provide improvement in the SNR and are characterized by boosting the reliability of the system with respect to fading.
- *Spatial multiplexing*: This provides additional data capacity by using the different paths to carry traffic, thereby increasing the total throughput of the system.

Since we use multiple antennas, MIMO can increase the capacity of a channel and still obey Shannons Law. With every pair of antennas that are added to the network, it is possible to linearly boost the throughput by increasing the receive and transmit antennas. Since the spectrum bandwidth is a very valuable commodity, such techniques are necessary to use the bandwidth that is available more efficiently and MIMO is one of those valuable techniques.

There are a number of MIMO configurations or formats that have been used. Each format has its own advantages and disadvantages and they can be used and balanced to provide the optimum solution for

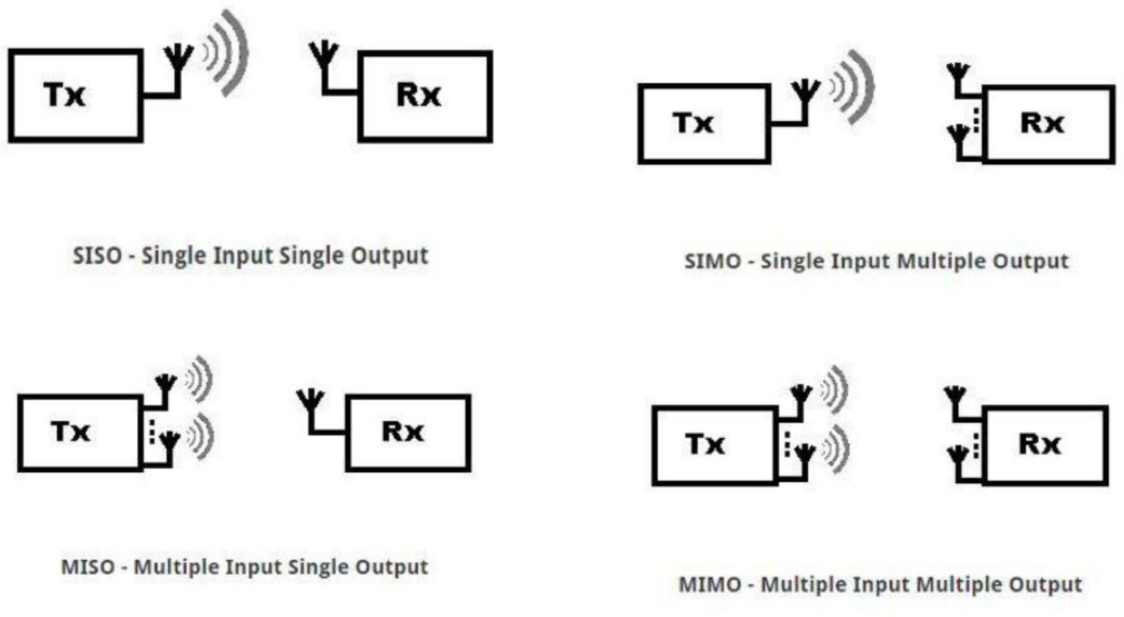
Tx
Rx
SISO - Single Input Single Output
Tx
Rx
SIMO - Single Input Multiple Output
Tx
Rx
MISO - Multiple Input Single Output
Tx
Rx
MIMO - Multiple Input Multiple Output

Fig. 2. Different MIMO configurations.

a particular application. The different MIMO formats—SISO, SIMO, MISO, MIMO all require different number of antennas and each requires different complexity. Depending on the format post processing might be required at one end or the other.

The different forms of antenna technology refer to the single or multiple inputs or outputs which are related to the radio link. In this way the input is the transmitter as it transmits into the system and the output is receiver. The different forms of single/multiple antenna links are shown in figure 2.

1) MIMO-SISO: The simplest form of radio link is SISO- Single Input Single Output. In this the transmitter operates with one antenna with the receiver. In this scheme no diversity and no additional processing is required. This can be viewed from the first figure.

2) MIMO-SIMO: In this scheme the transmitter has a single antenna while the receiver has multiple ones. This is called as receiver diversity. This is used to support a receiver system that receives signals from a number of sources to counter the effects of fading. This scheme has been in use for a long time with short wave listening/ receiving stations to fight ionospheric fading and interference. This is represented by the second figure. The main advantage is that is easy to implement however some processing is required at the receiver side. The use of SIMO is widespread but the receiver is located as a handset, hence it may be limited by size, cost and drain.

There are two general forms of SIMO combining that can be used, namely switched diversity and maximum ratio combining. In switched diversity SIMO, the antenna switches so that it always receives the strongest signal. In maximum ratio combining, the signals from all the receive antennas are combined in proportion to the strength of the received signal.

3) MIMO-MISO: MISO is also called as transmit diversity, where the same data is transmitted repeatedly from two transmitters. The receiver can then extract the needed data from the optimum signal. This is represented by the third figure. The main advantage of this scheme is that the multiple antennas and the repeated coding/processing is shifted from the receiver to the transmitter. If we take the case of cellphones, this results in massive savings in terms of space and reduces the processing required for coding. Again this results in reduced batter consumption, costs and smaller size.

4) MIMO: In this case there are multiple antennas at the transmitter as well as the receiver. It provides advantages in total channel throughput. This scheme has been represented by the fourth and final figure. This requires coding to separate the data from the multiple paths which requires processing but boosts the total throughput and capacity.

There are many types of MIMO that can be used from SISO, through SIMO and MISO to complete MIMO networks. These are all able to provide heavy gains in performance, but usually at the expense of processing and total number of antennas. Tradeoffs have to be made between cost, performance, size, battery life before we choose the correct scheme.

III. Space-Time Block Coding

With the carrier frequencies used in current 3.9G and 4G standards, only up to two receive antennas can be used while maintaining spatial diversity. Thus, it is necessary to use transmitter diversity, which uses multiple transmit antennas to provide the receiver with multiple uncorrelated copies of the same signal. Simple forms of transmit diversity include selective transmit diversity for time division duplexed systems as well as time division transmit diversity, time-switched transmit diversity, and delay transmit diversity for frequency division duplexed systems. More complex forms of transmit diversity use space-time, space-frequency, or space-time-frequency coding.

For MIMO systems, knowledge of the channel state information (CSI) at the transmitter (CSIT) and at the receiver (CSIR) is necessary to choose a diversity technique [2,3]. In general, the channel can be estimated at the receiver using various techniques (see [4]–[6]), so the knowledge of the CSIR can be assumed. At the transmitter, if the CSIT is known, then beamforming techniques are used to assure both the diversity gain and the array gain. However, without CSIT, space-time (ST) codes can still be used to assure the diversity gain. ST codes are a general class of error correcting codes in which the control symbols are inserted in both spatial and temporal domains.

The multi-layered space-time architecture was introduced by Foschini in [7]. Later, Space-Time Trellis Codes (STTrC) [8] were proposed, which provide the optimal tradeoff between constellation size, data rate, diversity gain, and trellis complexity but have a greater decoding complexity. Regarding encoding and decoding complexity, Alamouti [9] introduced a simple repetition diversity scheme for two transmit antennas with maximum likelihood combining at the receiver. Alamouti's transmit diversity scheme provides a maximum diversity gain and no coding gain for a minimum decoding complexity. Later, Tarokh et al. generalized the Alamouti code for an arbitrary number of transmit antennas as the Space-Time Block Code (STBC) [10].

The remainder of this section presents a mathematical model for a general MIMO system using STBC and details ML decoding methods for both linear and widely linear coding schemes. Additionally, some examples of STBCs are reviewed with emphasis on the code transmission matrix and the code parameters.

A. Space-Time Block Coded MIMO System

This paper considers digital wireless transmissions through a general MIMO system with N_T transmit antennas and N_R receive antennas, consisting of a ST encoder, a Rayleigh block-fading channel with AWGN, and a ST decoder with MRC and ML decoding. The channel is modeled as a discrete-time baseband-equivalent channel. This study assumes single-carrier modulated block transmission with zero padding is used to suppress intersymbol interference (ISI) and prevent inter-channel interference (ICI), as in [11,12] and uncorrelated path gains for the receive antennas.

1) Space-Time Encoding: Assume that the input to the ST encoder is a k-dimensional vector of complex symbols $\boldsymbol{s} = [s_1\, s_2 \cdots s_k]^{\mathrm{T}}$. The encoding method for the STBC is defined by its transmission matrix in the form

$$\boldsymbol{G} = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N_T} \\ g_{21} & g_{22} & \cdots & g_{2N_T} \\ \vdots & \vdots & & \vdots \\ g_{T1} & g_{T2} & \cdots & g_{TN_T} \end{bmatrix},$$

where g_{ij} is the modulated symbol transmitted in the ith time slot from the jth antenna. Note that the g_{ij} are combinations of the input complex symbols and their conjugates and the encoding is independent of the number of receive antennas N_R.

During each time slot $i = 1, \ldots, T$, the symbols are sent simultaneously from the N_T transmit antennas. Because k symbols are transmitted during T time slots, the code rate is $R = k/T$, where the maximum possible rate is unity [10]. For simplicity, it is useful to refer to a code by its parameters as $C(N_T, k, T)$. With this notation, an uncoded transmission is $C(N_T, 1, 1)$ and Alamouti code with two transmit antennas and full rate is $C(2, 2, 2)$.

In practice, the class of useful codes is restricted to Orthogonal STBCs (OSTBCs), for which the transmission matrix $\boldsymbol{G}$ satisfies [9]:

- *linearity*: its elements g_{ij} are linear combinations of the input symbols and their conjugates
- *orthogonality*:

$$\boldsymbol{G}^{\mathrm{H}}\boldsymbol{G} = \|\boldsymbol{s}\|_2^2 \mathbf{I}_{N_T} = \left(\sum_{i=1}^{k} |s_i|^2 \right) \mathbf{I}_{N_T},$$

where $\|\cdot\|_2$ is the Euclidean norm and $\mathbf{I}_{N_T}$ is the $N_T \times N_T$ identity matrix. OSTBCs have the advantage of providing full diversity equal to $N_T N_R$ as well as making MRC of the received signals equivalent to the linear minimum mean square error (MMSE) estimate, as will be detailed in the decoding section. Note that these codes have very little or no coding gain in contrast to space-time trellis codes that spread a conventional trellis code over space and time.

Orthogonal matrices can be designed using Hurwitz-Radon (HR) theory [13,14], which shows that a HR family with $N - 1$ matrices of size $N \times N$ exists only if $N \in \{2, 4, 8\}$. STBC transmission matrices can be constructed from the HR family of matrices. For real input constellations, square STBC matrices with full rate of unity exist only for $N_T \in \{2, 4, 8\}$. However, non-square full rate STBC matrices can be designed for $N_T \in \{3, 5, 6, 7\}$ by removing columns from the square full rate matrices.

For complex input constellations, a full rate OSTBC only exists for $N_T = 2$. However, for $N_T > 2$, $R = 1/2$ codes can be constructed for complex constellations from full rate codes for real constellations [10], in essence sacrificing code rate to gain an orthogonal design. In addition, $R = 3/4$ codes exist for $N_T = 4$ but the authors do not know of any orthogonal codes with $R > 1/2$ for $N_T > 4$. Also of interest are Quasi-Orthogonal STBCs (QOSTBCs), in which only adjacent columns of the transmission matrix are orthogonal. QOSTBCs can achieve full rate communication at the expense of decoding complexity and diversity gain due to the interference [15].

2) Space-Time Coded MIMO Channel: The transmission of the ST coded symbols over the MIMO channel considered by this paper is subject to fading and additive noise. Let h_{ijk} denote the path gain between in the ith time slot between the jth transmit antenna and the kth receive antenna. Then, the received symbol y_{ik} in the ith time slot by the kth receive antenna is

$$y_{ik} = [h_{i,1,k}\, h_{i,2,k} \cdots h_{i,N_T,k}] \begin{bmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,N_T} \end{bmatrix} + n_{ik},$$

where x_{ij} is the symbol transmitted by the jth transmit antenna in the ith time slot and n_{ik} is the additive noise for the kth receive antenna at the ith time slot. Note that the x_{ij} are combinations, not necessarily linear, of the input symbols s_i and their conjugates, so this system is difficult to solve in general.

For a block-fading channel with coherence time T_c, it is assumed that the fading coefficients h_{ijk} remain constant within a frame of length T_c time slots. With Rayleigh fading, these coefficients are generated as independent complex Gaussian random variables with zero mean and variance $\Omega_p/2$ in both the real and imaginary dimensions, where Ω_p is the average energy of the transmitted discrete-time symbol. In addition, the additive noise is assumed to be complex Gaussian random variables with zero mean and variance $1/(2E_b/N_0)$, where the E_b/N_0 is the bit energy-to-noise ratio of the channel.

3) Space-Time Decoding: ST decoding for a STBC consists of MRC followed by ML decoding of the combined symbols. For an OSTBC, MRC is equivalent to solving a linear system, which is advantageous for decoding performance. As the combined symbols are independent and assuming no ISI, the ML decoding can be performed independently of the MRC. A similar approach works for QOSTBC, but the performance is reduced to due interference between symbols.

For a general OSTBC or QOSTBC, the effect of the channel can be written in matrix form as

$$\bar{\boldsymbol{y}} = [\boldsymbol{H}_1 \, \boldsymbol{H}_2] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}^* \end{bmatrix} + \bar{\boldsymbol{n}},$$

where $\boldsymbol{x}$ is the input to the STBC encoder, $\bar{\boldsymbol{y}}$ is the output of the MIMO channel, $\bar{\boldsymbol{n}}$ is the AWGN, and $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are matrices representing the equivalent channel formed by the ST encoder and the MIMO channel. For a linearly decodable ST code where $\boldsymbol{x}$ and its complex conjugate $\boldsymbol{x}^*$ are uncorrelated, it is possible to rewrite the equation as

$$\begin{bmatrix} \bar{\boldsymbol{y}}_1 \\ \bar{\boldsymbol{y}}_2^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 \\ \boldsymbol{H}_2^* \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{n}_1 \\ \boldsymbol{n}_2^* \end{bmatrix},$$

where $\bar{\boldsymbol{y}} = [\bar{\boldsymbol{y}}_1^{\mathrm{T}} \, \bar{\boldsymbol{y}}_2^{\mathrm{T}}]^{\mathrm{T}}$ and $\bar{\boldsymbol{n}} = [\bar{\boldsymbol{n}}_1^{\mathrm{T}} \, \bar{\boldsymbol{n}}_2^{\mathrm{T}}]^{\mathrm{T}}$. Then, let $\boldsymbol{y} = [\bar{\boldsymbol{y}}_1^{\mathrm{T}} \, \bar{\boldsymbol{y}}_2^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{H}_{ef} = [\boldsymbol{H}_1^{\mathrm{T}} \, \boldsymbol{H}_2^{\mathrm{H}}]^{\mathrm{T}}$, and $\boldsymbol{n} = [\bar{\boldsymbol{n}}_1^{\mathrm{T}} \, \bar{\boldsymbol{n}}_2^{\mathrm{T}}]^{\mathrm{T}}$ to obtain:

$$\boldsymbol{y} = \boldsymbol{H}_{ef}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{H}_{ef}$ is the matrix of the equivalent channel formed by the ST encoder and the MIMO channel. Note that for an OSTBC, $\boldsymbol{H}_{ef}$ is an orthogonal matrix over all channel realizations because

$$\boldsymbol{H}_{ef}^{\mathrm{H}}\boldsymbol{H}_{ef} = \|\boldsymbol{H}\|_F^2 \mathbf{I}_k,$$

where $\boldsymbol{H}$ is the channel matrix, $\|\cdot\|_F$ is the Frobenius norm, and k is the dimension of the input.

Assuming perfect CSIR, the MRC coefficients equal the complex conjugate of the equivalent channel matrix [15,16]:

$$\tilde{\boldsymbol{x}} = \boldsymbol{H}_{ef}^{\mathrm{H}}\boldsymbol{y}, \tag{2}$$

where $\tilde{\boldsymbol{x}}$ is the vector of the combined symbols.

Using (1) and assuming an orthogonal ST code, (2) can be written as

$$\tilde{\boldsymbol{x}} = \boldsymbol{H}_{ef}^{\mathrm{H}}\boldsymbol{H}_{ef}\boldsymbol{x} + \boldsymbol{H}_{ef}^{\mathrm{H}}\boldsymbol{n} = \|\boldsymbol{H}\|_F^2 \mathbf{I}_k \boldsymbol{x} + \tilde{\boldsymbol{n}},$$

where $\tilde{\boldsymbol{n}}$ has zero-mean and autocorrelation $\mathrm{E}[\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^{\mathrm{H}}] = \sigma_n^2 \|\boldsymbol{H}\|_F^2 \mathbf{I}_k$. Thus, the combined symbols are decoupled into

$$\tilde{x}_i = \|\boldsymbol{H}\|_F^2 x_i + \tilde{n}_i, \quad i = 1, \ldots, k, \tag{3}$$

where k is the dimension of the $\boldsymbol{x}$. With this decoupling, ML decoding is trivial for the case of AWGN.

To see why MRC for the case of linearly decodable orthogonal ST codes is equivalent to the linear MMSE solution, consider solving for the input $\boldsymbol{x}$ in (1) using the pseudo-inverse:

$$\hat{\boldsymbol{x}} = \boldsymbol{H}_{ef}^{\dagger}\boldsymbol{y} = (\boldsymbol{H}_{ef}^{\mathrm{H}}\boldsymbol{H}_{ef})^{-1}\boldsymbol{H}_{ef}^{\mathrm{H}}\boldsymbol{y} = \frac{1}{\|\boldsymbol{H}\|_F^2}\mathbf{I}_k\tilde{\boldsymbol{x}} = \boldsymbol{x} + \frac{1}{\|\boldsymbol{H}\|_F^2}\tilde{n},$$

which is equivalent to the solution for (3).

For a nonlinearly decodable ST code where $\boldsymbol{x}$ and its complex conjugate $\boldsymbol{x}^*$ are correlated, it is still possible to perform MRC and ML decoding by solving a linear system of augmented symbols given by

$$\begin{bmatrix} \bar{\boldsymbol{y}} \\ \bar{\boldsymbol{y}}^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \\ \boldsymbol{H}_2^* & \boldsymbol{H}_1^* \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}^* \end{bmatrix} + \begin{bmatrix} \bar{\boldsymbol{n}} \\ \bar{\boldsymbol{n}}^* \end{bmatrix}.$$

This type of system is refered to as a widely linear system [17,18]. It can be see that the equivalent channel matrix has four times as many elements of the linear system and is given by

$$\boldsymbol{H}_{ef} = \begin{bmatrix} \boldsymbol{H}_1 & \boldsymbol{H}_2 \\ \boldsymbol{H}_2^* & \boldsymbol{H}_1^* \end{bmatrix}.$$

Then, the MMSE solution to this system is given by the pseudo-inverse of the equivalent channel matrix as before, and the ML decoding of the input symbols is given by the first half of the augmented input vector.

For QOSTBC, while the decoding is more complex than the diagonal decoding presented in (3), the system is still linear so a linear MMSE solution still exists. This increased decoding complexity is offset by its increased data rate over OSTBC, particularly for low E_b/N_0.

B. Examples of STBCs

The examples of STBCs given in this section are all for complex input constellations. These codes were chosen to give a good representation of current orthogonal or quasi-orthgonal codes, which have the advantage of being easily decodable by mobile devices using a linear MMSE solution. These examples only include $N_T \in \{2,4,8\}$, because current 4G standards only define up to eight transmit antennas and codes for other numbers of transmit antennas can be obtained by eliminating one or more columns from the transmission matrix of a code with greater N_T.

1) Alamouti's STBC: The Alamouti STBC uses two transmit antennas and is described by the transmission matrix

$$\boldsymbol{G} = \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix}.$$

This code uses two time slots to transmit two symbols over two antennas, making this a $C(2,2,2)$. The two antennas are assumed to have equal transmit power of $\Omega_p/2$ since CSIT is unknown and so the sum of their powers equals the transmit power of a singular transmit antenna.

For one receive antenna and assuming block-fading, the received symbols are

$$y_1 = h_1 s_1 + h_2 s_2 + n_1$$
$$y_2 = -h_1 s_2^* + h_2 s_1^* + n_2.$$

Taking the complex conjugate of the second equation yields

$$y_1 = h_1 s_1 + h_2 s_2 + n_1$$
$$y_2^* = h_2^* s_1 - h_1^* s_2 + n_2^*,$$

or in matrix form,

$$\begin{bmatrix} y_1 \\ y_2^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}.$$

The MRC output of this system is

$$\begin{aligned}\tilde{s}_1 &= h_1^* y_1 + h_2 y_2^* \\ \tilde{s}_2 &= h_2^* y_1 - h_1 y_2^*,\end{aligned}$$

which can also be written as

$$\begin{aligned}\hat{s}_1 &= (|h_1|^2 + |h_2|^2) s_1 + h_1^* n_1 + h_2 n_2^* \\ \hat{s}_2 &= (|h_1|^2 + |h_2|^2) s_2 - h_1 n_2^* + h_2^* n_1.\end{aligned}$$

The combined symbols have the expected form given by (3). It can be seen that the Alamouti's 2×1 transmit diversity scheme is equivalent to a 1×2 receive divesity scheme with MRC. This result can be extended to any number of receive antennas, where Alamouti's $2 \times N_R$ transmit diversity scheme is equivalent to a $1 \times 2N_R$ diversity with MRC [19].

2) $C(4,4,8)$: For $N_T = 4$, Tarokh et al. proposed a $R = 1/2$ code [10]:

$$\boldsymbol{G} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2 & s_1 & -s_4 & s_3 \\ -s_3 & s_4 & s_1 & -s_2 \\ -s_4 & -s_3 & s_2 & s_1 \\ s_1^* & s_2^* & s_3^* & s_4^* \\ -s_2^* & s_1^* & -s_4^* & s_3^* \\ -s_3^* & s_4^* & s_1^* & -s_2^* \\ -s_4^* & -s_3^* & s_2^* & s_1^* \end{bmatrix}.$$

It can be see that the symbols transmitted during the second set of four time slots are the complex conjugates of those transmitted during the first set. In fact, a transmission matrix made up of the first four rows of this one represents a full-rate code for real constellations. This demonstrates how a $R = 1/2$ code for complex constellations can be generated from any full-rate code for real constellations.

3) $C(4,3,4)$: To increase the bandwidth efficiency, Tarokh et al. also proposed a $R = 3/4$ code [10]:

$$\boldsymbol{G} = \begin{bmatrix} s_1 & s_2 & \frac{s_3}{\sqrt{2}} & \frac{s_3}{\sqrt{2}} \\ -s_2^* s_1^* & \frac{s_3}{\sqrt{2}} & -\frac{s_3}{\sqrt{2}} & \\ \frac{s_3^*}{\sqrt{2}} & \frac{s_3^*}{\sqrt{2}} & \frac{-s_1-s_1^*+s_2-s_2^*}{2} & \frac{-s_2-s_2^*+s_1-s_1^*}{2} \\ \frac{s_3^*}{\sqrt{2}} & -\frac{s_3^*}{\sqrt{2}} & \frac{s_2+s_2^*+s_1-s_1^*}{2} & -\frac{s_1+s_1^*+s_2-s_2^*}{2} \end{bmatrix}$$

Note that this code has unequal transmit power among the antennas, meaning the signal envelope is not constant and the antennas must vary their power each time slot, which is undesirable. An improved version with equal transmit power among the transmitting antennas is [20]

$$\boldsymbol{G} = \begin{bmatrix} s_1 & s_2 & s_3 & 0 \\ -s_2^* & s_1^* & 0 & s_3 \\ s_3^* & 0 & -s_1^* & s_2 \\ 0 & s_3^* & -s_2^* & -s_1 \end{bmatrix}.$$

While this code is orthogonal like the previous example and has a higher rate, it is widely linear rather than linear because there is a correlation between the transmitted symbols and their complex conjugates. Thus, this code trades increased bandwidth efficiency for increased decoding complexity.

4) $C(4,4,4)$: To achieve full-rate for $N_T = 4$ transmit antennas, it is necessary to use a QOSTBC. One version described in [15] has the transmission matrix

$$\boldsymbol{G} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2^* & s_1^* & -s_4^* & s_3^* \\ s_3 & s_4 & s_1 & s_2 \\ -s_4^* & s_3^* & -s_2^* & s_1^* \end{bmatrix}.$$

This code has full-rate but only second-order diversity. In addition, it is linearly decodable.

5) $C(8,8,16)$: For $N_T = 8$ transmit antennas, the authors only know of $R = 1/2$ codes. The first, given in [10], has the transmission matrix

$$\boldsymbol{G} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ -s_2 & s_1 & s_4 & -s_3 & s_6 & -s_5 & -s_8 & s_7 \\ -s_3 & -s_4 & s_1 & s_2 & s_7 & s_8 & -s_5 & -s_6 \\ -s_4 & s_3 & -s_2 & s_1 & s_8 & -s_7 & s_6 & -s_5 \\ -s_5 & -s_6 & -s_7 & -s_8 & s_1 & s_2 & s_3 & s_4 \\ -s_6 & s_5 & -s_8 & s_7 & -s_2 & s_1 & -s_4 & s_3 \\ -s_7 & s_8 & s_5 & -s_6 & -s_3 & s_4 & s_1 & -s_2 \\ -s_8 & -s_7 & s_6 & s_5 & -s_4 & -s_3 & s_2 & s_1 \\ s_1^* & s_2^* & s_3^* & s_4^* & s_5^* & s_6^* & s_7^* & s_8^* \\ -s_2^* & s_1^* & s_4^* & -s_3^* & s_6^* & -s_5^* & -s_8^* & s_7^* \\ -s_3^* & -s_4^* & s_1^* & s_2^* & s_7^* & s_8^* & -s_5^* & -s_6^* \\ -s_4^* & s_3^* & -s_2^* & s_1^* & s_8^* & -s_7^* & s_6^* & -s_5^* \\ -s_5^* & -s_6^* & -s_7^* & -s_8^* & s_1^* & s_2^* & s_3^* & s_4^* \\ -s_6^* & s_5^* & -s_8^* & s_7^* & -s_2^* & s_1^* & -s_4^* & s_3^* \\ -s_7^* & s_8^* & s_5^* & -s_6^* & -s_3^* & s_4^* & s_1^* & -s_2^* \\ -s_8^* & -s_7^* & s_6^* & s_5^* & -s_4^* & -s_3^* & s_2^* & s_1^* \end{bmatrix}$$

6) $C(8,4,8)$: Another code for $N_T = 8$ is [16]

$$\boldsymbol{G} = \begin{bmatrix} s_1 & 0 & 0 & 0 & -s_4^* & 0 & -s_2^* & s_3^* \\ 0 & s_1 & 0 & 0 & 0 & -s_4^* & -s_3 & -s_2 \\ 0 & 0 & s_1 & 0 & s_2 & s_3^* & -s_4 & 0 \\ 0 & 0 & 0 & s_1 & -s_3 & s_2^* & 0 & -s_4 \\ s_4 & 0 & -s_2^* & s_3^* & s_1^* & 0 & 0 & 0 \\ 0 & s_4 & -s_3 & -s_2 & 0 & s_1^* & 0 & 0 \\ s_2 & s_3^* & s_4^* & 0 & 0 & 0 & s_1^* & 0 \\ -s_3 & s_2^* & 0 & s_4^* & 0 & 0 & 0 & s_1^* \end{bmatrix}$$

While the previous two codes both have the same rate, it is expected that the $C(8,8,16)$ code will perform better since its transmission matrix is not sparse like that of $C(8,4,8)$ code, a greater diversity is expected.

IV. Methodology

This study investigated the performance of eight STBC codes for a STBC-MIMO system transmitting over a Rayleigh flat block-fading channel with AWGN using software simulation. The eight MIMO cases of 1×1, 1×2 with MRC, 2×1 and 2×2 using Alamouti's $C(2,2,2)$ code, 4×2 using $C(4,2,4)$, $C(4,3,4)$ and $C(4,4,4)$, and 8×2 using $C(8,8,16)$ are considered for the four modulation methods of BPSK, QPSK, 16-QAM, and 64-QAM. The MIMO and modulation choices represent realistic possibilities for current 4G systems. BPSK, while not in current 4G standards, was included for better comparison between half-rate and full-rate codes.

The channel was modeled as a discrete-time baseband-equivalent channel, and it was assumed that there was no ISI and ICI. The channel coherence time and time slot period were chosen to be $T_c = T = 1/(15\text{ kHz})$ to represent flat fading, where 15 kHz is the subcarrier spacing for LTE. In addition, block fading was assumed, with the channel quasi-static over $4T$ time slots.

At the transmitter, the modulated symbols were scaled to have an average energy of unity. For simplicity, OFDM was not used, because in theory IFFT at the transmitter and FFT at the receiver have no effect on the performance of a linear system according to Parseval's theorem. Then, the modulated symbols are transmitted over a Rayleigh faded channel with Doppler frequencies corresponding to speeds of v = 0, 5, 15, and 45 m/s for a carrier frequency of f_c = 1800 MHz. For the stationary case, the uncorrelated path gains were directly generated. For the moving cases, they were generated according to the statistical method for multiple uncorrelated faded envelopes given by [19]. In addition, white Gaussian noise is added the output of the Rayleigh channel, with the energy of the noise scaled to give the desired E_b/N_0.

It is assumed that CSIR is known, so at the receiver, the pseudo-inverse of the equivalent channel matrix $\boldsymbol{H}_{ef}^{\dagger}$ is used to perform the MRC. In the case of the widely linear code $C(4,3,4)$, additional steps are needed to generate the augmented output matrix and extract the desired symbols from the augmented input matrix. This is followed by ML demodulation of the combined symbols.

To evaluate the performance of the STBC-MIMO system, the bit error rate (BER) P_b is measured and plotted versus the bit energy-to-noise ratio E_b/N_0. The simulation was performed using packets of 120 symbols and 10^5 total packets were used per data point. The range of E_b/N_0 values considered was from 0 to 20 dB with a spacing of 2 dB.

The simulations were done in a mixture of Matlab versions 2012b, 2014a, 2014b, and 2015a. Care was taken to ensure results were comparable by using the same seed to the random number generator for each run to ensure repeatability in any Matlab version. The modulation and demodulation as well as the AWGN used built-in Matlab functions, while the STBC encoding and decoding and the generation of the Rayleigh fading channel for the various Doppler frequencies used code written by the authors. These Matlab functions are included in the appendix of this paper along with a sample script showing how to use the functions to duplicate the simulation results.

V. Results

VI. Summary

References

[1] 3GPP, "Feasibility study for Further Advancements for E-UTRA (LTE-Advanced)," 3rd Generation Partnership Project (3GPP), TR 36.912, 06 2010. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36912.htm

[2] G. Caire and S. Shamai, "On the achievable throughput of a multi-antenna Gaussian broadcast channel," *IEEE Trans. Inf. Theory*, July 2003.

[3] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inf. Theory*, vol. 52, pp. 3936–3964, Sept 2006.

[4] M. Nasseri and H. Bakhshi, "Iterative channel estimation algorithm in multiple input multiple output orthogonal frequency division multiplexing systems," *J. of Comput. Sci.*, vol. 6, no. 2, pp. 224–228, 2010.

[5] E. Björnson and B. Ottersten, "A framework for training-based estimation in arbitrary correlated Rician MIMO channels with Rician disturbance," *IEEE Trans. Signal Process.*, vol. 58, pp. 1807–1820, Mar 2010.

[6] M. Biguesh and A. Gershman, "Training-based MIMO channel estimation: a study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, pp. 884–893, Mar 2006.

[7] G. J. Foschini, "Layered space-time architecture for wireless communications in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.

[8] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance analysis and code construction," *IEEE Trans. Inf. Theory*, vol. 44, pp. 744–765, Mar 1998.

[9] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, 1998.

[10] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1456–1458, July 1999.

[11] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications," *IEEE Signal Processing Mag.*, vol. 17, no. 3, pp. 29–48, 2000.

[12] Z. Wang, X. Ma, and G. B. Giannakis, “OFDM or single-carrier block transmissions?” *IEEE Trans. Commun.*, vol. 52, pp. 380–394, Mar 2004.
[13] J. Seberry, S. A. Spence, and T. A. Wysocki, “A construction technique for generalized complex orthogonal designs and applications to wireless communications,” *Linear algebra and its applications*, vol. 405, pp. 163–176, 2005.
[14] W. Wolfe, “Amicable orthogonal designs—existence,” *Canad. J. Math.*, vol. 28, no. 5, pp. 1006–1020, 1976.
[15] H. Jafarkhani, *Space-time coding: theory and practice*. New York: Cambridge University Press, 2005.
[16] E. G. Larsson and P. Stoica, *Space-time block coding for wireless communications*. New York: Cambridge University Press, 2003.
[17] Y. A. S. Dama, R. A. Abd-alhameed, T. S. Ghazaany, and S. Zhu, “A new approach for OSTBC and QOSTBC,” *Int. J. of Comput. Applicat.*, vol. 67, no. 6, pp. 45–48, 2013.
[18] B. Picinbono and P. Chevalier, “Widely linear estimation with complex data,” *IEEE Trans. Signal Process.*, vol. 43, pp. 2030–2033, Aug 1995.
[19] G. L. Stüber, *Principles of mobile communications*, 3rd ed. New York: Springer Science+Business Media, 2012.
[20] G. Ganesan and P. Stoica, “Space-time block codes: a maximum SNR approach,” *IEEE Trans. Inf. Theory*, vol. 47, pp. 1650–1656, May 2001.

APPENDIX

The appendix is organized as follows. The first two sections contain functions for simulating the STBC-MIMO channel for the eight STBCs and any given modulation and demodulation methods. The code in the rest of the sections implement encoding and decoding of the various STBCs. The following sample script shows how to perform a simulation run.

```
%% simulation parameters
numPackets = 1e5; % each packets contains 120 symbols
EbNo = 0:2:30; % in dB
v = 5; % speed for calculating Doppler frequency in m/s

%% calculate normalized Doppler frequency
Ts = 1/15e3; % symbol rate is 15 kHz = sub-carrier spacing for LTE
fc = 1800e6; % carrier freq of 1800 MHz
fm_norm = v*fc/3e8*Ts; % normalized max. Doppler freq. (fm*Ts)

%% modulation method
% 16-QAM shown
M = 16;
hMod = comm.RectangularQAMModulator(M, ...
    'NormalizationMethod', 'Average power', 'BitInput', true);
hDemod = comm.RectangularQAMDemodulator(M, ...
    'NormalizationMethod', 'Average power', 'BitOutput', true);
% uncomment below lines for PSK
% hMod = comm.PSKModulator(M, 'PhaseOffset', 0, ...
%     'BitInput', true);
% hDemod = comm.PSKDemodulator(M, 'PhaseOffset', 0, ...
%     'BitOutput', true);

%% Run simulation
% stationary MS
% ber = sim_flatRayleigh(EbNo,numPackets,M,hMod,hDemod);
% moving MS
ber = sim_flatRayleighDoppler(EbNo,numPackets,M,hMod,hDemod, ...
    fm_norm);
```

Appendix A
Stationary Receiver

```
%% Comparison of MIMO methods for flat Rayleigh fading channel
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function ber = sim_flatRayleigh(EbNo,numPackets,Mmod,hMod,hDemod)

charCount = 0;
wb = 0;
fprintf('Running... ')
call_wb = @(x,count) fprintf([repmat('\b',1,count) ...
    '%0.1f%%, t=%f'], ...
    round(x*1000)/10, toc);
wb_step = numPackets/100;

% parameters
frameLen = 120;
blockLen = 4;
N = 8; % max Tx
M = 2; % max Rx
% preallocate channel
minrate = 1/2;
H = zeros(frameLen/minrate,N,M);

% Set the global random stream for repeatability
s = RandStream.create('mt19937ar', 'seed',55408);
prevStream = RandStream.setGlobalStream(s);

%% Modulation
hAWGN1Rx = comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/↙
   ↳ No)', ...
    'SignalPower',1,'BitsPerSymbol',Mmod);
hAWGN2Rx = clone(hAWGN1Rx);
hAWGN2Rx_r34 = clone(hAWGN1Rx);
hAWGN2Rx_r12 = clone(hAWGN1Rx);

hError11 = comm.ErrorRate;
hError21 = comm.ErrorRate;
hError12 = comm.ErrorRate;
hError22 = comm.ErrorRate;
hError42 = comm.ErrorRate;
hError42q = comm.ErrorRate;
hError42_12 = comm.ErrorRate;
hError82 = comm.ErrorRate;
ber11 = zeros(3,length(EbNo));
```

```
ber21 = zeros(3,length(EbNo));
ber12 = zeros(3,length(EbNo));
ber22 = zeros(3,length(EbNo));
ber42 = zeros(3,length(EbNo));
ber42q = zeros(3,length(EbNo));
ber42_12 = zeros(3,length(EbNo));
ber82 = zeros(3,length(EbNo));

%% Transmission
% Assume Rayleigh fading, no shadowing, no Doppler

for idx = 1:length(EbNo)
    reset(hError11);
    reset(hError21);
    reset(hError12);
    reset(hError22);
    reset(hError42);
    reset(hError42q);
    reset(hError42_12);
    reset(hError82);
    hAWGN1Rx.EbNo = EbNo(idx);
    hAWGN2Rx.EbNo = EbNo(idx);
    hAWGN2Rx_r12.EbNo = EbNo(idx);
    hAWGN2Rx_r34.EbNo = EbNo(idx);

    for packetIdx = 1:numPackets
        data = randi([0 1],frameLen*log2(Mmod),1);
        modData = step(hMod, data);

        %%% Flat Rayleigh fading
        H(1:blockLen:end,:,:) = (randn(frameLen/minrate/blockLen,N,M) ↙
            ↪ ...
            + 1j*randn(frameLen/minrate/blockLen,N,M))/sqrt(2);
        for k = 2:blockLen
            H(k:blockLen:end,:,:) = H(1:blockLen:end,:,:);
        end

        %%% SISO
        H11 = H(1:frameLen,1,1);
        chanOut11 = H11 .* modData;
        rxSig11 = step(hAWGN1Rx, chanOut11);
        demod11 = step(hDemod, rxSig11./H11);

        %%% MIMO: 1x2 MRC
        H12 = H(1:frameLen,1,:);
        chanOut12 = squeeze(sum(H12 .* repmat(modData, [1 1 2]), 2));
        rxSig12 = step(hAWGN2Rx, chanOut12);
        decData12 = fun_MRCx2(rxSig12, H12);
        demod12 = step(hDemod, decData12);

        %%% MIMO: 2x1 Alamouti
        encData = fun_AlamoutiEnc(modData);
        H21 = H(1:frameLen,1:2,1)/sqrt(2);
        chanOut21 = sum(H21.*encData,2);
        rxSig21 = step(hAWGN1Rx, chanOut21);
        decData21 = fun_AlamoutiDec2x1(rxSig21, H21);
        demod21 = step(hDemod, decData21);

        %%% MIMO: 2x2 Alamouti
        H22 = H(1:frameLen,1:2,:)/sqrt(2);
        chanOut22 = squeeze(sum(H22 .* repmat(encData, [1 1 2]), 2));
        rxSig22 = step(hAWGN2Rx, chanOut22);
        decData22 = fun_AlamoutiDec2x2(rxSig22, H22);
        demod22 = step(hDemod, decData22);

        %%% MIMO: 4x2 OSTBC, rate=1/2
        H42_12 = H(1:2*frameLen,1:4,:)/sqrt(4);
        encData = fun_OSTBC2Enc4x(modData);
        chanOut42_12 = squeeze(sum(H42_12 .* repmat(encData, [1 1 2]), 2));
        rxSig42_12 = step(hAWGN2Rx_r12, chanOut42_12);
        decData42_12 = fun_OSTBC2Dec4x2(rxSig42_12, H42_12);
        demod42_12 = step(hDemod, decData42_12);

        %%% MIMO: 4x2 OSTBC, rate=3/4
        H42 = H(1:4/3*frameLen,1:4,:)/sqrt(4);
        encData = fun_OSTBCEnc4x(modData);
        chanOut42 = squeeze(sum(H42 .* repmat(encData, [1 1 2]), 2));
        rxSig42 = step(hAWGN2Rx_r34, chanOut42);
        decData42 = fun_OSTBCDec4x2(rxSig42, H42);
        demod42 = step(hDemod, decData42);

        %%% MIMO: 4x2 QOSTBC, rate=1
        H42q = H(1:frameLen,1:4,:)/sqrt(4);
        encData = fun_QOSTBCEnc4x(modData);
        chanOut42q = squeeze(sum(H42q .* repmat(encData, [1 1 2]), 2));
        rxSig42q = step(hAWGN2Rx, chanOut42q);
        decData42q = fun_QOSTBCDec4x2(rxSig42q, H42q);
        demod42q = step(hDemod, decData42q);

        %%% MIMO: 8x2 OSTBC, rate=1/2
        H82 = H(1:2*frameLen,1:8,:)/sqrt(8);
        encData = fun_OSTBC2Enc8x(modData);
        chanOut82 = squeeze(sum(H82 .* repmat(encData, [1 1 2]), 2));
        rxSig82 = step(hAWGN2Rx_r12, chanOut82);
        decData82 = fun_OSTBC2Dec8x2(rxSig82, H82);
        demod82 = step(hDemod, decData82);

        % get BER
        ber11(:,idx) = step(hError11, data, demod11);
        ber21(:,idx) = step(hError21, data, demod21);
        ber12(:,idx) = step(hError12, data, demod12);
        ber22(:,idx) = step(hError22, data, demod22);
        ber42(:,idx) = step(hError42, data, demod42);
        ber42q(:,idx) = step(hError42q, data, demod42q);
        ber42_12(:,idx) = step(hError42_12, data, demod42_12);
        ber82(:,idx) = step(hError82, data, demod82);

        % waitbar
        if mod(packetIdx,wb_step) == 0
            wb = wb + wb_step/length(EbNo)/numPackets;
            charCount = call_wb(wb,charCount) - charCount;
        end
    end
    % waitbar
    wb = idx/length(EbNo);
end

% Restore default stream
RandStream.setGlobalStream(prevStream);

%% Collect results
ber = {ber11;ber21;ber12;ber22;ber42_12;ber42;ber42q;ber82};
ber = cellfun(@(x) x(1,:), ber, 'uni', 0);
```

Appendix B
Mobile Receiver

```
%% Comparison of MIMO methods for flat Rayleigh fading channel with ↙
   ↳ Doppler shift
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function ber = sim_flatRayleighDoppler(EbNo,numPackets,Mmod,hMod,↙
   ↳ hDemod,fm_norm)

charCount = 0;
wb = 0;
fprintf('Running... ')
call_wb = @(x,count) fprintf([repmat('\b',1,count) ...
    '%0.1f%%, t=%f'], ...
    round(x*1000)/10, toc);
```

```
wb_step = numPackets/100;

% parameters
frameLen = 120;
blockLen = 4;
N = 8; % max Tx
M = 2; % max Rx
% preallocate channel
minrate = 1/2;
H = zeros(frameLen/minrate,N,M);

% Set the global random stream for repeatability
s = RandStream.create('mt19937ar', 'seed',55408);
prevStream = RandStream.setGlobalStream(s);

%% Modulation
hAWGN1Rx = comm.AWGNChannel('NoiseMethod','Signal to noise ratio (Eb/
    No)', ...
    'SignalPower',1,'BitsPerSymbol',Mmod);
hAWGN2Rx = clone(hAWGN1Rx);
hAWGN2Rx_r34 = clone(hAWGN1Rx);
hAWGN2Rx_r12 = clone(hAWGN1Rx);

hError11 = comm.ErrorRate;
hError21 = comm.ErrorRate;
hError12 = comm.ErrorRate;
hError22 = comm.ErrorRate;
hError42 = comm.ErrorRate;
hError42q = comm.ErrorRate;
hError42_12 = comm.ErrorRate;
hError82 = comm.ErrorRate;
ber11 = zeros(3,length(EbNo));
ber21 = zeros(3,length(EbNo));
ber12 = zeros(3,length(EbNo));
ber22 = zeros(3,length(EbNo));
ber42 = zeros(3,length(EbNo));
ber42q = zeros(3,length(EbNo));
ber42_12 = zeros(3,length(EbNo));
ber82 = zeros(3,length(EbNo));

%% Transmission
% Assume Rayleigh fading, no shadowing, no Doppler

for idx = 1:length(EbNo)
    reset(hError11);
    reset(hError21);
    reset(hError12);
    reset(hError22);
    reset(hError42);
    reset(hError42q);
    reset(hError42_12);
    reset(hError82);
    hAWGN1Rx.EbNo = EbNo(idx);
    hAWGN2Rx.EbNo = EbNo(idx);
    hAWGN2Rx_r12.EbNo = EbNo(idx);
    hAWGN2Rx_r34.EbNo = EbNo(idx);

    for packetIdx = 1:numPackets
        data = randi([0 1],frameLen*log2(Mmod),1);
        modData = step(hMod, data);

        %%% Flat Rayleigh fading with Doppler shift (statistical model↙
           ↳ )
        alpha = -pi + 2*pi*randn(1,2*N,1);
        beta = -pi + 2*pi*randn(1,2*N,2*N);
        phi = -pi + 2*pi*randn(1,2*N,2*N);
        theta = bsxfun(@plus, pi/(4*N)*reshape(1:2*N,[1 1 2*N]), ...
            pi/(32*N^2)*(0:2*N-1) + (alpha-pi)/(8*N));
        tmp = bsxfun(@plus, phi, bsxfun(@times, cos(theta), ...
            2*pi*fm_norm*(0:frameLen/minrate/blockLen-1).'));
        gI = sum( bsxfun(@times, cos(beta), cos(tmp)), 3);
        gQ = sum( bsxfun(@times, sin(beta), sin(tmp)), 3);
        H(1:blockLen:end,:,:) = reshape((gI + 1j*gQ)/sqrt(2), ...
            frameLen/minrate/blockLen,N,M);
        for k = 2:blockLen
            H(k:blockLen:end,:,:) = H(1:blockLen:end,:,:);
        end

        %%% SISO
        H11 = H(1:frameLen,1,1);
        chanOut11 = H11 .* modData;
        rxSig11 = step(hAWGN1Rx, chanOut11);
        demod11 = step(hDemod, rxSig11./H11);

        %%% MIMO: 1x2 MRC
        H12 = H(1:frameLen,1,:);
        chanOut12 = squeeze(sum(H12 .* repmat(modData, [1 1 2]), 2));
        rxSig12 = step(hAWGN2Rx, chanOut12);
        decData12 = fun_MRCx2(rxSig12, H12);
        demod12 = step(hDemod, decData12);

        %%% MIMO: 2x1 Alamouti
        encData = fun_AlamoutiEnc(modData);
        H21 = H(1:frameLen,1:2,1)/sqrt(2);
        chanOut21 = sum(H21.*encData,2);
        rxSig21 = step(hAWGN1Rx, chanOut21);
        decData21 = fun_AlamoutiDec2x1(rxSig21, H21);
        demod21 = step(hDemod, decData21);

        %%% MIMO: 2x2 Alamouti
        H22 = H(1:frameLen,1:2,:)/sqrt(2);
        chanOut22 = squeeze(sum(H22 .* repmat(encData, [1 1 2]), 2));
        rxSig22 = step(hAWGN2Rx, chanOut22);
        decData22 = fun_AlamoutiDec2x2(rxSig22, H22);
        demod22 = step(hDemod, decData22);

        %%% MIMO: 4x2 OSTBC, rate=1/2
        H42_12 = H(1:2*frameLen,1:4,:)/sqrt(4);
        encData = fun_OSTBC2Enc4x(modData);
        chanOut42_12 = squeeze(sum(H42_12 .* repmat(encData, [1 1 2]),↙
            ↳  2));
        rxSig42_12 = step(hAWGN2Rx_r12, chanOut42_12);
        decData42_12 = fun_OSTBC2Dec4x2(rxSig42_12, H42_12);
        demod42_12 = step(hDemod, decData42_12);

        %%% MIMO: 4x2 OSTBC, rate=3/4
        H42 = H(1:4/3*frameLen,1:4,:)/sqrt(4);
        encData = fun_OSTBCEnc4x(modData);
        chanOut42 = squeeze(sum(H42 .* repmat(encData, [1 1 2]), 2));
        rxSig42 = step(hAWGN2Rx_r34, chanOut42);
        decData42 = fun_OSTBCDec4x2(rxSig42, H42);
        demod42 = step(hDemod, decData42);

        %%% MIMO: 4x2 QOSTBC, rate=1
        H42q = H(1:frameLen,1:4,:)/sqrt(4);
        encData = fun_QOSTBCEnc4x(modData);
        chanOut42q = squeeze(sum(H42q .* repmat(encData, [1 1 2]), 2))↙
            ↳ ;
        rxSig42q = step(hAWGN2Rx, chanOut42q);
        decData42q = fun_QOSTBCDec4x2(rxSig42q, H42q);
        demod42q = step(hDemod, decData42q);

        %%% MIMO: 8x2 OSTBC, rate=1/2
        H82 = H(1:2*frameLen,1:8,:)/sqrt(8);
        encData = fun_OSTBC2Enc8x(modData);
        chanOut82 = squeeze(sum(H82 .* repmat(encData, [1 1 2]), 2));
        rxSig82 = step(hAWGN2Rx_r12, chanOut82);
        decData82 = fun_OSTBC2Dec8x2(rxSig82, H82);
        demod82 = step(hDemod, decData82);

        % get BER
        ber11(:,idx) = step(hError11, data, demod11);
        ber21(:,idx) = step(hError21, data, demod21);
        ber12(:,idx) = step(hError12, data, demod12);
        ber22(:,idx) = step(hError22, data, demod22);
        ber42(:,idx) = step(hError42, data, demod42);
        ber42q(:,idx) = step(hError42q, data, demod42q);
        ber42_12(:,idx) = step(hError42_12, data, demod42_12);
        ber82(:,idx) = step(hError82, data, demod82);

        % waitbar
        if mod(packetIdx,wb_step) == 0
            wb = wb + wb_step/length(EbNo)/numPackets;
            charCount = call_wb(wb,charCount) - charCount;
        end
    end
    % waitbar
    wb = idx/length(EbNo);
end

% Restore default stream
RandStream.setGlobalStream(prevStream);

%% Collect results
ber = {ber11;ber21;ber12;ber22;ber42_12;ber42;ber42q;ber82};
ber = cellfun(@(x) x(1,:), ber, 'uni', 0);
```

APPENDIX C
MRC: 1×2

```
%% MRC Decoder for 1x2 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_MRCx2(x,chan)

N = size(x,1);
y = zeros(N,1);

for idx = 1:N
    y(idx) = chan(idx,:).'\x(idx,:).';
end
```

APPENDIX D
ALAMOUTI ENCODING

```
%% Alamouti Encoder
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

```

```
function y = fun_AlamoutiEnc(x)

N = length(x);
y = zeros(N,2);

y(1:2:end,1) = x(1:2:end);
y(2:2:end,1) = -conj(x(2:2:end));
y(1:2:end,2) = x(2:2:end);
y(2:2:end,2) = conj(x(1:2:end));
```

Appendix E
Alamouti Decoding: 2×1

```
%% Alamouti Decoder for 2x1 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_AlamoutiDec2x1(x,chan)

N = size(x,1);
y = zeros(N,1);

x(2:2:end) = conj(x(2:2:end));
for idx = 1:2:N
    H = [chan(idx,:); [conj(chan(idx+1,2)) -conj(chan(idx+1,1))]];
    y(idx:idx+1) = H\x(idx:idx+1);
end
```

Appendix F
Alamouti Decoding: 2×2

```
%% Alamouti Decoder for 2x2 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_AlamoutiDec2x2(x,chan)

N = size(x,1);
y = zeros(N,1);

x(2:2:end,:) = conj(x(2:2:end,:));
for idx = 1:2:N
    H = [chan(idx,:,1); chan(idx,:,2); ...
        conj(chan(idx+1,2,1)) -conj(chan(idx+1,1,1)); ...
        conj(chan(idx+1,2,2)) -conj(chan(idx+1,1,2))];
    y(idx:idx+1) = H\reshape(x(idx:idx+1,:).',4,1);
```

```
end
```

APPENDIX G
$C(4,4,8)$ ENCODING

```
%% OSTBC Encoder for 4 Tx
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_OSTBC2Enc4x(x)
% Lt = 4, R = 1/2 code
% C = [ s1   s2   s3   s4;
%      -s2   s1  -s4   s3;
%      -s3   s4   s1  -s2;
%      -s4  -s3   s2   s1;
%       s1*  s2*  s3*  s4*;
%      -s2*  s1* -s4*  s3*;
%      -s3*  s4*  s1* -s4*;
%      -s4* -s3*  s2*  s1*;

N = length(x);
y = zeros(2*N,4);
x1 = x(1:4:end);
x2 = x(2:4:end);
x3 = x(3:4:end);
x4 = x(4:4:end);

y(1:8:end,:) = [x1 x2 x3 x4];
y(2:8:end,:) = [-x2 x1 -x4 x3];
y(3:8:end,:) = [-x3 x4 x1 -x2];
y(4:8:end,:) = [-x4 -x3 x2 x1];
for k = 5:8
    y(k:8:end,:) = conj(y(k-4:8:end,:));
end
```

APPENDIX H
$C(4,4,8)$ DECODING: 4×2

```
%% OSTBC Decoder for 4x2 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_OSTBC2Dec4x2(x,chan)
% Decoder for
% Lt = 4, R = 1/2 code
% C = [ s1   s2   s3   s4;
%       -s2   s1  -s4   s3;
%       -s3   s4   s1  -s2;
%       -s4  -s3   s2   s1;
%        s1*  s2*  s3*  s4*;
%       -s2*  s1* -s4*  s3*;
%       -s3*  s4*  s1* -s4*;
%       -s4* -s3*  s2*  s1;

N = size(x,1);
% calculate y with 0 every 5th-8th element
y = zeros(N,1);

inds = [5:8:N; 6:8:N; 7:8:N; 8:8:N];
x(inds,:) = conj(x(inds,:));
for idx = 1:8:N
    H1 = [chan(idx,:,1); chan(idx,:,2);
        chan(idx+1,2,1) -chan(idx+1,1,1) chan(idx+1,4,1) -chan(idx↙
           ↪ +1,3,1);
        chan(idx+1,2,2) -chan(idx+1,1,2) chan(idx+1,4,2) -chan(idx↙
           ↪ +1,3,2);
        chan(idx+2,3,1) -chan(idx+2,4,1) -chan(idx+2,1,1) chan(idx↙
           ↪ +2,2,1);
        chan(idx+2,3,2) -chan(idx+2,4,2) -chan(idx+2,1,2) chan(idx↙
           ↪ +2,2,2);
        chan(idx+3,4,1) chan(idx+3,3,1) -chan(idx+3,2,1) -chan(idx↙
           ↪ +3,1,1);
        chan(idx+3,4,2) chan(idx+3,3,2) -chan(idx+3,2,2) -chan(idx↙
           ↪ +3,1,2)];
    H2 = [chan(idx+4,:,1); chan(idx+4,:,2);
        chan(idx+5,2,1) -chan(idx+5,1,1) chan(idx+5,4,1) -chan(idx↙
           ↪ +5,3,1);
        chan(idx+5,2,2) -chan(idx+5,1,2) chan(idx+5,4,2) -chan(idx↙
           ↪ +5,3,2);
        chan(idx+6,3,1) -chan(idx+6,4,1) -chan(idx+6,1,1) chan(idx↙
           ↪ +6,2,1);
        chan(idx+6,3,2) -chan(idx+6,4,2) -chan(idx+6,1,2) chan(idx↙
           ↪ +6,2,2);
        chan(idx+7,4,1) chan(idx+7,3,1) -chan(idx+7,2,1) -chan(idx↙
           ↪ +7,1,1);
        chan(idx+7,4,2) chan(idx+7,3,2) -chan(idx+7,2,2) -chan(idx↙
           ↪ +7,1,2)];
    y(idx:idx+3) = [H1; conj(H2)]\reshape(x(idx:idx+7,:).',16,1);
end

% remove extra entries
y(inds) = [];
```

Appendix I
$C(4,3,4)$ Encoding

```
%% OSTBC Encoder for 4 Tx
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_OSTBCEnc4x(x)
% Lt = 4, R = 3/4 code
% C = [ s1   s2   s3    0;
%      -s2*  s1*   0   s3;
%       s3*   0  -s1*  s2;
%        0   s3* -s2* -s1]

N = length(x);
y = zeros(4/3*N,4);
x1 = x(1:3:end);
x2 = x(2:3:end);
x3 = x(3:3:end);

y(1:4:end,1) = x1;
y(1:4:end,2) = x2;
y(1:4:end,3) = x3;

y(2:4:end,1) = -conj(x2);
y(2:4:end,2) = conj(x1);
y(2:4:end,4) = x3;

y(3:4:end,1) = conj(x3);
y(3:4:end,3) = -conj(x1);
y(3:4:end,4) = x2;

y(4:4:end,2) = conj(x3);
y(4:4:end,3) = -conj(x2);
y(4:4:end,4) = -x1;
```

Appendix J
$C(4,3,4)$ Decoding: 4×2

```
%% OSTBC Decoder for 4x2 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_OSTBCDec4x2(x,chan)
% Decoder for
% Lt = 4, R = 3/4 code
% C = [ s1   s2   s3    0;
%      -s2*  s1*   0   s3;
%       s3*   0  -s1*  s2;
%        0   s3* -s2* -s1]
%
% Decode using augmented matrix
% [y y*].' = H*[x x*].'
% where H = [H1 H2; H2* H1*]

N = size(x,1);
% calculate y with 0 every 4th element
y = zeros(N,1);

for idx = 1:4:N
    H1 = [chan(idx,1:3,1); chan(idx,1:3,2);
        0 0 chan(idx+1,4,1); 0 0 chan(idx+1,4,2);
        0 chan(idx+2,4,1) 0; 0 chan(idx+3,4,2) 0;
        -chan(idx+3,4,1) 0 0; -chan(idx+3,4,2) 0 0];
    H2 = [zeros(2,3);
        chan(idx+1,2,1) -chan(idx+1,1,1) 0;
        chan(idx+1,2,2) -chan(idx+1,1,2) 0;
        -chan(idx+2,3,1) 0 chan(idx+2,1,1);
        -chan(idx+2,3,2) 0 chan(idx+2,1,2);
        0 -chan(idx+3,3,1) chan(idx+3,2,1);
        0 -chan(idx+3,3,2) chan(idx+3,2,2)];
    H = [H1 H2; conj(H2) conj(H1)];
    xblk = x(idx:idx+3,:).';
    yAug = H\[xblk(:); conj(xblk(:))];
    y(idx:idx+2) = yAug(1:3);
end

% remove extra entries
y(4:4:end) = [];
```

Appendix K
$C(4,4,4)$ Encoding

```
%% QOSTBC Encoder for 4 Tx
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_QOSTBCEnc4x(x)
% Lt = 4, R = 1 code
% C = [ s1   s2   s3   s4;
%      -s2*  s1* -s4*  s3*;
%       s3   s4   s1   s2;
%      -s4*  s3* -s2*  s1*]
```

```

N = length(x);
y = zeros(N,4);
x1 = x(1:4:end);
x2 = x(2:4:end);
x3 = x(3:4:end);
x4 = x(4:4:end);

y(1:4:end,:) = [x1 x2 x3 x4];
y(2:4:end,:) = conj([-x2 x1 -x4 x3]);
y(3:4:end,:) = [x3 x4 x1 x2];
y(4:4:end,:) = conj([-x4 x3 -x2 x1]);
```

Appendix L
$C(4,4,4)$ Decoding: 4×2

```
%% QOSTBC Decoder for 4x2 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_QOSTBCDec4x2(x,chan)
% Decoder for
% Lt = 4, R = 1 code
% C = [ s1   s2   s3   s4;
%      -s2*  s1* -s4*  s3*;
%       s3   s4   s1   s2;
%      -s4*  s3* -s2*  s1*]

N = size(x,1);
y = zeros(N,1);

x(2:4:end,:) = conj(x(2:4:end,:));
x(4:4:end,:) = conj(x(4:4:end,:));
for idx = 1:4:N
    H = [chan(idx,:,1); chan(idx,:,2);
        chan(idx+1,2,1) -chan(idx+1,1,1) chan(idx+1,4,1) -chan(idx+1,3,1);
        chan(idx+1,2,2) -chan(idx+1,1,2) chan(idx+1,4,2) -chan(idx+1,3,2);
        chan(idx+2,[3 4 1 2],1); chan(idx+2,[3 4 1 2],2);
        chan(idx+3,4,1) -chan(idx+3,3,1) chan(idx+3,2,1) -chan(idx+3,1,1);
        chan(idx+3,4,2) -chan(idx+3,3,2) chan(idx+3,2,2) -chan(idx+3,1,2)];
    H([3 4 7 8],:) = conj(H([3 4 7 8],:));
    y(idx:idx+3) = H\reshape(x(idx:idx+3,:).',8,1);
end
```

Appendix M
$C(8,8,16)$ Encoding

```
%% OSTBC Encoder for 4 Tx
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_OSTBC2Enc4x(x)
% Lt = 4, R = 1/2 code
% C = [ s1   s2   s3   s4;
%      -s2   s1  -s4   s3;
%      -s3   s4   s1  -s2;
%      -s4  -s3   s2   s1;
%       s1*  s2*  s3*  s4*;
%      -s2*  s1* -s4*  s3*;
%      -s3*  s4*  s1* -s4*;
%      -s4* -s3*  s2*  s1*;

N = length(x);
y = zeros(2*N,4);
x1 = x(1:4:end);
x2 = x(2:4:end);
x3 = x(3:4:end);
x4 = x(4:4:end);

y(1:8:end,:) = [x1 x2 x3 x4];
y(2:8:end,:) = [-x2 x1 -x4 x3];
y(3:8:end,:) = [-x3 x4 x1 -x2];
y(4:8:end,:) = [-x4 -x3 x2 x1];
for k = 5:8
    y(k:8:end,:) = conj(y(k-4:8:end,:));
end
```

Appendix N
$C(8,8,16)$ Decoding: 8×2

```
%% OSTBC Decoder for 4x2 MIMO
%
% ECE 6604: 4G MIMO Research Project
% Klaus Okkelberg and Abhishek Obla Hema

function y = fun_OSTBC2Dec4x2(x,chan)
% Decoder for
% Lt = 4, R = 1/2 code
% C = [ s1   s2   s3   s4;
%      -s2   s1  -s4   s3;
%      -s3   s4   s1  -s2;
%      -s4  -s3   s2   s1;
%       s1*  s2*  s3*  s4*;
%      -s2*  s1* -s4*  s3*;
%      -s3*  s4*  s1* -s4*;
%      -s4* -s3*  s2*  s1;

N = size(x,1);
% calculate y with 0 every 5th-8th element
y = zeros(N,1);

inds = [5:8:N; 6:8:N; 7:8:N; 8:8:N];
x(inds,:) = conj(x(inds,:));
for idx = 1:8:N
    H1 = [chan(idx,:,1); chan(idx,:,2);
        chan(idx+1,2,1) -chan(idx+1,1,1) chan(idx+1,4,1) -chan(idx+1,3,1);
        chan(idx+1,2,2) -chan(idx+1,1,2) chan(idx+1,4,2) -chan(idx+1,3,2);
        chan(idx+2,3,1) -chan(idx+2,4,1) -chan(idx+2,1,1) chan(idx+2,2,1);
        chan(idx+2,3,2) -chan(idx+2,4,2) -chan(idx+2,1,2) chan(idx+2,2,2);
        chan(idx+3,4,1) chan(idx+3,3,1) -chan(idx+3,2,1) -chan(idx+3,1,1);
        chan(idx+3,4,2) chan(idx+3,3,2) -chan(idx+3,2,2) -chan(idx+3,1,2)];
    H2 = [chan(idx+4,:,1); chan(idx+4,:,2);
        chan(idx+5,2,1) -chan(idx+5,1,1) chan(idx+5,4,1) -chan(idx+5,3,1);
        chan(idx+5,2,2) -chan(idx+5,1,2) chan(idx+5,4,2) -chan(idx+5,3,2);
        chan(idx+6,3,1) -chan(idx+6,4,1) -chan(idx+6,1,1) chan(idx+6,2,1);
        chan(idx+6,3,2) -chan(idx+6,4,2) -chan(idx+6,1,2) chan(idx+6,2,2);
        chan(idx+7,4,1) chan(idx+7,3,1) -chan(idx+7,2,1) -chan(idx+7,1,1);
        chan(idx+7,4,2) chan(idx+7,3,2) -chan(idx+7,2,2) -chan(idx+7,1,2)];
    y(idx:idx+3) = [H1; conj(H2)]\reshape(x(idx:idx+7,:).',16,1);
end

% remove extra entries
y(inds) = [];
```