



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

2010-12

# Implementation and performance evaluation of WiMAX STC for OFDMA

Chye, Chia Boon

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/55198>

---

Copyright is reserved by the copyright owner.

*Downloaded from NPS Archive: Calhoun*



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

<http://www.nps.edu/library>



**NAVAL  
POSTGRADUATE  
SCHOOL**

**MONTEREY, CALIFORNIA**

**THESIS**

**IMPLEMENTATION AND PERFORMANCE EVALUATION  
OF WIMAX STC FOR OFDMA**

by

Chia Boon Chye

December 2010

Thesis Co-Advisors:

Tri Ha  
Weilian Su

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
<b>1. AGENCY USE ONLY (Leave blank)</b>	<b>2. REPORT DATE</b> December 2010	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE</b> Implementation and Performance Evaluation of WIMAX STC for OFDMA		<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> Chia Boon Chye		<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000		<b>10. SPONSORING/MONITORING AGENCY REPORT NUMBER</b>	
<b>9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A		<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number: N/A	
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited		<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT</b>  The major driver for broadband wireless communications has been reliable, high-data rate services. In wireless communication, the multipath fading constitutes a bottleneck for increasing data rates and causes performance degradation. To combat fading, we can use diversity. Wireless systems with multiple antennas at the transmitter and receiver have much larger capacity in fading channels than standard wireless systems.  The objective of this thesis is to investigate the transmission scheme provided by matrix $A$ and $B$ in the 802.16 standard and show how it can be implemented.  The research focuses on using maximal-ratio combining (MRC) to demodulate the transmitted symbols. Modifications to the existing matrix by using more frequency bands were introduced; this reduces the number of transmitting antennas and uses fewer time slots to transmit the same number of symbols. The modulator and demodulator design is also discussed. The performance of orthogonal and non-orthogonal space time codes (STC) are evaluated.			
<b>14. SUBJECT TERMS</b>  MRC, STBC, ML, WiMAX, Alamouti Code, Quasi-Orthogonal, Non-Orthogonal		<b>15. NUMBER OF PAGES</b>  107	
<b>17. SECURITY CLASSIFICATION OF REPORT</b>  Unclassified		<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b>  Unclassified	<b>16. PRICE CODE</b>
<b>17. SECURITY CLASSIFICATION OF REPORT</b>  Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b>  Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b>  Unclassified	<b>20. LIMITATION OF ABSTRACT</b>  UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**IMPLEMENTATION AND PERFORMANCE EVALUATION OF WIMAX STC  
FOR OFDMA**

Chia Boon Chye  
Engineer, Singapore Technologies Kinetics  
B.Eng., Nanyang Technological University, 2007

Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL  
December 2010**

Author: Chia Boon Chye

Approved by: Tri Ha  
Thesis Co-Advisor

Weilian Su  
Thesis Co-Advisor

R. Clark Robertson  
Chairman, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

## ABSTRACT

The major driver for broadband wireless communications has been reliable, high-data rate services. In wireless communication, the multipath fading constitutes a bottleneck for increasing data rates and causes performance degradation. To combat fading, we can use diversity. Wireless systems with multiple antennas at the transmitter and receiver have much larger capacity in fading channels than standard wireless systems.

The objective of this thesis is to investigate the transmission scheme provided by matrix  $A$  and  $B$  in the 802.16 standard and show how it can be implemented.

The research focuses on using maximal-ratio combining (MRC) to demodulate the transmitted symbols. Modifications to the existing matrix by using more frequency bands were introduced; this reduces the number of transmitting antennas and uses fewer time slots to transmit the same number of symbols. The modulator and demodulator design is also discussed. The performance of orthogonal and non-orthogonal space time codes (STC) are evaluated.



THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
	<b>A. OVERVIEW .....</b>	<b>1</b>
	<b>B. THESIS OBJECTIVE .....</b>	<b>1</b>
	<b>C. THESIS SYNOPSIS .....</b>	<b>2</b>
<b>II.</b>	<b>BACKGROUND .....</b>	<b>3</b>
	<b>A. DIVERSITY .....</b>	<b>3</b>
	1. Time Diversity .....	3
	2. Frequency Diversity .....	4
	3. Space Diversity .....	4
	<b>B. DIVERSITY COMBINING METHODS .....</b>	<b>5</b>
	1. Selection Combining .....	5
	2. Switched Combining .....	6
	3. Maximal-Ratio Combining (MRC) .....	7
	4. Equal Gain Combining .....	8
	<b>C. SPACE-TIME BLOCK CODING (STBC) .....</b>	<b>8</b>
	1. Use of Space Time Coding (STC) .....	9
	<b>D. ORTHOGONALITY OF THE CODE MATRIX.....</b>	<b>9</b>
	1. Quasi-Orthogonal STBCs .....	10
<b>III.</b>	<b>IMPLEMENTATION AND PERFORMANCE EVALUATION OF WIMAX</b>	
	<b>ORTHOGONAL STC FOR OFDMA.....</b>	<b>11</b>
	<b>A. MATRIX A IN 802.16.....</b>	<b>11</b>
	1. Matrix A .....	11
	2. Matrix $A_2$ (Variation of Matrix A).....	23
	3. Matrix $A_3$ (Variation of Matrix A).....	25
	4. Matrix $D$ (Variation of Matrix A Using Two Frequency Bands).....	26
	<b>B. PERFORMANCE EVALUATION.....</b>	<b>32</b>
	<b>C. CHAPTER SUMMARY.....</b>	<b>35</b>
<b>IV.</b>	<b>IMPLEMENTATION AND PERFORMANCE EVALUATION OF WIMAX</b>	
	<b>NON-ORTHOGONAL STC FOR OFDMA .....</b>	<b>37</b>
	<b>A. MATRIX B IN 802.16.....</b>	<b>37</b>
	1. Matrix $B$ .....	37
	2. Matrix $B_2$ (Variation of Matrix $B$ ).....	44
	3. Matrix $E$ (Variation of Matrix $B$ Using Four Frequency Bands).....	46
	<b>B. PERFORMANCE EVALUATION.....</b>	<b>54</b>
	<b>C. CHAPTER SUMMARY.....</b>	<b>64</b>
<b>V.</b>	<b>CONCLUSION AND FUTURE WORKS.....</b>	<b>65</b>
	<b>A. CONCLUSION .....</b>	<b>65</b>
	<b>B. FUTURE WORK.....</b>	<b>65</b>
	<b>APPENDIX A .....</b>	<b>67</b>
	<b>APPENDIX B .....</b>	<b>69</b>

<b>APPENDIX C</b> .....	<b>79</b>
<b>APPENDIX D</b> .....	<b>81</b>
<b>APPENDIX E</b> .....	<b>85</b>
<b>LIST OF REFERENCES</b> .....	<b>87</b>
<b>INITIAL DISTRIBUTION LIST</b> .....	<b>89</b>

## LIST OF FIGURES

Figure 1.	Selection combining method.....	6
Figure 2.	Switched combining method.....	7
Figure 3.	Maximal-ratio combining method. ....	8
Figure 4.	Transmission process for matrix $A$ . ....	12
Figure 5.	Modulator circuit from the base station for matrix $A$ . ....	20
Figure 6.	Demodulator circuit at the subscriber station for matrix $A$ . ....	20
Figure 7.	Modulator circuit from the base station for the Alamouti code.....	23
Figure 8.	Demodulator circuit at the subscriber station for the Alamouti code. ....	23
Figure 9.	Transmission process for matrix $D$ . ....	28
Figure 10.	Modulator for matrix $D$ . ....	32
Figure 11.	Demodulator for matrix $D$ . ....	32
Figure 12.	Graph of bit error rate vs $E_b/N_0$ (dB) for QPSK and 16QAM. ....	34
Figure 13.	Transmission process for matrix $B$ . ....	38
Figure 14.	Modulator circuit from the base station for matrix $B$ . ....	43
Figure 15.	Demodulator circuit at the subscriber station for matrix $B$ with two receiving antennas.....	43
Figure 16.	Transmission process for matrix $E$ . ....	47
Figure 17.	Modulator for matrix $E$ . ....	53
Figure 18.	Demodulator for matrix $E$ . ....	54
Figure 19.	Simulation results for $H_{1-1}, H_{1-2}, H_{1-3}, H_{1-4}$ and QPSK closed form solution...59	59
Figure 20.	Simulation results for $H_{1-5}, H_{1-6}, H_{1-7}, H_{1-8}$ and QPSK closed form solution...60	60
Figure 21.	Simulation results for $H_{1-9}, H_{1-10}, H_{1-11}, H_{1-12}$ and QPSK closed form solution.....	61
Figure 22.	Simulation results for $H_{1-1}, H_{1-1}$ with only one receiving antenna and QPSK closed form solution. ....	62
Figure 23.	Simulation results for $H_{1-1}, H_{1-2}, H_{1-3}, H_{1-4}$ and 16QAM closed form solution.....	63

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF ACRONYMS AND ABBREVIATIONS

BER	bit error rate
BS	base station
IID	independent identically distributed
ISI	intersymbol interference
MIMO	multiple input multiple output
MISO	multiple output single input
ML	maximum likelihood
MMSE	minimum mean-square error
MQAM	M-ary quadrature amplitude modulation
MRC	maximal-ratio combining
OFDM	orthogonal frequency division multiplexing
OFDMA	orthogonal frequency division multiple access
QAM	quadrature amplitude modulation
QPSK	quadrature phase shift keying
SISO	single input single output
SMX	spatial multiplexing
SNR	signal to noise ratio
SS	subscriber station
ST	space-time
STBC	space-time block code
STC	space-time code
WiMAX	worldwide interoperability for microwave access

THIS PAGE INTENTIONALLY LEFT BLANK

## EXECUTIVE SUMMARY

Wireless systems with multiple antennas at the transmitter and receiver have a much larger capacity to combat fading channels than standard wireless systems.

These systems use diversity techniques to improve the reliability of transmission and reduce the effect of multipath fading without increasing the transmitted power or increasing the bandwidth. Unlike time and frequency diversity, space diversity does not require more bandwidth. This property makes space diversity very popular for high data rate wireless communications. This research examines the maximal-ratio combining (MRC) method. This method maximizes the signal-to-noise ratio (SNR) since the maximum output SNR is equal to the sum of the instantaneous SNRs of the individual signals.

Two matrices, matrix  $A$  and  $B$  in the 802.16 standard, are evaluated in this research. Matrix  $A$  refers to the Space time block coding (STBC) technique and multiple input multiple-output (MIMO), whereas matrix  $B$  refers to the spatial multiplexing (SMX) technique. In matrix  $A$ , four symbols are transmitted using four antennas in four time slots. It has a code rate of one. On the other hand, matrix  $B$  transmits eight symbols using four antennas in four time slots. Matrix  $B$  has a code rate of two. The designs of the modulator and demodulator for both matrices are discussed, and the performances are evaluated.

Matrix  $A$  is quasi-orthogonal, and matrix  $B$  is non-orthogonal. The closed form solution to evaluate the performance of matrix  $A$  was used in this study, and MATLAB was used to evaluate the performance of matrix  $B$ . Maximum likelihood (ML) decoding was employed to perform the simulations. The simulation results show that there is a 4 dB drop in performance for quadrature phase-shift keying (QPSK) and 16-quadrature amplitude modulation (16QAM) when they are compared to the closed form solution. In addition, the comparison of using a single receiving antenna and two receiving antennas was performed, and the results show that there is a 11 dB drop in performance when one receiving antenna is used.



This research also examined the twelve variations of matrix  $B$  and evaluated their individual performances. Since matrix  $B$  sends out four symbols for the first two time slots and the next four symbols in the next two time slots, we can separate the matrix into two matrices. Thus, the simulation in MATLAB is a program based on the first two time slots and this saves time in the simulation process. The performances for four out of the twelve matrices were found to be worse than the other eight matrices. Therefore, we must avoid using these four matrices in the transmission process.

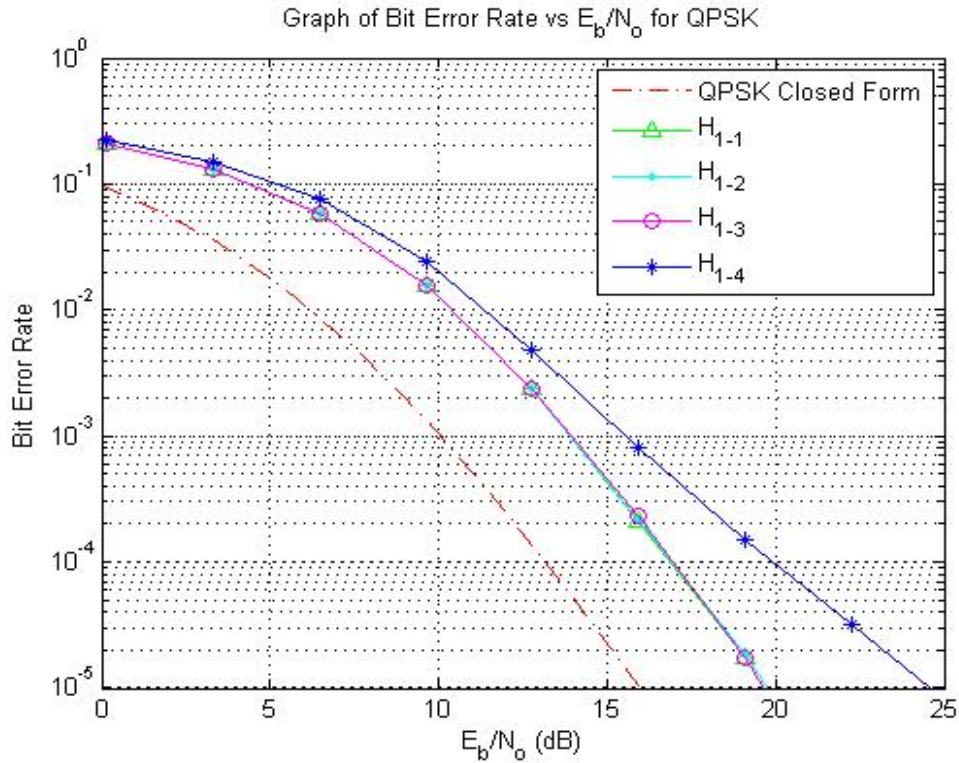


Figure 1. Simulation results for  $H_{1-1}$ ,  $H_{1-2}$ ,  $H_{1-3}$ ,  $H_{1-4}$  and QPSK closed form solution.

From Figure 1, the results for  $H_{1-4}$ , which is the fourth variation of matrix  $B$ , has a poorer performance than the rest of the three matrices by 5 dB.

Finally, we can increase the code rate and have better performance in terms of bit error rate (BER) by introducing more frequency bands to the STBC, matrix  $B$ . This can be proven mathematically using MRC. For example, by introducing four frequency bands to matrix  $B$ , eight symbols are sent using two time slots, and the code rate is increased from two to four.

## **ACKNOWLEDGMENTS**

I would like to dedicate this work to my wife, Shwu Chin Lee, for her continuous loving support and the happiness that she brings to my life.

I would like to thank Professors Tri Ha and Weilian Su for their patience and guidance throughout the thesis process.

Also, I would like to thank, MAJ Wei Gee Ng for providing a large part of the MATLAB code used to obtain the simulation results. My coursemate, Chin Leong Lu, Henry Seet and MAJ Gabriel Tham for helping me along. Mrs. Donna L. Miller for her support. I enjoy my time in the CRL Lab.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

## A. OVERVIEW

The major driver for broadband wireless communications has been reliable, high-data-rate services. The scarcity of bandwidth resources motivates researchers to develop highly efficient modulation schemes and more efficient coding that utilize less bandwidth and, at the same, time provide better quality.

In wireless communication, multipath fading constitutes a bottleneck for increasing data rates and causes performance degradation. Combating this fading requires the use of diversity.

Spatial multiplexing (SMX) using space-time coding (STC) is effective for combating fading and enhancing data rates. It relies on simultaneous coding across space and time to gain diversity without using too much bandwidth. Transmit diversity has been studied extensively as a method to combat impairments in wireless fading channels [1]. The main advantage of using transmit diversity is that it is relative simple to implement and is feasible to install multiple antennas at the base station (BS).

One attractive approach of space–time code design is to construct space-time block codes (STBCs) from orthogonal designs as proposed by Alamouti [2]. These codes achieve full diversity and have fast maximum-likelihood (ML) decoding at the receiver

The use of maximal-ratio combining (MRC) as a combining method is popular as it maximizes the signal-to-noise ratio (SNR) since the maximum output SNR is equal to the sum of the instantaneous SNRs of the individual signals [3].

## B. THESIS OBJECTIVE

The target of this research is to implement and evaluate the performance of WiMAX (Worldwide Interoperability for Microwave Access) orthogonal space time code (STC) for orthogonal frequency-division multiple access (OFDMA). The key objective is to study the transmission scheme, matrix  $A$  and  $B$  in the 802.16 standard, and try to improve the performances by modifying these matrices. The performances of the STCs

were evaluated, and their modulator and demodulator designs were discussed. Simulations were developed in MATLAB and used for comparison for the respective modification of the matrices.

### **C. THESIS SYNOPSIS**

The structure of this thesis comprises an Introduction (Chapter I), Background (Chapter II), Implementation and Performance Evaluation of WiMAX orthogonal STC for OFDMA, an Implementation and Performance Evaluation of WiMAX non-orthogonal STC for OFDMA and a Conclusion. The evaluation of matrix  $A$  from the 802.16 standard and its variations are provided in Chapter III. In Chapter IV, matrix  $B$  was evaluated and the performance of its variations are presented. Simulations were developed in MATLAB. The performance of using one and two receive antennas for decoding was compared in this chapter as well. Finally, the conclusion of this thesis report can be found in Chapter V.

## II. BACKGROUND

The basic principles and background knowledge of diversity [3], the diversity combining method [3], SMX and space-time block coding (STBC) are presented in this chapter.

### A. DIVERSITY

Wireless mobile communications use diversity techniques to improve the reliability of transmission and reduce the effect of multipath fading without increasing the transmitted power or increasing the bandwidth. For example, a time diversity technique requires the transmission of same symbols numerous times. Each transmission carries the same information but has a small correlation in fading statistics.

The principle behind diversity is that when two or more independent samples of a signal are received, the samples may fade in an uncorrelated manner. Some samples are more attenuated, while others are less faded. Therefore, the probability of all samples falling below a given level is much lower than the probability of any individual sample falling below that level. As such, with proper combinations of the various samples, one can greatly reduce the severity of fading and improve the communication link.

Wireless communication systems use and combine a number of diversity methods to achieve the desired performance. Diversity techniques can be classified into time, frequency and space diversity.

#### 1. Time Diversity

One can employ time diversity by transmitting the same symbol in multiple time slots at different times so the receiver will receive uncorrelated fading signals. The required time separation is at least the coherence time of the channel, or the reciprocal of the fading rate  $1/f_d = c/vf_c$ , where  $f_d$  is the Doppler shift of the received signal,  $f_c$  is the frequency of the carrier,  $v$  is the vehicle speed, and  $c$  is the speed of light.

The coherence time is a statistical measure of the period of the time over which the channel fading process is correlated. One of the drawbacks of this scheme is that the redundancy introduced in the time domain leads to a loss in bandwidth efficiency.

## **2. Frequency Diversity**

Using a number of different frequencies to transmit a message is called frequency diversity. For this to work, it is necessary to separate the frequencies enough to ensure that each carrier frequency experiences independent fading. To achieve that, the frequency separation must be in the order of several times the channel coherence bandwidth as this will guarantee that the fading statistics for the various frequencies are uncorrelated. The coherence bandwidth is different for different environments. Similar to time diversity, frequency diversity will occupy a larger bandwidth as redundancy is introduced in the frequency domain.

## **3. Space Diversity**

Space diversity, sometimes also known as antenna diversity, uses multiple antennas or antenna arrays arranged together in space for transmission and reception. The antennas are separated physically by proper distances, so that individual signals are uncorrelated. The separation requirements vary with the environment, height of antenna and frequency. In space diversity, the repeated copies of the transmitted signals are usually given to the receiver in the space domain. Unlike time and frequency diversity, space diversity does not require more bandwidth. This property makes space diversity very popular for high data rate wireless communications.

The use of space diversity to implement the STC will be explored in this thesis; therefore, a more in-depth discussion of space diversity is necessary.

One can classify space diversity into two categories: transmit and receive diversity. For transmit diversity, multiple antennas are installed at the transmitter station. The transmitter processes and combines the messages and delivers them across multiple antennas for transmission to one or more receiving antennas. For receive diversity, multiple antennas are installed at the receiver station to pick up copies of the transmit

signals. The replicas of the transmitted signals are properly combined to minimize fading due to multipath with the hope to improve the signal-to-noise ratio (SNR).

Space diversity using different transmit antennas alone cannot provide separate copies of the symbol at the receiver since the receive antenna receives the sum of these channel-tap weighted copies but does not have the capability to separate them for combining. In general, two or more diversity schemes are usually necessary to provide multidimensional diversity to meet system performance specifications. A combination of two diversity techniques should provide the receiver sufficient statistics to separate copies of a transmitted symbol for combining. This is dictated by maximal-ratio combining (MRC). As for the receiving end, one can assume that the transmit antennas have sufficient inter-element spacing to ensure that all paths from the transmitter to the receiver are uncorrelated.

## **B. DIVERSITY COMBINING METHODS**

The key feature of all diversity techniques is the low probability of deep fades occurring simultaneously in the various diversity subchannels. The performance of the communication link depends largely on how the repeated signals are combined, which results in an increase to the SNR of the communication system. The type of combining methods implemented at the receiver can, therefore, classify diversity schemes. According to the implementation complexity and the level of channel state information required by the combining method at the receiver, we find that there are four main types of combining methods. They are selection combining, switched combining, maximal-ratio combining (MRC), and equal gain combining.

### **1. Selection Combining**

Consider a receive diversity system with  $n_R$  receive antennas. The block diagram [3] of the selection combining scheme is illustrated in Figure 1. In this system, the signal with the largest instantaneous SNR at every symbol interval is selected as the output; the output SNR is equal to that of the best incoming signal. In practice, the signal with the



highest sum of the signal and noise power is usually the one selected, as it is difficult to measure the signal and noise power. This method is not recommended when the noise power is high.

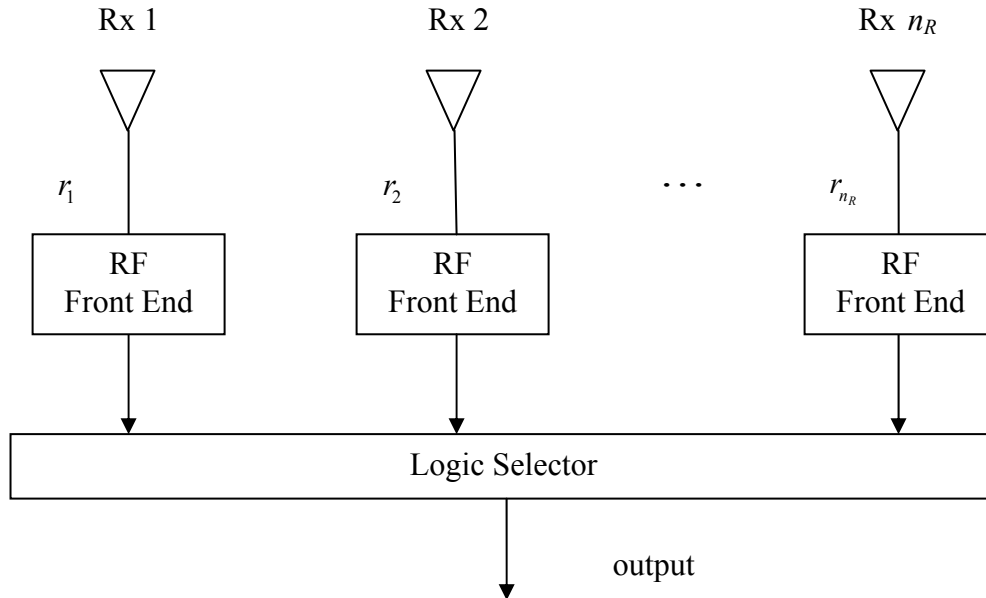


Figure 1. Selection combining method.

## 2. Switched Combining

In switched combining, the receiver scans all the diversity branches and selects the particular branch with the SNR above a specified threshold. The receiver then selects this signal as the output. When the SNR drops below the threshold, the receiver starts scanning again and switches to another branch with the SNR above the threshold. This scheme is also called scanning diversity [3] (shown in Figure 2).

Switched diversity is inferior to selection combining since it does not continually pick up the best instantaneous signal. However, it is simpler to design because it does not require continuous and simultaneous monitoring of all the diversity branches.

Both switched and selection diversity schemes obtain their output signals from only one of the diversity branches and do not require any prior knowledge of channel state information.

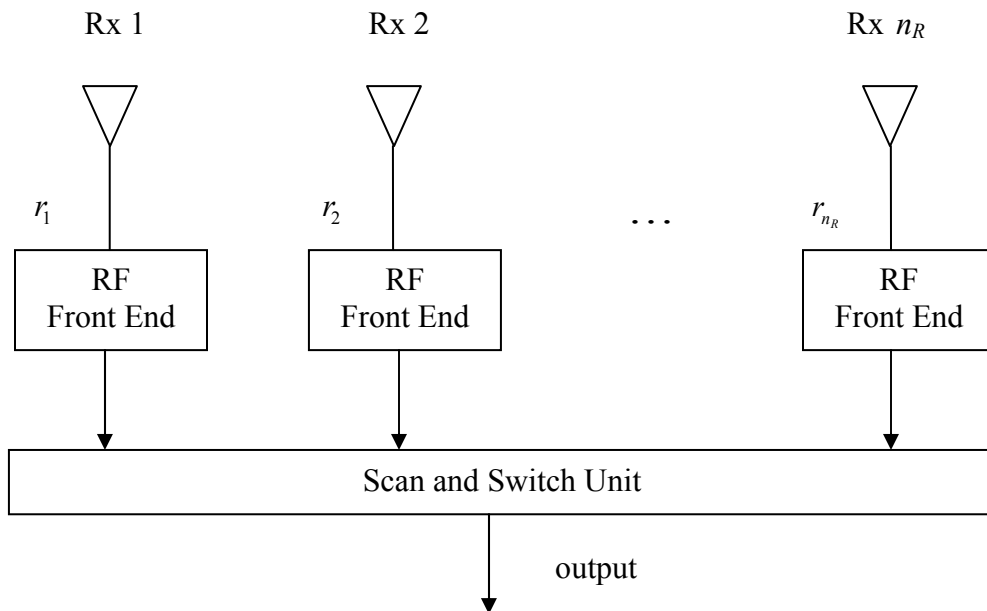


Figure 2. Switched combining method.

### 3. Maximal-Ratio Combining (MRC)

Maximal-ratio combining is a linear combining method. The received signals are individually weighted and added together coherently to obtain a decision statistic. There are several ways to choose the weighting factors. The decision statistic is a linear combination of a weighted replica of all the received signals. The block diagram [3] for maximal-ratio combining diversity is illustrated in Figure 3. In MRC, the weighting factor of each receive antenna is chosen to be in proportion to its own SNR. In this scheme, each individual signal must be co-phased, weighted with its corresponding amplitude, and then summed. This scheme requires knowledge of signal phases and channel fading amplitude. This method is also known as optimum combining, since it maximizes the SNR. The maximum output SNR is equal to the sum of the instantaneous SNRs of the individual signals.

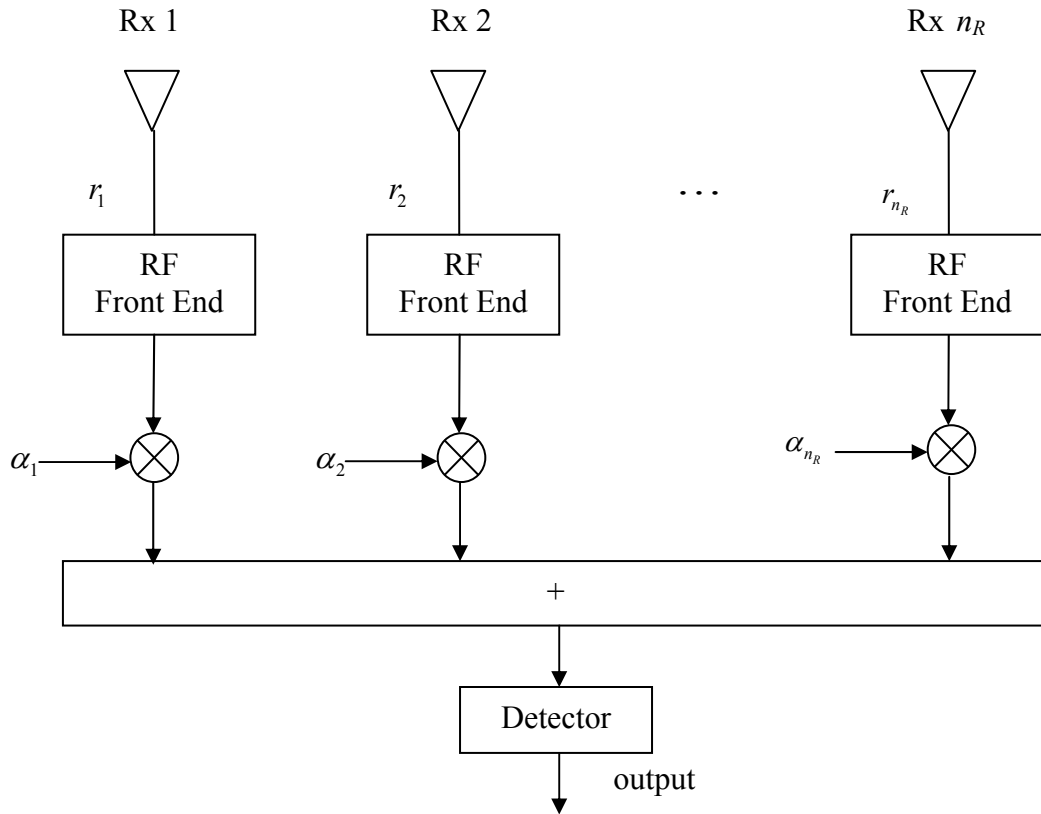


Figure 3. Maximal-ratio combining method.

#### 4. Equal Gain Combining

Equal gain combining is another linear combining method but is suboptimal. It does not require the estimation of the fading amplitude for each branch. The receiver simply sets the amplitudes of the weighting factors to be unity.

#### C. SPACE-TIME BLOCK CODING (STBC)

Space-time block coding is a technique used in wireless communications to transmit multiple copies of a symbol across a number of antennas and combine various received versions of the symbol to improve the reliability of the communications link. Scattering, reflection, refraction and thermal noise normally affect the transmitted signal.

Some of the received copies of the symbol will be 'better' than others. This redundancy allows for the use of multiple copies to compensate for fading and helps to correctly decode the received signal. One can use STC to combine all copies of the same symbol in an optimal way to get as much information as possible from each of them.

### 1. Use of Space Time Coding (STC)

Using an example from [4], we have a space-time block code that allows the transmission of  $m$  symbol copies in each word of  $L_s$  symbols; hence, the code rate is  $m/L_s$ . The diversity order of the space-time code is  $M_T$ , the number of transmit antennas. To separate the symbol copies for combining, it is necessary that the columns of the code matrix  $G$  be (complex) orthogonal since the transmitter sends all  $L_t$  code words concurrently over  $L_s$  symbol times. One can assume that the channel remains constant for  $m$  symbol times, and there is perfect channel estimation. Given the transmitted space-time block code matrix  $G$ , the sufficient statistics for maximal-ratio combining is dictated to obtain a matrix  $H$  from the vector:

$$\begin{aligned} Gh &\rightarrow Hs \\ H^*H &= \|h\|^2 I_m \end{aligned} \tag{1}$$

where  $h = [h_1 \ h_2 \ \dots \ h_{L_t}]^T$  is the channel tap vector representing  $L_t$  paths,  $s = [s_1 \ s_2 \ \dots \ s_m]^T$  is the symbol vector representing  $m$  transmitted symbols, and  $H$  is the channel tap matrix. The columns of  $H$  must be orthogonal to be able to detect  $s$  and to decompose  $s$  into  $m$  separate complex variables. This implies that the columns of  $G$  must also be orthogonal such that  $G^*G = \|s\|^2 I_{L_s}$ . The matrix  $H$  is known by the receiver via channel estimation and the known structure of  $G$ .

### D. ORTHOGONALITY OF THE CODE MATRIX

Some STBCs used for WiMAX are orthogonal. This means that the STBC is designed such that the vectors representing any pair of columns taken from the coding matrix are orthogonal. This enables the design of a simple and optimal decoder at the

receiver. Most importantly, all but one of the codes that satisfy this criterion must sacrifice some proportion of their data rate (the Alamouti code [2] is the only STBC of unity rate for complex symbols).

In addition, there exist quasi-orthogonal STBCs that achieve higher data rates at the cost of intersymbol interference (ISI). Thus, their bit error rate (BER) performance is lower bounded by the Alamouti unity code that provides ISI free transmissions due to orthogonality.

### 1. Quasi-Orthogonal STBCs

The following code is an example of a quasi-orthogonal code:

$$c_{4,1} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \\ -c_2^* & c_1^* & -c_4^* & c_3^* \\ -c_3^* & -c_4^* & c_1^* & c_2^* \\ c_4 & -c_3 & -c_2 & c_1 \end{bmatrix}. \quad (2)$$

The code exhibits partial orthogonality, as illustrated in (2). The orthogonality criterion only holds for columns (1 and 2), (1 and 3), (2 and 4), and (3 and 4). For example, the first two rows of column one and column two will give us a two by two matrix. Let us define the two by two matrix as matrix  $p$ . Matrix  $p$  must have the property of  $(p)^* p = p(p)^* = I_2$ , where  $p^*$  represents the conjugate and transpose of vector  $p$ . The same property must apply for row three and row four for them to be considered as quasi-orthogonal. All the column pairs mentioned have this property. The code is full-rate and requires linear processing at the receiver.



time slots. The notation  $s^*$  represents the conjugate and transpose of vector  $s$ . The transmission process at different time slots is illustrated in Figure 4.

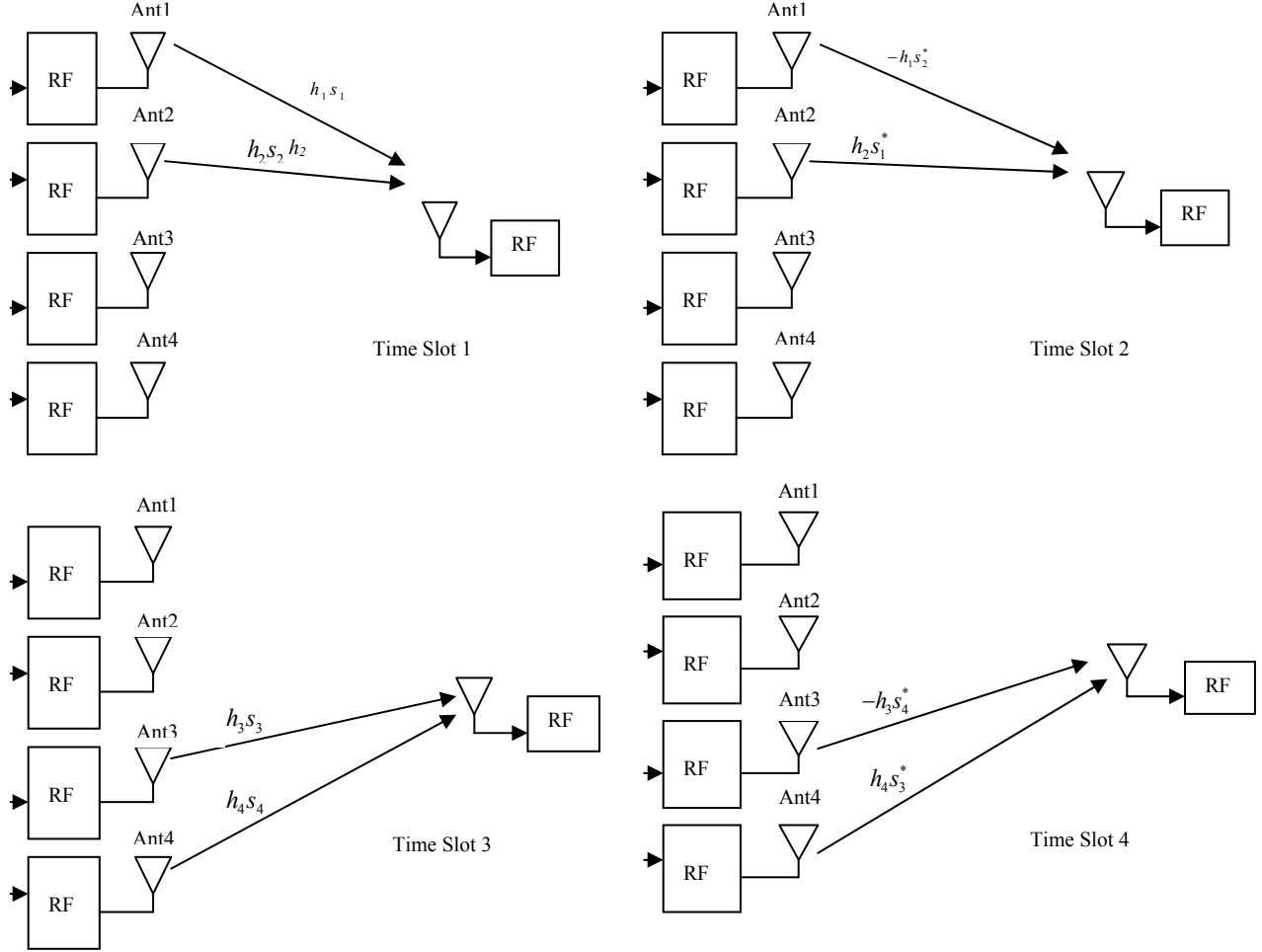


Figure 4. Transmission process for matrix  $A$ .

We use MRC to decode matrix  $A$ . The first step is to prove if matrix  $A^T$  is orthogonal. Matrix  $A^T$  is orthogonal only if  $(A^T)^* A^T = A' A^T = I_4$  where  $A^T$  represents the transpose of matrix  $A$ ,  $A'$  represents the conjugate of  $A$ , and  $(A^T)^*$  represents the conjugate transpose of  $A^T$ , which is the same as  $A'$ :

$$(A^T)^* = A' = \begin{bmatrix} s_1^* & -s_2 & 0 & 0 \\ s_2^* & s_1 & 0 & 0 \\ 0 & 0 & s_3^* & -s_4 \\ 0 & 0 & s_4^* & s_3 \end{bmatrix}. \quad (4)$$

Therefore,

$$(A^T)^* A^T = A' A^T = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & 0 & 0 \\ 0 & s_1 s_1^* + s_2 s_2^* & 0 & 0 \\ 0 & 0 & s_3 s_3^* + s_4 s_4^* & 0 \\ 0 & 0 & 0 & s_3 s_3^* + s_4 s_4^* \end{bmatrix} \quad (5)$$

and

$$A^T (A^T)^* = A^T A' = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & 0 & 0 \\ 0 & s_1 s_1^* + s_2 s_2^* & 0 & 0 \\ 0 & 0 & s_3 s_3^* + s_4 s_4^* & 0 \\ 0 & 0 & 0 & s_3 s_3^* + s_4 s_4^* \end{bmatrix} \quad (6)$$

$$A' A^T = A^T A' = \begin{bmatrix} |s_1|^2 + |s_2|^2 & 0 & 0 & 0 \\ 0 & |s_1|^2 + |s_2|^2 & 0 & 0 \\ 0 & 0 & |s_3|^2 + |s_4|^2 & 0 \\ 0 & 0 & 0 & |s_3|^2 + |s_4|^2 \end{bmatrix} \quad (7)$$

Let  $\|s_a\|^2 = |s_1|^2 + |s_2|^2$  and  $\|s_b\|^2 = |s_3|^2 + |s_4|^2$  and  $Q = A' A^T = A^T A'$ .



Therefore,

$$Q = A'A^T = A^T A' = \begin{bmatrix} \|s_a\|^2 & 0 & 0 & 0 \\ 0 & \|s_a\|^2 & 0 & 0 \\ 0 & 0 & \|s_b\|^2 & 0 \\ 0 & 0 & 0 & \|s_b\|^2 \end{bmatrix} \quad (8)$$

$$Q^{-1} = \begin{bmatrix} \frac{1}{\|s_a\|^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\|s_a\|^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\|s_b\|^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\|s_b\|^2} \end{bmatrix} \quad (9)$$

From Equations (5) and (6), we can prove that matrix  $A$  is not orthogonal, but it is quasi-orthogonal.

Next, we can derive the MRC equation to demodulate  $s = [s_1 \ s_2 \ s_3 \ s_4]^T$  directly from the four antennas. This requires the performance of MRC on matrix  $A$ , with the receive matrix  $[Z] = [G][h] + [n]$ , where  $n = [n_1, n_2, \dots, n_{L_s}]^T$  is the Gaussian noise vector. The  $A^T$  matrix is the  $G$  matrix for this case. Therefore,

$$\begin{aligned}
Z &= \begin{bmatrix} s_1 & s_2 & 0 & 0 \\ -s_2^* & s_1^* & 0 & 0 \\ 0 & 0 & s_3 & s_4 \\ 0 & 0 & -s_4^* & s_3^* \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \\
&= \begin{bmatrix} s_1 h_1 + s_2 h_2 \\ -s_2^* h_1 + s_1^* h_2 \\ s_3 h_3 + s_4 h_4 \\ -s_4^* h_3 + s_3^* h_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}
\end{aligned} \tag{10}$$

Next, we make use of the channel estimation to estimate the channel tap vector  $h$ . Consider an example from [4]. To estimate the channel tap vector  $h$ , consider the received vector  $Z = Gh + n$  where  $G = A^T$ . Since  $G^*G = Q$ , the channel estimation can be carried out with a known pilot symbol vector  $s$ , and  $G$  is known to the receiver. The sufficient statistics provide the minimum mean-square error (MMSE) of the vector  $h$ :

$$\begin{aligned}
\hat{h} &= Q^{-1}Qh + Q^{-1}G^*n \\
&= h + \hat{n}.
\end{aligned} \tag{11}$$

The jointly complex Gaussian noise vector  $(G^*Q^{-1})n$  has the diagonal covariance matrix

$$\begin{aligned}
E(Q^{-1}A^{T*}nm^*A^TQ^{-1}) &= Q^{-1}A'E(nm^*)A^TQ^{-1} \\
&= 2\sigma^2Q^{-1}A'A^TQ^{-1} \\
&= 2\sigma^2Q^{-1}QQ^{-1} \\
&= 2\sigma^2Q^{-1} \\
&= 2\sigma^2 \begin{bmatrix} \frac{1}{\|s_a\|^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\|s_a\|^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\|s_b\|^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\|s_b\|^2} \end{bmatrix}.
\end{aligned} \tag{12}$$

This implies that the components of this noise vector are white and have identical variance  $2\sigma^2Q^{-1}$ . The components of  $h$  can be estimated in the mean square error sense. At high SNR,  $\hat{h}$  can serve as the estimate of  $h$ . In practice,  $h$  may be made available from pilot symbols extracted from the pilot tone transmitting simultaneously with the signal.

Since

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} = \begin{bmatrix} s_1 h_1 + s_2 h_2 \\ -s_2^* h_1 + s_1^* h_2 \\ s_3 h_3 + s_4 h_4 \\ -s_4^* h_3 + s_3^* h_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}, \quad (13)$$

the vector  $Z=Gh+n$  can be mapped into  $Y=Hs+N$  by changing the second element of  $Z = [Z_1 \ Z_2 \ Z_3 \ Z_4]^T$ , which is  $Z_2 = h_2 s_1^* - h_1 s_2^* + n_2$ , into its complex conjugate  $Z_2^* = h_2^* s_1 - h_1^* s_2 + n_2^*$ , and the fourth element, which is from  $Z_4 = h_4 s_3^* - h_3 s_4^* + n_4$ , into its complex conjugate  $Z_4^* = h_4^* s_3 - h_3^* s_4 + n_4^*$ . Thus,

$$\begin{aligned} Y = Hs + n &= \begin{bmatrix} Z_1 \\ Z_2^* \\ Z_3 \\ Z_4^* \end{bmatrix} = \begin{bmatrix} s_1 h_1 + s_2 h_2 \\ -s_2^* h_1 + s_1^* h_2 \\ s_3 h_3 + s_4 h_4 \\ -s_4^* h_3 + s_3^* h_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix} \\ &= \begin{bmatrix} h_1 & h_2 & 0 & 0 \\ h_2^* & -h_1^* & 0 & 0 \\ 0 & 0 & h_3 & h_4 \\ 0 & 0 & h_4^* & -h_3^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix} = [H][s] + [n]. \end{aligned} \quad (14)$$

From Equation (14), we can generate four separate equations to solve for the four unknown symbols,  $s_1, s_2, s_3$  and  $s_4$ . Since the noise  $n_1, n_2, n_3$  and  $n_4$  are independent and  $h_1, h_2, h_3$  and  $h_4$  are known from the channel estimation, we can solve for  $s_1, s_2, s_3$ , and  $s_4$ .

The next step is to use the estimated channel taps information  $[\hat{h}_1, \hat{h}_2, \hat{h}_3, \hat{h}_4]$  via the channel estimation process to estimate  $h_1, h_2, h_3$  and  $h_4$ . The receiver can then use  $[\hat{H}]$  to perform MRC.

For simplicity, assume perfect channel estimation where  $[\hat{h}_1, \hat{h}_2, \hat{h}_3, \hat{h}_4] \rightarrow [h_1, h_2, h_3, h_4]$  and apply MRC on  $Y$  to obtain the corresponding decision statistics vector  $[X]$ , where

$$\begin{aligned}
[X] = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} &= \begin{bmatrix} \frac{h_1}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1^*}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3^*}{\|h_b\|} \end{bmatrix}^* [Y] \\
&= \begin{bmatrix} \frac{h_1^*}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3^*}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3}{\|h_b\|} \end{bmatrix} \{[H][s] + [N]\} \\
&= \begin{bmatrix} \frac{h_1^*}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3^*}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3}{\|h_b\|} \end{bmatrix} [H][s] + \begin{bmatrix} \frac{h_1^*}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3^*}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3}{\|h_b\|} \end{bmatrix} [N] \tag{15}
\end{aligned}$$

$$\begin{aligned}
&= \begin{bmatrix} \frac{|h_1|^2 + |h_2|^2}{\|h_a\|} & 0 & 0 & 0 \\ 0 & \frac{|h_1|^2 + |h_2|^2}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{|h_3|^2 + |h_4|^2}{\|h_b\|} & 0 \\ 0 & 0 & 0 & \frac{|h_3|^2 + |h_4|^2}{\|h_b\|} \end{bmatrix} [S] + \begin{bmatrix} \frac{h_1^*}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3^*}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3}{\|h_b\|} \end{bmatrix} [N] \\
&= \begin{bmatrix} \sqrt{|h_1|^2 + |h_2|^2} & 0 & 0 & 0 \\ 0 & \sqrt{|h_1|^2 + |h_2|^2} & 0 & 0 \\ 0 & 0 & \sqrt{|h_3|^2 + |h_4|^2} & 0 \\ 0 & 0 & 0 & \sqrt{|h_3|^2 + |h_4|^2} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} \frac{h_1^*}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3^*}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3}{\|h_b\|} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \\
&= \begin{bmatrix} \|h_a\| & 0 & 0 & 0 \\ 0 & \|h_a\| & 0 & 0 \\ 0 & 0 & \|h_b\| & 0 \\ 0 & 0 & 0 & \|h_b\| \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} \frac{h_1^*}{\|h_a\|} & \frac{h_2}{\|h_a\|} & 0 & 0 \\ \frac{h_2^*}{\|h_a\|} & \frac{-h_1}{\|h_a\|} & 0 & 0 \\ 0 & 0 & \frac{h_3^*}{\|h_b\|} & \frac{h_4}{\|h_b\|} \\ 0 & 0 & \frac{h_4^*}{\|h_b\|} & \frac{-h_3}{\|h_b\|} \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \\
&= \begin{bmatrix} \|h_a\| s_1 \\ \|h_a\| s_2 \\ \|h_b\| s_3 \\ \|h_b\| s_4 \end{bmatrix} + \begin{bmatrix} \frac{h_1^*}{\|h_a\|} n_1 + \frac{h_2}{\|h_a\|} n_2 \\ \frac{h_2^*}{\|h_a\|} n_1 - \frac{h_1}{\|h_a\|} n_2 \\ \frac{h_3^*}{\|h_b\|} n_3 + \frac{h_4}{\|h_b\|} n_4 \\ \frac{h_4^*}{\|h_b\|} n_3 - \frac{h_3}{\|h_b\|} n_4 \end{bmatrix}
\end{aligned}$$

where  $\|h_a\|$  and  $\|h_b\|$  are defined as

$$\|h_a\| = \sqrt{|h_1|^2 + |h_2|^2}, \|h_b\| = \sqrt{|h_3|^2 + |h_4|^2}. \quad (16)$$

Next, we would like to determine if the channel is independent. Since

$$\begin{aligned} H^*H &= \begin{bmatrix} h_1h_1^* + h_2h_2^* & 0 & 0 & 0 \\ 0 & h_1h_1^* + h_2h_2^* & 0 & 0 \\ 0 & 0 & h_3h_3^* + h_4h_4^* & 0 \\ 0 & 0 & 0 & h_3h_3^* + h_4h_4^* \end{bmatrix} \\ &= \begin{bmatrix} |h_1|^2 + |h_2|^2 & 0 & 0 & 0 \\ 0 & |h_1|^2 + |h_2|^2 & 0 & 0 \\ 0 & 0 & |h_3|^2 + |h_4|^2 & 0 \\ 0 & 0 & 0 & |h_3|^2 + |h_4|^2 \end{bmatrix}, \end{aligned} \quad (17)$$

letting  $R = H^*H = HH^*$ . Now,

$$R = HH^* = H^*H = \begin{bmatrix} \|h_a\|^2 & 0 & 0 & 0 \\ 0 & \|h_a\|^2 & 0 & 0 \\ 0 & 0 & \|h_b\|^2 & 0 \\ 0 & 0 & 0 & \|h_b\|^2 \end{bmatrix} \quad (18)$$

and

$$R^{-1} = \begin{bmatrix} \frac{1}{\|h_a\|^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\|h_a\|^2} & 0 & 0 \\ 0 & 0 & \frac{1}{\|h_b\|^2} & 0 \\ 0 & 0 & 0 & \frac{1}{\|h_b\|^2} \end{bmatrix}. \quad (19)$$

From (19), we see that the channel is independent.

The block diagrams [5] for the transmitters and receiver are shown in Figure 5 and 6, respectively.

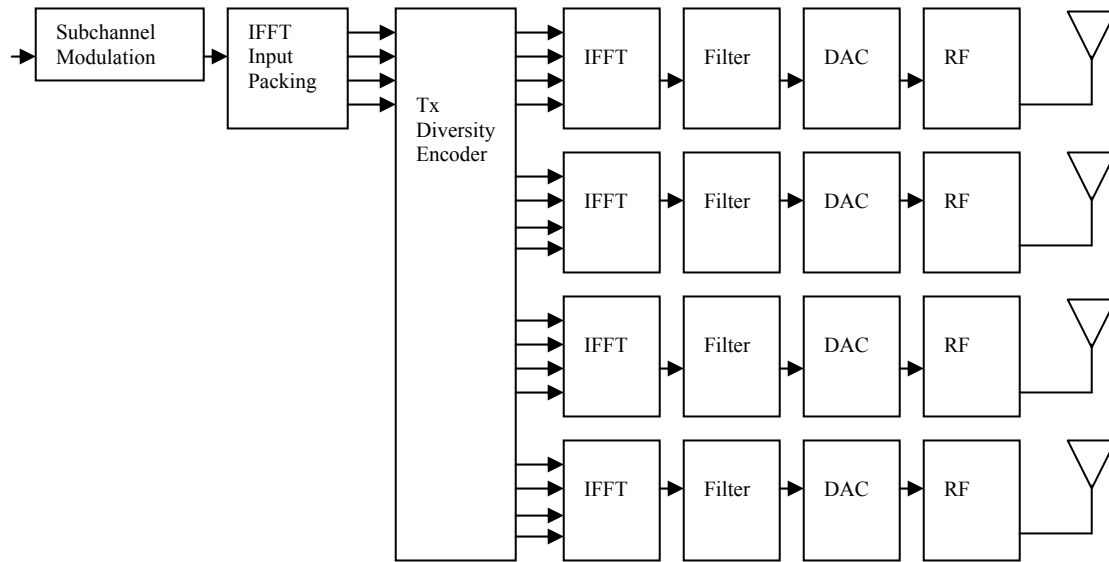


Figure 5. Modulator circuit from the base station for matrix  $A$ .

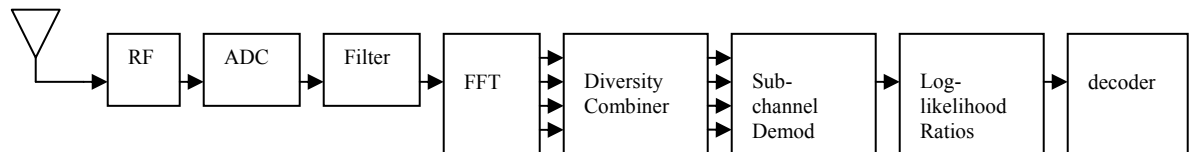


Figure 6. Demodulator circuit at the subscriber station for matrix  $A$ .

There are four transmitting antennas on the BS side and one receiving antenna on the SS side. Two different antennas transmit two different data symbols at the same time. The receiver design will have a higher complexity compared to the transmitter. The transmitter can be less complex if the matrix can be split into two Alamouti [2] codes. This will be discussed in the next design.

Since both antennas transmit at the same time, they share the same local oscillator. Thus, the received signal has exactly the same autocorrelation properties as it would for a single antenna. This scheme requires multiple-input single-output (MISO) channel estimation.

From matrix  $A$ , we can further break down the matrix into two independent matrices  $G_1 = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix}$  and  $G_2 = \begin{bmatrix} s_3 & -s_4^* \\ s_4 & s_3^* \end{bmatrix}$  that follow the Alamouti STBC.

From [2], the Alamouti STBC is specified by matrix  $G_1$  whose codewords  $\begin{bmatrix} s_1 & -s_2^* \end{bmatrix}$  and  $\begin{bmatrix} s_2 & s_1^* \end{bmatrix}$  are the rows of  $G_1$ . Taking  $G_1$  as an example, we have

$$\begin{array}{ccc}
 & \text{Ant1} & \text{Ant 2} \\
 & \uparrow & \uparrow \\
 G_1^T = & \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix} & \begin{array}{l} \rightarrow \text{Symbol Time 1} \\ \rightarrow \text{Symbol Time 2} \end{array}
 \end{array} \tag{20}$$

The columns of  $G_1^T$  are orthogonal and so are the rows; for example,  $G_1^* G_1 = G_1 G_1^* = \|s\|^2 I_2$  where  $s = [s_1 \ s_2]^T$ . The Alamouti code allows the transmission of two symbols over two antennas in two consecutive symbol times. The code is perfect STBC and requires no bandwidth expansion. The sum of the channel-tap weighted code words plus noise is

$$\begin{aligned}
 Z &= h_1 \begin{bmatrix} s_1 \\ -s_2^* \end{bmatrix} + h_2 \begin{bmatrix} s_2 \\ s_1^* \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
 &= \begin{bmatrix} h_1 s_1 + h_2 s_2 \\ h_2 s_1^* - h_1 s_2^* \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
 &= Gh + n.
 \end{aligned} \tag{21}$$

The vector  $Z = Gh + n$  can be mapped into  $Y = Hs + n$  by changing the second element of  $Z = [Z_1 \ Z_2]^t$ , which is  $Z_2 = h_2 s_1^* - h_1 s_2^* + n_2$ , into its complex conjugate  $Z_2^* = h_2^* s_1 - h_1^* s_2 + n_2^*$ . Thus, we have

$$\begin{aligned}
 Y = Hs + n &= \begin{bmatrix} Z_1 \\ Z_2^* \end{bmatrix} = \begin{bmatrix} h_1 s_1 + h_2 s_2 \\ h_2^* s_1 - h_1^* s_2 \end{bmatrix} + \begin{bmatrix} N_1 \\ N_2 \end{bmatrix} \\
 &= \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}
 \end{aligned} \tag{22}$$

where  $N = [N_1 \ N_2]^T = [n_1 \ n_2^*]^T$ , and the corresponding channel-tap matrix  $H$  of the code matrix  $G_1$  is given by



$$H = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix}. \quad (23)$$

Assume that the channel taps  $h_1$  and  $h_2$  are obtained from perfect channel estimation. The receiver uses the matrix  $H$  to perform MRC as follows:

$$\begin{aligned} [X] &= \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\ &= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{ [H][s] + [N] \} \\ &= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|\hat{h}\|} [N] \\ &= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\ &= \|h\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\ &= \|h\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2^* n_1 - h_1 n_2 \end{bmatrix}. \end{aligned} \quad (24)$$

Similarly, one can perform the same operation to demodulate  $s_3$  and  $s_4$  by replacing  $s_1$  and  $s_2$  with  $s_3$  and  $s_4$ ,  $h_1$  and  $h_2$  with  $h_3$  and  $h_4$ , and  $X_1$  and  $X_2$  with  $X_3$  and  $X_4$ . The resulting matrix is

$$\begin{aligned} [X] &= \begin{bmatrix} X_3 \\ X_4 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\ &= \|h\| \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_3^* n_3 + h_4 n_4 \\ h_4^* n_3 - h_3 n_4 \end{bmatrix}. \end{aligned} \quad (25)$$

The block diagrams [5] for the transmitters and receiver are shown in Figure 7 and 8, respectively.

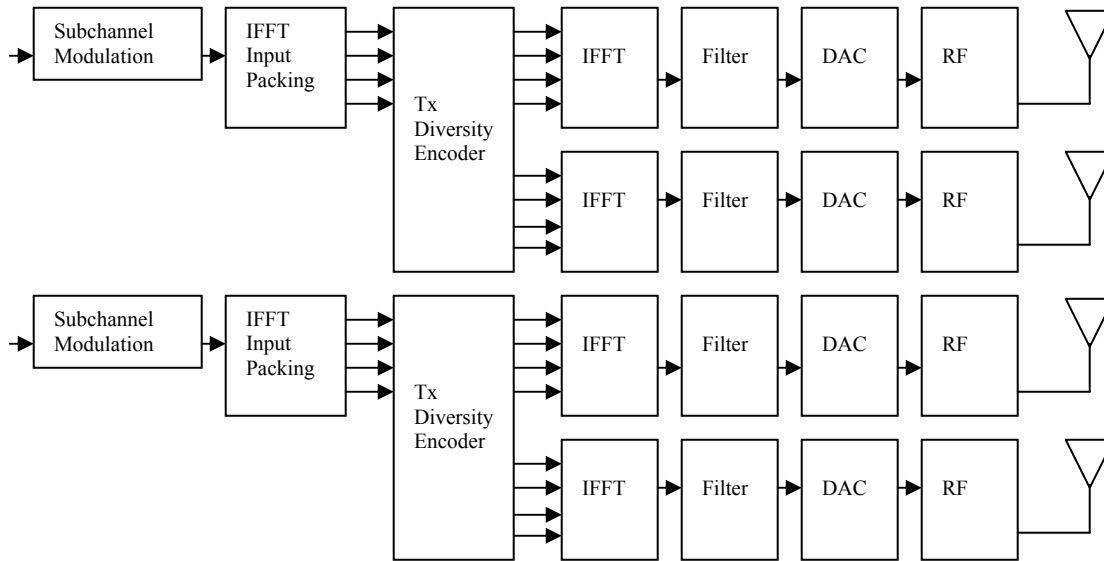


Figure 7. Modulator circuit from the base station for the Alamouti code.

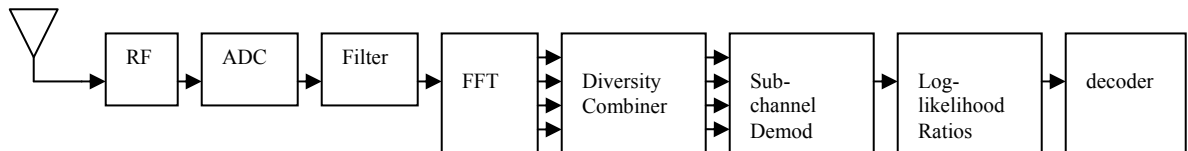


Figure 8. Demodulator circuit at the subscriber station for the Alamouti code.

There are four transmitting antennas on the BS side and one receiving antenna on the SS side. The only difference is that the modulator circuit can be repeated and reproduced. The modulator design is now simplified as the transmit diversity encoder only has to process two symbols instead of four.

## 2. Matrix $A_2$ (Variation of Matrix $A$ )

Consider the STBC defined by the matrix  $A_2$  in the 802.16 standard for OFDMA (shown in Appendix E):

$$\begin{array}{cccc}
 T1 & T2 & T3 & T4 \\
 \uparrow & \uparrow & \uparrow & \uparrow \\
 & & & 23
 \end{array}$$

$$A_2 = \begin{bmatrix} s_1 & -s_2^* & 0 & 0 \\ 0 & 0 & s_3 & -s_4^* \\ s_2 & s_1^* & 0 & 0 \\ 0 & 0 & s_4 & s_3^* \end{bmatrix} \begin{array}{l} \longrightarrow \text{Antenna 1} \\ \longrightarrow \text{Antenna 2} \\ \longrightarrow \text{Antenna 3} \\ \longrightarrow \text{Antenna 4.} \end{array} \quad (26)$$

From the matrix shown in (26), at time slot 1 (T1), symbols  $s_1$  and  $s_2$  are sent through Antenna 1 and Antenna 3, respectively. At time slot 2 (T2), symbols  $-s_2^*$  and  $s_1^*$  are sent through Antenna 1 and Antenna 3, respectively. At time slot 3 (T3), symbols  $s_3$  and  $s_4$  are sent through Antenna 2 and Antenna 4, respectively, and symbols  $-s_4^*$  and  $s_3^*$  are sent through Antenna 2 and Antenna 4 at time slot 4 (T4).

We use MRC to decode matrix  $A_2$ . The first step is to prove if matrix  $A_2^T$  is orthogonal. Matrix  $A_2^T$  is orthogonal only if  $(A_2^T)^* A_2^T = A_2' A_2^T = I_4$  where  $A_2^T$  represents the transpose of matrix  $A_2$ ,  $A_2'$  represents the conjugate of  $A_2$ , and  $(A_2^T)^*$  represents the conjugate transpose of  $A_2^T$ , which is the same as  $A_2'$ :

$$(A_2^T)^* = A_2' = \begin{bmatrix} s_1^* & -s_2 & 0 & 0 \\ 0 & 0 & s_3^* & -s_4 \\ s_2^* & s_1 & 0 & 0 \\ 0 & 0 & s_4^* & s_3 \end{bmatrix}. \quad (27)$$

Therefore,

$$(A_2^T)^* A_2^T = A_2' A_2^T = \begin{bmatrix} s_1^* s_1 + s_2 s_2^* & 0 & 0 & 0 \\ 0 & s_3 s_3^* + s_4 s_4^* & 0 & 0 \\ 0 & 0 & s_1^* s_1 + s_2 s_2^* & 0 \\ 0 & 0 & 0 & s_3 s_3^* + s_4 s_4^* \end{bmatrix} \quad (28)$$

and

$$A_2^T (A_2^T)^* = A_2^T A_2' = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & 0 & 0 \\ 0 & s_1 s_1^* + s_2 s_2^* & 0 & 0 \\ 0 & 0 & s_3 s_3^* + s_4 s_4^* & 0 \\ 0 & 0 & 0 & s_3 s_3^* + s_4 s_4^* \end{bmatrix}. \quad (29)$$

We can see that Equations (28) and (29) are quasi-orthogonal. Consequently, one can use the same modulator and demodulator as discussed for matrix  $A$  for this matrix.

### 3. Matrix $A_3$ (Variation of Matrix $A$ )

Consider the STBC defined by matrix  $A_3$  in the 802.16 standard for OFDMA (shown in Appendix E):

$$A_3 = \begin{array}{cccc} & \begin{array}{c} \uparrow \\ \text{T1} \end{array} & \begin{array}{c} \uparrow \\ \text{T2} \end{array} & \begin{array}{c} \uparrow \\ \text{T3} \end{array} & \begin{array}{c} \uparrow \\ \text{T4} \end{array} \\ \begin{bmatrix} s_1 & -s_2^* & 0 & 0 \\ 0 & 0 & s_3 & -s_4^* \\ 0 & 0 & s_4 & s_3^* \\ s_2 & s_1^* & 0 & 0 \end{bmatrix} & \longrightarrow & \text{Antenna 1} \\ & \longrightarrow & \text{Antenna 2} \\ & \longrightarrow & \text{Antenna 3} \\ & \longrightarrow & \text{Antenna 4.} \end{array} \quad (30)$$

From matrix given by (30), we see that in time slot 1 (T1), symbols  $s_1$  and  $s_2$  are sent through Antenna 1 and Antenna 4, respectively. In time slot 2 (T2), symbols  $-s_2^*$  and  $s_1^*$  are sent through Antenna 1 and Antenna 4, respectively. In time slot 3 (T3), symbols  $s_3$  and  $s_4$  are sent through Antenna 2 and Antenna 3, respectively, and symbols  $-s_4^*$  and  $s_3^*$  are sent through Antenna 2 and Antenna 3 in time slot 4 (T4).

We use MRC to decode matrix  $A_3$ . The first step is to prove if matrix  $A_3^T$  is orthogonal. Matrix  $A_3^T$  is orthogonal only if  $(A_3^T)^* A_3^T = A_3' A_3^T = I_4$  where  $A_3^T$  represents

the transpose of matrix  $A_3$ ,  $A_3'$  represents the conjugate of  $A_3$ , and  $(A_3^T)^*$  represents the conjugate transpose of  $A_3^T$ , which is the same as  $A_3'$ :

$$(A_3^T)^* = A_3' = \begin{bmatrix} s_1^* & -s_2 & 0 & 0 \\ 0 & 0 & s_3^* & -s_4 \\ 0 & 0 & s_4^* & s_3 \\ s_2^* & s_1 & 0 & 0 \end{bmatrix}. \quad (31)$$

Therefore,

$$(A_3^T)^* A_3^T = A_3' A_3^T = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & 0 & 0 \\ 0 & s_3 s_3^* + s_4 s_4^* & 0 & 0 \\ 0 & 0 & s_3 s_3^* + s_4 s_4^* & 0 \\ 0 & 0 & 0 & s_1 s_1^* + s_2 s_2^* \end{bmatrix} \quad (32)$$

and

$$A_3^T (A_3^T)^* = A_3^T A_3' = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & 0 & 0 \\ 0 & s_1 s_1^* + s_2 s_2^* & 0 & 0 \\ 0 & 0 & s_3 s_3^* + s_4 s_4^* & 0 \\ 0 & 0 & 0 & s_3 s_3^* + s_4 s_4^* \end{bmatrix}. \quad (33)$$

We can see that Equations (32) and (33) are quasi-orthogonal. As a result, one can use the same modulator and demodulator as discussed for matrix  $A$  for this matrix. Matrix  $A_3$ ,  $A_3' A_3^T \neq A_3^T A_3'$ , is similar to  $A_2$ , as it is quasi-orthogonal. As a result, one can use the same modulator and demodulator as discussed for matrix  $A$ .

#### 4. Matrix $D$ (Variation of Matrix $A$ Using Two Frequency Bands)

Consider the STBC defined by matrix  $A$  and given in the 802.16 standard:

$$D = \begin{array}{c} \begin{array}{cccc} \text{T1} & \text{T2} & \text{T1} & \text{T2} \\ \uparrow & \uparrow & \uparrow & \uparrow \\ \begin{bmatrix} s_1 & -s_2^* & 0 & 0 \\ s_2 & s_1^* & 0 & 0 \\ 0 & 0 & s_3 & -s_4^* \\ 0 & 0 & s_4 & s_3^* \end{bmatrix} & \begin{array}{l} \longrightarrow \text{Ant 1} \\ \longrightarrow \text{Ant 2} \\ \longrightarrow \text{Ant 1} \\ \longrightarrow \text{Ant 2} \end{array} & \left. \begin{array}{l} \text{Frequency} \\ \text{Band 1} \\ W_1 \\ \text{Frequency} \\ \text{Band 2} \\ W_2 \end{array} \right\} \end{array} \end{array} \quad (34)$$

In this case, we utilize two frequency bands but use two fewer time slots and two fewer transmitting antennas. We can see from the matrix given by (34) that in time slot 1 (T1), symbols  $s_1$  and  $s_2$  are sent using frequency band  $W_1$  through Antenna 1 and Antenna 2, respectively. In the same time slot, symbols  $s_3$  and  $s_4$  are sent through Antenna 1 and Antenna 2, respectively, using frequency band  $W_2$ . In time slot 2 (T2), symbols  $-s_2^*$  and  $s_1^*$  are sent using frequency band  $W_1$  through Antenna 1 and Antenna 2, respectively. In the same time slot, symbols  $-s_4^*$  and  $s_3^*$  are sent through Antenna 1 and Antenna 2, respectively, using frequency band  $W_2$ . The above matrix  $D$  has a diversity of two, since the same symbol is transmitted using two antennas. For example, symbol  $s_1$  is transmitted by Antenna 1 in time slot 1 (T1) and Antenna 2 in time slot 2 (T2) by the same frequency band,  $W_1$ . Matrix  $D$  has a code rate of two, since four symbols are transmitted in two time slots. The transmission process is illustrated in Figure 9.

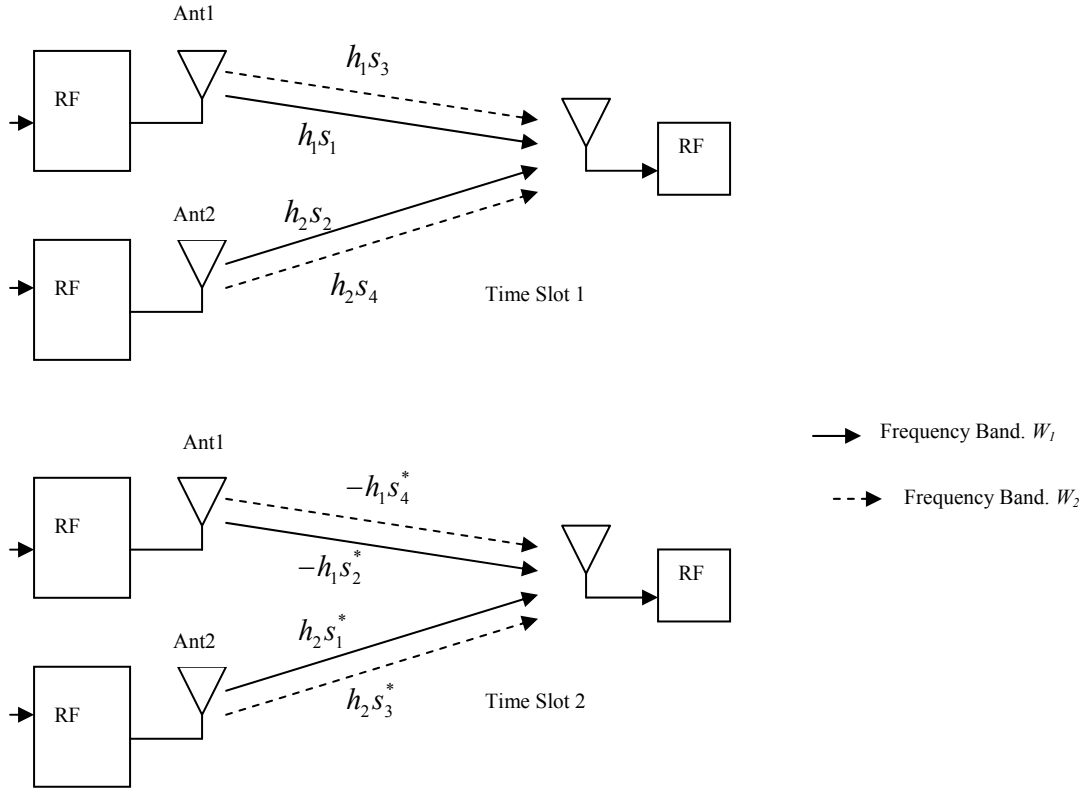


Figure 9. Transmission process for matrix  $D$ .

Observe from Figure 9 that one can reduce the matrix  $D$  to

$$D = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \\ s_3 & -s_4^* \\ s_4 & s_3^* \end{bmatrix} \quad (35)$$

and

$$(D^T)^* = D' = \begin{bmatrix} s_1^* & -s_2 \\ s_2^* & s_1 \\ s_3^* & -s_4 \\ s_4^* & s_3 \end{bmatrix}. \quad (36)$$

Therefore,

$$(D^T)^* D^T = D' D'^T = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & s_3 s_1^* + s_2 s_4^* & s_4 s_1^* - s_2 s_3^* \\ 0 & s_1 s_1^* + s_2 s_2^* & s_3 s_2^* - s_1 s_4^* & s_1 s_3^* + s_4 s_2^* \\ s_1 s_3^* + s_4 s_2^* & s_2 s_3^* - s_4 s_1^* & s_3 s_3^* + s_4 s_4^* & 0 \\ s_1 s_4^* - s_3 s_2^* & s_3 s_1^* + s_2 s_4^* & 0 & s_3 s_3^* + s_4 s_4^* \end{bmatrix} \quad (37)$$

and

$$D^T (D^T)^* = D^T D' = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* + s_3 s_3^* + s_4 s_4^* & 0 \\ 0 & s_1 s_1^* + s_2 s_2^* + s_3 s_3^* + s_4 s_4^* \end{bmatrix}. \quad (38)$$

From (37) and (38) we see that the matrix  $D$  is quasi-orthogonal and can be demodulated separately using two frequency bands,  $W_1$  and  $W_2$ .

In frequency band  $W_1$ ,

$$\begin{aligned} Z &= \begin{bmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\ &= \begin{bmatrix} s_1 h_1 + s_2 h_2 \\ -s_2^* h_1 + s_1^* h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \end{aligned} \quad (39)$$

Mapping this to  $[Y] = [H][s] + [n]$ , we get

$$Y = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \quad (40)$$

The next step is to use the channel taps  $[\hat{h}_1 \ \hat{h}_2]$  obtained via the channel estimation process to estimate  $h_1$  and  $h_2$ . The receiver can then use  $[\hat{H}]$  to perform MRC. For



simplicity, assume perfect channel estimation, where  $[\hat{h}_1 \ \hat{h}_2] \rightarrow [h_1 \ h_2]$  and apply MRC on  $Y$  to obtain the corresponding decision statistics vector  $[X]$ , where

$$\begin{aligned}
[X] &= \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\
&= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{[H][s] + [N]\} \\
&= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|h\|} [N] \\
&= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\
&= \|h\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \|h\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2^* n_1 - h_1 n_2 \end{bmatrix}
\end{aligned} \tag{41}$$

and  $\|h\|$  is defined as

$$\|h\| = \sqrt{|h_1|^2 + |h_2|^2}. \tag{42}$$

In frequency band  $W_2$ ,

$$\begin{aligned}
Z &= \begin{bmatrix} s_3 & s_4 \\ -s_4^* & s_3^* \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \begin{bmatrix} s_3 h_1 + s_4 h_2 \\ -s_4^* h_1 + s_3^* h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}.
\end{aligned} \tag{43}$$

Mapping this to  $[Y] = [H][s] + [n]$ , we get

$$Y = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{44}$$

The corresponding decision statistics vector  $[X]$  is

$$\begin{aligned}
[X] &= \begin{bmatrix} X_3 \\ X_4 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\
&= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{[H][s] + [N]\} \\
&= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|\hat{h}\|} [N] \\
&= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\
&= \|h\| \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \|h\| \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2^* n_1 - h_1 n_2 \end{bmatrix}. \tag{45}
\end{aligned}$$

From Equations (41) and (45), we see that the two noise components  $n_1$  and  $n_2$  are independent since  $E\left[(h_1^* n_1 + h_2 n_2)(h_2^* n_1 - h_1 n_2)^*\right] = 0$ .

The above scheme uses two less transmit antennas and two less time slots to transmit the same data but uses two frequency bands. This scheme is useful for high speed data rate communications since it can achieve a code rate of two.

The block diagrams [5] for the transmitters and receiver are shown in Figure 10 and 11, respectively.

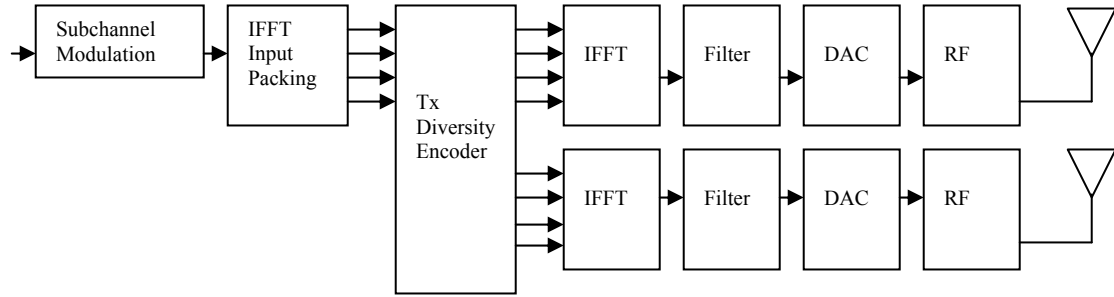


Figure 10. Modulator for matrix  $D$ .

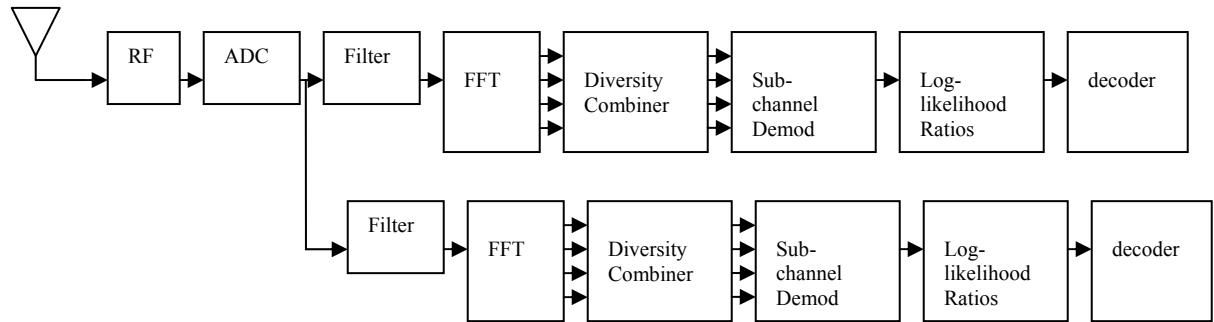


Figure 11. Demodulator for matrix  $D$ .

There are two transmitting antennas on the BS side and one receive antenna on the SS side. In this case, the receiver will have higher complexity compared to the receiver for matrix  $A$  since it receives four symbols at a time slot from two different frequency bands. To demodulate the received signal, two matched filters are used to separate the signal and process them separately, as seen in Figure 11.

## B. PERFORMANCE EVALUATION

All of the matrices discussed in this section are quasi-orthogonal, and MRC can be used for detection. Therefore, one can use closed form equations to evaluate their performance [4].

Assume a Rayleigh fading channel. The independent, identically distributed (iid) subchannel tap magnitudes  $|h_i|$ 's are Rayleigh distributed with normalized mean square value  $E(|h_i|^2) = 1$ . The combined channel tap magnitude  $\|h\|^2 = \sum_{l=1}^L |h_l|^2$  is a  $\chi^2$ -density function with  $2L$  degrees of freedom. The corresponding bit error probability from [2] is given by

$$\begin{aligned} P_b &\approx \frac{N_n}{2 \log_2 M} \left[ 1 - \mu \sum_{k=0}^{L-1} \binom{2k}{k} \left( \frac{1-\mu^2}{4} \right)^k \right] \\ &\approx \frac{N_n}{2 \log_2 M} \left( \frac{1-\mu}{2} \right)^L \sum_{l=0}^{L-1} \binom{L-1+l}{l} \left( \frac{1+\mu}{2} \right)^l \end{aligned} \quad (46)$$

where

$$\mu = \sqrt{\frac{\alpha SNR}{1 + \alpha SNR}}. \quad (47)$$

For QPSK, the minimum Euclidean distance is  $d_{\min} = \sqrt{2\mathcal{E}_s}$  where the diversity symbol energy is  $\mathcal{E}_s$ . With  $\sigma^2 = N_0/2$ , one obtains  $\alpha SNR = \mathcal{E}_s / 2N_0 = E_s / 2LN_0$ . Furthermore,  $N_n = 2, M = 4$ ; therefore, the bit error probability for QPSK from [4] is

$$\begin{aligned} P_b &\approx \frac{1}{2} \left[ 1 - \mu \sum_{k=0}^{L-1} \binom{2k}{k} \left( \frac{1-\mu^2}{4} \right)^k \right] \\ &= \left( \frac{1-\mu}{2} \right)^L \sum_{l=0}^{L-1} \binom{L-1+l}{l} \left( \frac{1+\mu}{2} \right)^l. \end{aligned} \quad (48)$$

For MQAM, the minimum Euclidean distance is  $d_{\min} = \sqrt{6\mathcal{E}_s / (M-1)}$ . Using  $\sigma^2 = N_0/2$ , one obtains  $\alpha SNR = 3(\mathcal{E}_s / N_0) / 2(M-1) = 3(E_s / LN_0) / 2(M-1)$ . Furthermore,  $N_n = 4 - 4 / \sqrt{M}$ ; therefore, the bit error probability for MQAM from [4] is given by

$$\begin{aligned}
P_b &\approx \frac{2 - 2/\sqrt{M}}{\log_2 M} \left[ 1 - \mu \sum_{k=0}^{L-1} \binom{2k}{k} \left( \frac{1 - \mu^2}{4} \right)^k \right] \\
&\approx \frac{4 - 4/\sqrt{M}}{\log_2 M} \left( \frac{1 - \mu}{2} \right)^L \sum_{l=0}^{L-1} \binom{L-1+l}{l} \left( \frac{1 - \mu}{2} \right)^l.
\end{aligned} \tag{49}$$

The two closed forms plots for QPSK and 16QAM are illustrated in Figure 12. The MATLAB code is attached in Appendix A.

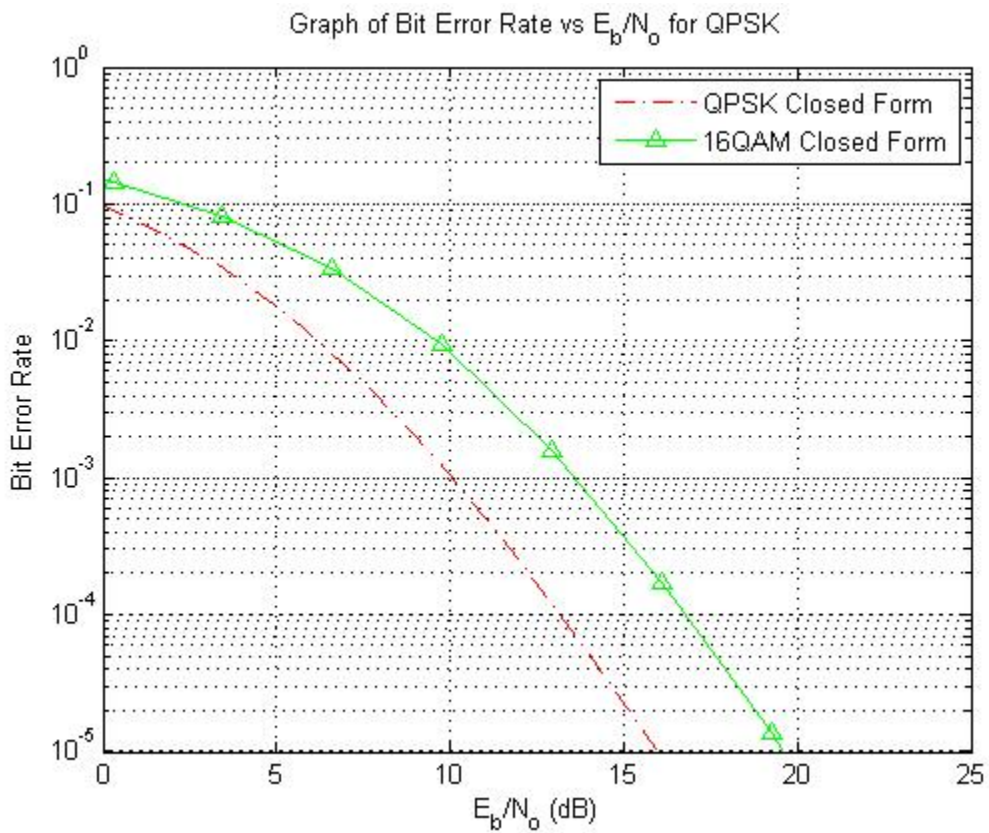


Figure 12. Graph of bit error rate vs  $E_b/N_0$ (dB) for QPSK and 16QAM.

## C. CHAPTER SUMMARY

The variations of the matrix  $A$  are quasi-orthogonal. Therefore, one can use a simple modulator and demodulator to implement the design. In addition, one can utilize two frequency bands to transmit at a faster rate at the expense of using two frequency bands. In the next chapter, we will evaluate matrix  $B$  where eight symbols are transmitted using four transmitting antennas and four time slots.

THIS PAGE INTENTIONALLY LEFT BLANK

## IV. IMPLEMENTATION AND PERFORMANCE EVALUATION OF WIMAX NON-ORTHOGONAL STC FOR OFDMA

In this section, we shall look at matrix  $B$  defined in the 802.16 standard to evaluate the STC and its variations. One can assume there are multiple antennas on the BS side and one receiving antenna on the SS side.

### A. MATRIX $B$ IN 802.16

#### 1. Matrix $B$

Consider the STBC defined by matrix  $B$  given in the 802.16 standard for OFDMA (shown in Appendix E):

$$\begin{array}{cccc}
 & \text{T1} & \text{T2} & \text{T3} & \text{T4} \\
 & \uparrow & \uparrow & \uparrow & \uparrow \\
 B = & \begin{bmatrix} s_1 & -s_2^* & s_5 & -s_7^* \\ s_2 & s_1^* & s_6 & -s_8^* \\ s_3 & -s_4^* & s_7 & s_5^* \\ s_4 & s_3^* & s_8 & s_6^* \end{bmatrix} & \begin{array}{l} \longrightarrow \\ \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{array} & \begin{array}{l} \text{Antenna 1} \\ \text{Antenna 2} \\ \text{Antenna 3} \\ \text{Antenna 4.} \end{array}
 \end{array} \quad (50)$$

We can see from the matrix shown in (50) that in time slot 1 (T1), symbols  $s_1, s_2, s_3$  and  $s_4$  are sent through Antenna 1, Antenna 2, Antenna 3 and Antenna 4, respectively. In time slot 2 (T2), symbols  $-s_2^*, s_1^*, -s_4^*$  and  $s_3^*$  are sent through Antenna 1, Antenna 2, Antenna 3 and Antenna 4, respectively. In time slot 3 (T3), symbols  $s_5, s_6, s_7$  and  $s_8$  are sent through Antenna 1, Antenna 2, Antenna 3 and Antenna 4, respectively, and in time slot 4 (T4), symbols  $-s_7^*, -s_8^*, s_5^*$  and  $s_6^*$  are sent through Antenna 1, Antenna 2, Antenna 3 and Antenna 4, respectively. This matrix uses four time slots and four antennas to transmit eight symbols. Use of matrix  $B$  results in a diversity of two since the same symbol is transmitted twice using two antennas. For example, symbol  $s_1$  is transmitted by Antenna 1 in timeslot 1 (T1) and Antenna 2 in time slot 2 (T2). Use of



matrix  $B$  also results in a code rate of two since eight symbols are transmitted in four time slots. The transmission process at different time slots is illustrated in Figure 13.

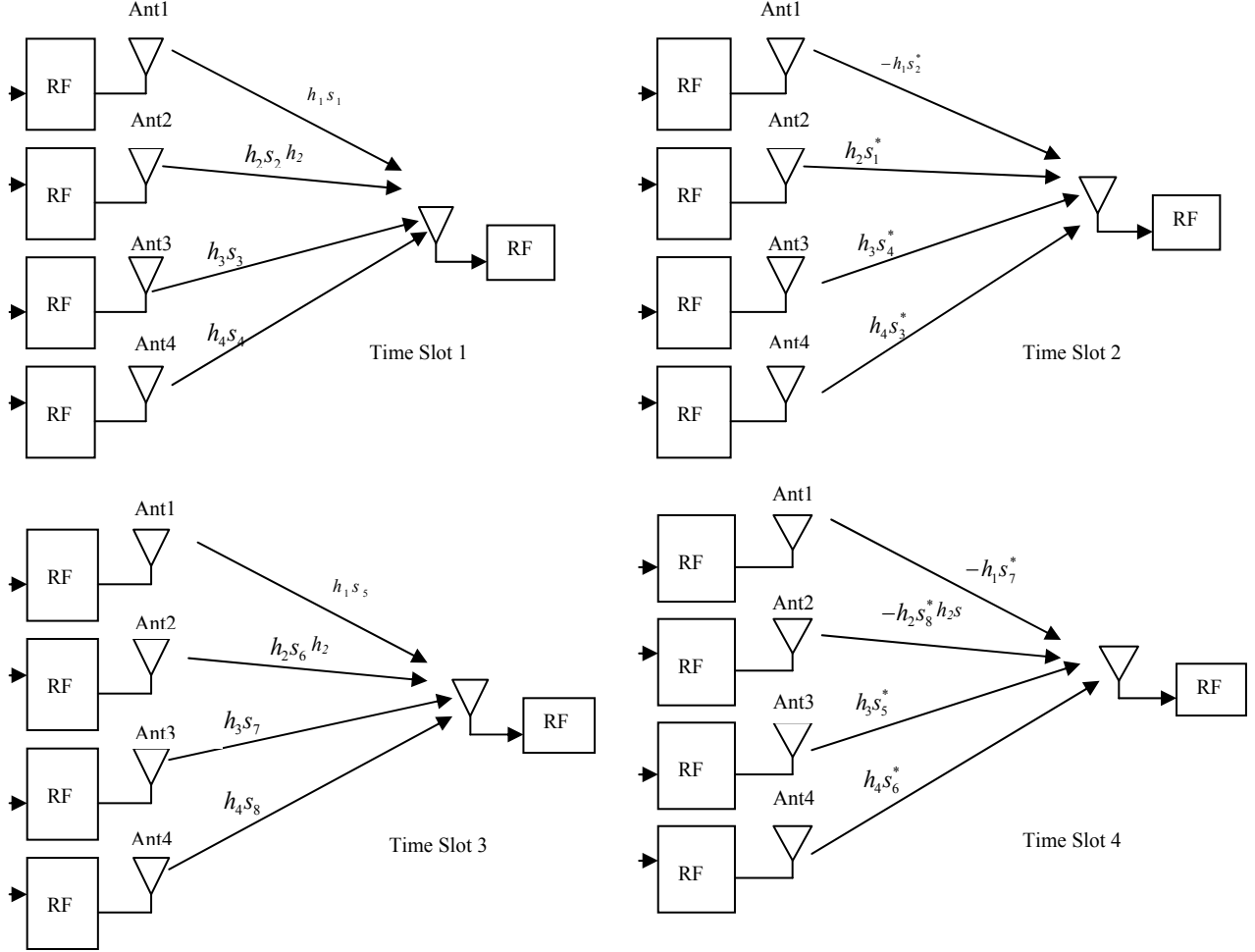


Figure 13. Transmission process for matrix  $B$ .

We use MRC to decode matrix  $B$ . The first step is to determine if matrix  $B$  is orthogonal. Matrix  $B$  is orthogonal if and only if  $(B^T)^* B^T = B' B^T = I_4$  where  $B^T$  represents the transpose of matrix  $B$ ,  $B'$  represents the conjugate of  $B$ , and  $(B^T)^*$  represents the conjugate transpose of  $B^T$ , which is the same as  $B'$ :

$$(B^T)^* = B' = \begin{bmatrix} s_1^* & -s_2 & s_5^* & -s_7 \\ s_2^* & s_1 & s_6^* & -s_8 \\ s_3^* & -s_4 & s_7^* & s_5 \\ s_4^* & s_3 & s_8^* & s_6 \end{bmatrix}. \quad (51)$$

Therefore,

$$(B^T)^* B^T = B' B^T = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & s_6 s_5^* + s_7 s_8^* & s_3 s_1^* + s_2 s_4^* & s_4 s_1^* - s_2 s_3^* \\ +s_5 s_5^* + s_7 s_7^* & & & -s_7 s_6^* + s_8 s_5^* \\ s_5 s_6^* + s_8 s_7^* & s_1 s_1^* + s_2 s_2^* & s_3 s_2^* - s_1 s_4^* & s_1 s_3^* + s_4 s_2^* \\ +s_6 s_6^* + s_8 s_8^* & +s_7 s_6^* - s_8 s_5^* & & \\ s_1 s_3^* + s_4 s_2^* & s_2 s_2^* - s_4 s_1^* & s_1 s_1^* + s_2 s_2^* & s_5 s_6^* + s_8 s_7^* \\ -s_5 s_8^* + s_6 s_7^* & +s_5 s_5^* + s_7 s_7^* & & \\ s_1 s_4^* - s_3 s_2^* & s_3 s_1^* + s_2 s_4^* & s_6 s_5^* + s_7 s_8^* & s_3 s_3^* + s_4 s_4^* \\ -s_5 s_8^* + s_6 s_7^* & & & +s_6 s_6^* + s_8 s_8^* \end{bmatrix} \quad (52)$$

and



The vector  $Z=Gh+n$  can be mapped into  $Y=Hs+N$  by changing the second element of  $Z=[Z_1 Z_2 Z_3 Z_4]^T$ , which is  $Z_2 = -s_2^*h_1 + s_1^*h_2 - s_4^*h_3 + s_3^*h_4$ , into its complex conjugate  $Z_2^* = -s_2h_1^* + s_1h_2^* - s_4h_3^* + s_3h_4^*$ , and the fourth element, which is from  $Z_4 = -s_7^*h_1 - s_8^*h_2 + s_5^*h_3 + s_6^*h_4$ , into its complex conjugate  $Z_4^* = -s_7h_1^* - s_8h_2^* + s_5h_3^* + s_6h_4^*$ . Thus,

$$\begin{aligned}
Y = Hs + n &= \begin{bmatrix} Z_1 \\ Z_2^* \\ Z_3 \\ Z_4^* \end{bmatrix} = \begin{bmatrix} s_1h_1 + s_2h_2 + s_3h_3 + s_4h_4 \\ -s_2h_1^* + s_1h_2^* - s_4h_3^* + s_3h_4^* \\ s_5h_1 + s_6h_2 + s_7h_3 + s_8h_4 \\ -s_7h_1^* - s_8h_2^* + s_5h_3^* + s_6h_4^* \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix} \\
&= \begin{bmatrix} h_1 & h_2 & h_3 & h_4 & 0 & 0 & 0 & 0 \\ h_2^* & -h_1^* & h_4^* & -h_3^* & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & h_4 \\ 0 & 0 & 0 & 0 & h_3^* & h_4^* & -h_1^* & -h_2^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix} = [H][s] + [n]. \tag{55}
\end{aligned}$$

From Equation (55), there are eight unknown symbols  $s_1, s_2, s_3, s_5, s_6, s_7$  and  $s_8$ . Since the noise components are not independent, the eight equations are not enough to accurately decode the symbols even if one knows  $h_1, h_2, h_3$  and  $h_4$  from the channel estimation. We require at least two copies of the same data to accurately decode the symbols. The received  $Z$  matrix for two receiving antennas is shown in (56), and the corresponding  $Y$  matrix is shown in (57). The MATLAB code attached in Appendix C is used to simulate the results for both one receiving antenna and two receiving antennas. For the two receiving antenna case,

$$Z = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_1 & s_2 & s_3 & s_4 \\ -s_2^* & s_1^* & -s_4^* & s_3^* & -s_2^* & s_1^* & -s_4^* & s_3^* \\ s_5 & s_6 & s_7 & s_8 & s_5 & s_6 & s_7 & s_8 \\ -s_7^* & -s_8^* & s_5^* & s_6^* & -s_7^* & -s_8^* & s_5^* & s_6^* \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{14} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{24} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \quad (56)$$

$$= \begin{bmatrix} s_1 h_{11} + s_2 h_{12} + s_3 h_{13} + s_4 h_{14} + s_1 h_{21} + s_2 h_{22} + s_3 h_{23} + s_4 h_{24} \\ -s_2^* h_{11} + s_1^* h_{12} - s_4^* h_{13} + s_3^* h_{14} - s_2^* h_{21} + s_1^* h_{22} - s_4^* h_{23} + s_3^* h_{24} \\ s_5 h_{11} + s_6 h_{12} + s_7 h_{13} + s_8 h_{14} + s_5 h_{21} + s_6 h_{22} + s_7 h_{23} + s_8 h_{24} \\ -s_7^* h_{11} - s_8^* h_{12} + s_5^* h_{13} + s_6^* h_{14} - s_7^* h_{21} - s_8^* h_{22} + s_5^* h_{23} + s_6^* h_{24} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}$$

and

$$Y = HS + n = \begin{bmatrix} Z_1 \\ Z_2^* \\ Z_3 \\ Z_4^* \end{bmatrix} = \begin{bmatrix} s_1 h_{11} + s_2 h_{12} + s_3 h_{13} + s_4 h_{14} + s_1 h_{21} + s_2 h_{22} + s_3 h_{23} + s_4 h_{24} \\ -s_2^* h_{11} + s_1^* h_{12} - s_4^* h_{13} + s_3^* h_{14} - s_2^* h_{21} + s_1^* h_{22} - s_4^* h_{23} + s_3^* h_{24} \\ s_5 h_{11} + s_6 h_{12} + s_7 h_{13} + s_8 h_{14} + s_5 h_{21} + s_6 h_{22} + s_7 h_{23} + s_8 h_{24} \\ -s_7^* h_{11} - s_8^* h_{12} + s_5^* h_{13} + s_6^* h_{14} - s_7^* h_{21} - s_8^* h_{22} + s_5^* h_{23} + s_6^* h_{24} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix} \quad (57)$$

$$= \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{21} & h_{22} & h_{23} & h_{24} \\ h_{12}^* & -h_{11}^* & h_{14}^* & -h_{13}^* & h_{23}^* & h_{24}^* & -h_{21}^* & -h_{22}^* \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{11} & h_{12} & h_{13} & h_{14} \\ h_{22}^* & -h_{21}^* & h_{24}^* & -h_{23}^* & h_{13}^* & h_{14}^* & -h_{11}^* & -h_{12}^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \\ n_3 \\ n_4^* \end{bmatrix}$$

The block diagrams [5] for the four transmitters and two receivers design are illustrated in Figure 14 and 15, respectively.

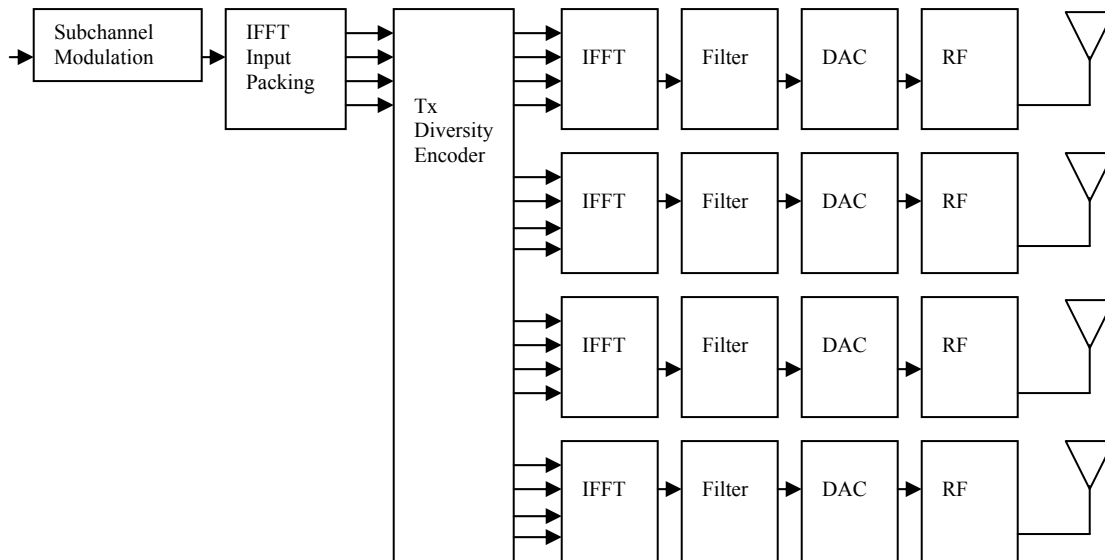


Figure 14. Modulator circuit from the base station for matrix  $B$ .

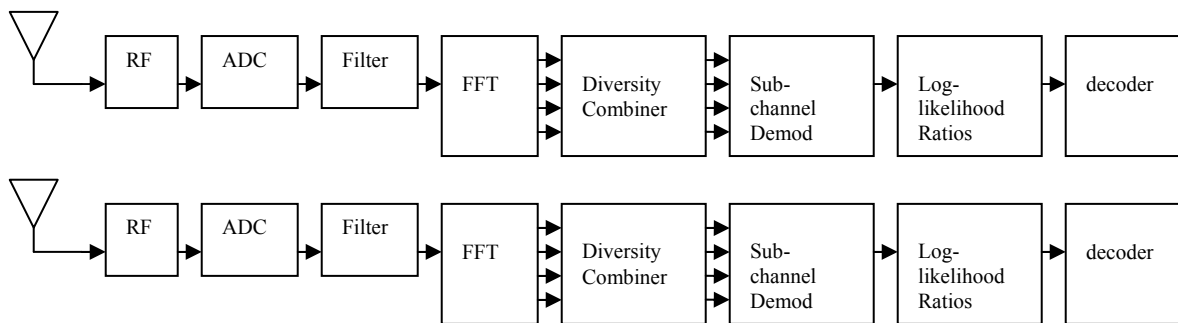


Figure 15. Demodulator circuit at the subscriber station for matrix  $B$  with two receiving antennas.

There are four transmitting antennas on the BS side and two receiving antennas on the SS side. The two receivers at the SS provide two copies of the message to accurately demodulate the received symbols.



$$(B_2^T)^* = B' = \begin{bmatrix} s_1^* & -s_2 & s_5^* & -s_7 \\ s_2^* & s_1 & s_6^* & -s_8 \\ s_4^* & s_3 & s_8^* & s_6 \\ s_3^* & -s_4 & s_7^* & s_5 \end{bmatrix}. \quad (59)$$

Therefore,

$$(B_2^T)^* B_2^T = B' B_2^T = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & s_6 s_5^* + s_7 s_8^* & s_4 s_1^* - s_2 s_3^* & s_3 s_1^* + s_2 s_4^* \\ +s_5 s_5^* + s_7 s_7^* & & -s_7 s_6^* + s_8 s_5^* & \\ s_5 s_6^* + s_8 s_7^* & s_1 s_1^* + s_2 s_2^* & s_1 s_3^* + s_4 s_2^* & s_3 s_2^* - s_1 s_4^* \\ +s_6 s_6^* + s_8 s_8^* & & & -s_8 s_5^* + s_7 s_6^* \\ s_1 s_4^* - s_3 s_2^* & s_3 s_1^* + s_2 s_4^* & s_3 s_3^* + s_4 s_4^* & s_6 s_5^* + s_7 s_8^* \\ -s_6 s_7^* + s_5 s_8^* & & +s_6 s_6^* + s_8 s_8^* & \\ s_1 s_3^* + s_4 s_2^* & s_2 s_3^* - s_4 s_1^* & s_5 s_6^* + s_8 s_7^* & s_3 s_3^* + s_4 s_4^* \\ & -s_5 s_8^* + s_6 s_7^* & & +s_7 s_7^* + s_8 s_8^* \end{bmatrix} \quad (60)$$

and



$$B_2^T (B_2^T)^* = B_2^T B' = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & s_1 s_5^* + s_2 s_6^* & s_3 s_5 - s_1 s_7 \\ +s_3 s_3^* + s_4 s_4^* & & +s_3 s_7^* + s_4 s_8^* & s_4 s_6 - s_2 s_8 \\ & 0 & s_1 s_1^* + s_2 s_2^* & s_1 s_6^* - s_2 s_5^* \\ & & +s_3 s_3^* + s_4 s_4^* & +s_3 s_8^* - s_4 s_7^* \\ & & & s_6 s_3^* - s_5 s_4^* \\ & & & +s_7 s_2^* + s_8 s_1^* \\ s_5 s_1^* + s_6 s_2^* & s_1 s_6 - s_2 s_5 & s_5 s_5^* + s_6 s_6^* & 0 \\ +s_7 s_3^* + s_8 s_4^* & +s_3 s_8 + s_4 s_7 & +s_7 s_7^* + s_8 s_8^* & \\ s_3 s_5^* - s_1 s_7^* & s_2 s_7^* - s_1 s_8^* & 0 & s_5 s_5^* + s_6 s_6^* \\ -s_2 s_8^* + s_4 s_6^* & +s_3 s_6^* - s_4 s_5^* & & +s_7 s_7^* + s_8 s_8^* \end{bmatrix}. \quad (61)$$

From (60) and (61), we can see that matrix  $B_2$  is non-orthogonal. We use simulations to evaluate performance when it is used.

### 3. Matrix $E$ (Variation of Matrix $B$ Using Four Frequency Bands)

Consider the STBC defined by matrix  $B$  given in the 802.16 standard:

$$E = \begin{bmatrix} s_1 & -s_2^* & s_5 & -s_7^* \\ s_2 & s_1^* & s_6 & -s_8^* \\ s_3 & -s_4^* & s_7 & s_5^* \\ s_4 & s_3^* & s_8 & s_6^* \end{bmatrix} \quad (62)$$

The diagram illustrates the matrix  $E$  in the context of four frequency bands and two time slots. The matrix is a 4x4 block matrix. The first two columns correspond to Time Slot 1 (T1) and Time Slot 2 (T2). The first two rows correspond to Frequency Band 1 ( $W_1$ ) and Frequency Band 2 ( $W_2$ ). The last two rows correspond to Frequency Band 3 ( $W_3$ ) and Frequency Band 4 ( $W_4$ ). The matrix elements are  $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8$  and their conjugates. The diagram also shows the antenna configuration: Antenna 1 and Antenna 2 are used in each time slot, with Antenna 1 and Antenna 2 being the same physical antennas in different time slots.

In this case, we utilize four frequency bands but use two fewer time slots and two fewer transmitting antennas. We can see from the matrix given by (62) that in time slot 1

(T1), symbols  $s_1$  and  $s_2$ ,  $s_3$  and  $s_4$ ,  $s_5$  and  $s_7$ ,  $s_8$  and  $s_6$  are sent by Antenna 1 and Antenna 2 using frequency band  $W_1$ ,  $W_2$ ,  $W_3$  and  $W_4$ , respectively. In time slot 2 (T2), symbols  $-s_2^*$  and  $s_1^*$ ,  $-s_4^*$  and  $s_3^*$ ,  $-s_7^*$  and  $s_5^*$ ,  $-s_8^*$  and  $s_6^*$  are sent by Antenna 1 and Antenna 2 using frequency band  $W_1$ ,  $W_2$ ,  $W_3$  and  $W_4$ , respectively. Use of matrix  $E$  results in a diversity of two since the same symbol is transmitted using two antennas. For example, symbol  $s_1$  is transmitted by Antenna 1 in time slot 1 (T1) and Antenna 2 in time slot 2 (T2) over the same frequency band  $W_1$ . Use of matrix  $E$  results in a code rate of four since eight symbols are transmitted in two time slots. The transmission process for different time slots is illustrated in Figure 16.

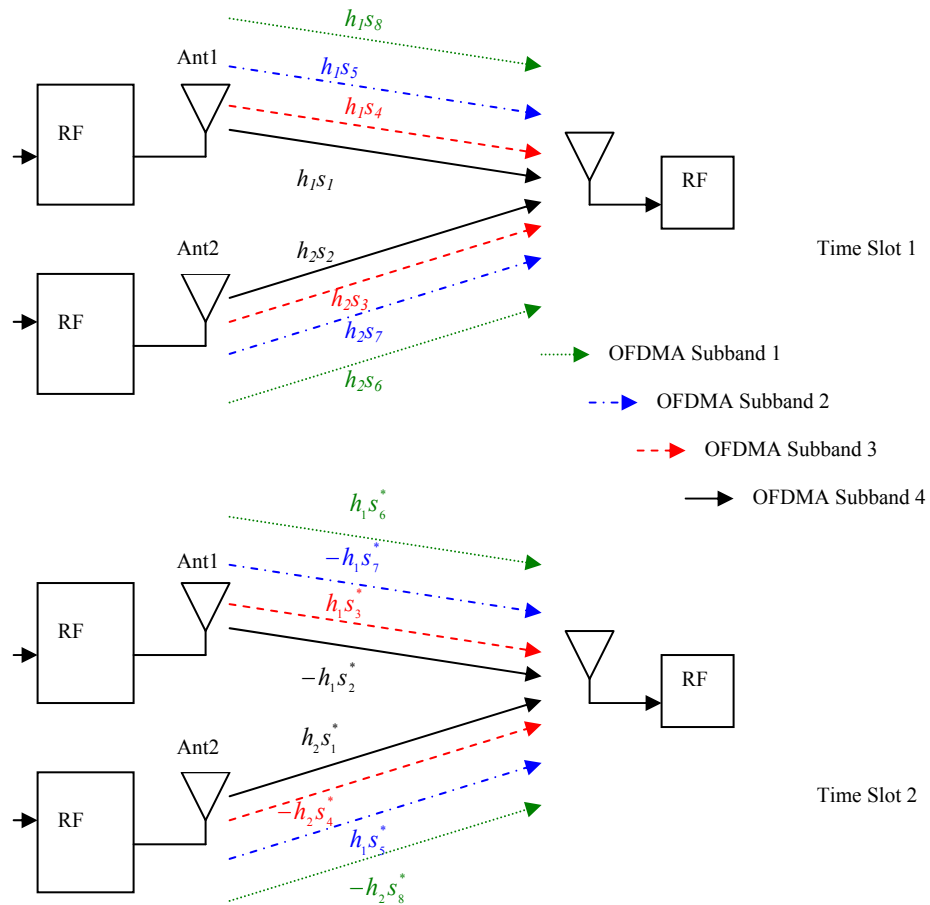


Figure 16. Transmission process for matrix  $E$ .

Observe from Figure 16 that one can reduce the matrix  $E$  to

$$E = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \\ s_3 & -s_4^* \\ s_4 & s_3^* \\ s_5 & -s_7^* \\ s_6 & s_5^* \\ s_7 & -s_8^* \\ s_8 & s_6^* \end{bmatrix} \quad (63)$$

and

$$(E^T)^* = E' = \begin{bmatrix} s_1^* & -s_2 \\ s_2^* & s_1 \\ s_3^* & -s_4 \\ s_4^* & s_3 \\ s_5^* & -s_7 \\ s_7^* & s_5 \\ s_6^* & -s_8 \\ s_8^* & s_6 \end{bmatrix}. \quad (64)$$

Therefore,

$$(E^T)^* E^T = E' E'^T = \begin{bmatrix} s_1 s_1^* + s_2 s_2^* & 0 & s_3 s_1^* + s_2 s_4^* & s_4 s_1^* - s_2 s_3^* & s_5 s_1^* + s_2 s_7^* & s_7 s_1^* - s_2 s_5^* & s_6 s_1^* + s_2 s_8^* & s_8 s_1^* - s_2 s_6^* \\ 0 & s_1 s_1^* + s_2 s_2^* & s_3 s_2^* - s_1 s_4^* & s_1 s_3^* + s_4 s_2^* & s_5 s_2^* - s_1 s_7^* & s_1 s_5^* + s_7 s_2^* & s_6 s_2^* - s_1 s_8^* & s_1 s_6^* + s_8 s_2^* \\ s_1 s_3^* + s_4 s_2^* & s_2 s_3^* - s_4 s_1^* & s_3 s_3^* + s_4 s_4^* & 0 & s_5 s_3^* + s_4 s_7^* & s_7 s_3^* - s_4 s_5^* & s_6 s_3^* + s_4 s_8^* & s_8 s_3^* - s_4 s_6^* \\ s_1 s_4^* - s_3 s_2^* & s_3 s_1^* + s_2 s_4^* & 0 & s_3 s_3^* + s_4 s_4^* & s_5 s_4^* - s_3 s_7^* & s_3 s_5^* + s_7 s_4^* & s_6 s_4^* - s_3 s_8^* & s_3 s_6^* + s_8 s_4^* \\ s_1 s_5^* + s_7 s_2^* & s_2 s_5^* - s_7 s_1^* & s_3 s_5^* + s_7 s_4^* & s_4 s_5^* - s_7 s_3^* & s_5 s_5^* + s_7 s_7^* & 0 & s_6 s_5^* + s_7 s_8^* & s_8 s_5^* - s_7 s_6^* \\ s_1 s_7^* - s_5 s_2^* & s_5 s_1^* + s_2 s_7^* & s_3 s_7^* - s_5 s_4^* & s_5 s_3^* + s_4 s_7^* & 0 & s_5 s_5^* + s_7 s_7^* & s_6 s_7^* - s_5 s_8^* & s_5 s_6^* + s_8 s_7^* \\ s_1 s_6^* + s_8 s_2^* & s_2 s_6^* - s_8 s_1^* & s_3 s_6^* + s_8 s_4^* & s_4 s_6^* - s_8 s_3^* & s_5 s_6^* + s_8 s_7^* & s_7 s_6^* - s_8 s_5^* & s_6 s_6^* + s_8 s_8^* & 0 \\ s_1 s_8^* - s_6 s_2^* & s_6 s_1^* + s_2 s_8^* & s_3 s_8^* - s_6 s_4^* & s_6 s_3^* + s_4 s_8^* & s_5 s_8^* - s_6 s_7^* & s_6 s_5^* + s_7 s_8^* & 0 & s_6 s_6^* + s_8 s_8^* \end{bmatrix} \quad (65)$$

and



$$\begin{aligned}
[X] &= \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\
&= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{[H][s] + [N]\} \\
&= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|\hat{h}\|} [N] \\
&= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\
&= \|h\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \|h\| \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2^* n_1 - h_1 n_2 \end{bmatrix} \tag{69}
\end{aligned}$$

and  $\|h\|$  is defined as

$$\|h\| = \sqrt{|h_1|^2 + |h_2|^2}. \tag{70}$$

In frequency band  $W_2$ ,

$$\begin{aligned}
Z &= \begin{bmatrix} s_3 & s_4 \\ -s_4^* & s_3^* \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \begin{bmatrix} s_3 h_1 + s_4^* h_2 \\ -s_4^* h_1 + s_3 h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{71}
\end{aligned}$$

Mapping this to  $[Y] = [H][s] + [n]$ , we get

$$Y = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{72}$$

The corresponding decision statistics vector  $[X]$  is,

$$\begin{aligned}
[X] &= \begin{bmatrix} X_3 \\ X_4 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\
&= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{[H][s] + [N]\} \\
&= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|\hat{h}\|} [N] \\
&= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\
&= \|h\| \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \|h\| \begin{bmatrix} s_3 \\ s_4 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2 n_1 - h_1 n_2 \end{bmatrix}.
\end{aligned} \tag{73}$$

In frequency band  $W_3$ ,

$$\begin{aligned}
Z &= \begin{bmatrix} s_5 & s_7 \\ -s_7^* & s_5^* \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \begin{bmatrix} s_5 h_1 + s_7^* h_2 \\ -s_7^* h_1 + s_5 h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}.
\end{aligned} \tag{74}$$

Mapping this to  $[Y] = [H][s] + [n]$ , we get

$$Y = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} s_5 \\ s_7 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{75}$$

The corresponding decision statistics vector  $[X]$  is,

$$\begin{aligned}
[X] &= \begin{bmatrix} X_5 \\ X_7 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\
&= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{[H][s] + [N]\} \\
&= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|\hat{h}\|} [N] \\
&= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\
&= \|h\| \begin{bmatrix} s_5 \\ s_7 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \|h\| \begin{bmatrix} s_5 \\ s_7 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2 n_1 - h_1 n_2 \end{bmatrix}.
\end{aligned} \tag{76}$$

In frequency band  $W_4$ ,

$$\begin{aligned}
Z &= \begin{bmatrix} s_6 & s_8 \\ -s_8^* & s_6^* \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \begin{bmatrix} s_6 h_1 + s_8^* h_2 \\ -s_8^* h_1 + s_6 h_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}.
\end{aligned} \tag{77}$$

Mapping this to  $[Y] = [H][s] + [n]$ , we get

$$Y = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_6 \\ s_8 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{78}$$

The corresponding decision statistics vector  $[X]$  is,

$$\begin{aligned}
[X] &= \begin{bmatrix} X_6 \\ X_8 \end{bmatrix} = \frac{[\hat{H}]^*}{\|\hat{h}\|} [Y] \\
&= \frac{[\hat{H}]^*}{\|\hat{h}\|} \{[H][s] + [N]\} \\
&= \frac{|h_1|^2 + |h_2|^2}{\|h\|} [s] + \frac{[\hat{H}]^*}{\|\hat{h}\|} [N] \\
&= \|h\| [s] + \frac{[H]^*}{\|h\|} [N] \\
&= \|h\| \begin{bmatrix} s_6 \\ s_8 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \\
&= \|h\| \begin{bmatrix} s_6 \\ s_8 \end{bmatrix} + \frac{1}{\|h\|} \begin{bmatrix} h_1^* n_1 + h_2 n_2 \\ h_2 n_1 - h_1 n_2 \end{bmatrix}.
\end{aligned} \tag{79}$$

The above scheme uses two less transmit antennas and two less time slots to transmit the eight symbols but uses four frequency bands. It has a code rate of four since it transmits eight symbols using two time slots. This scheme is useful for high speed data rate communications since it can achieve a code rate of four.

The block diagrams for the transmitters and receiver are illustrated in Figure 17 and 18, respectively [5].

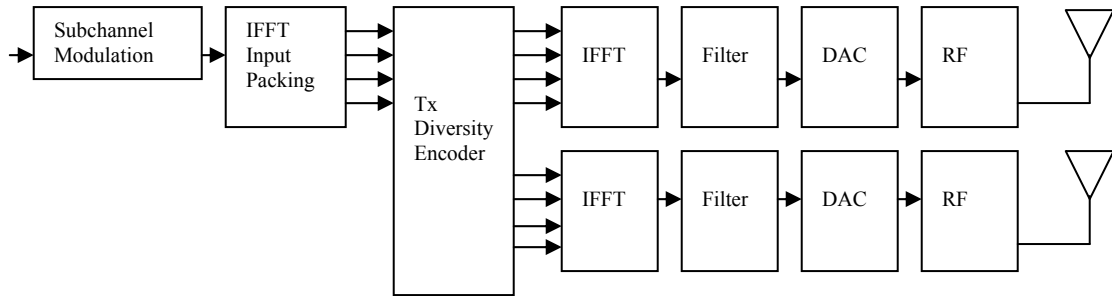


Figure 17. Modulator for matrix  $E$ .



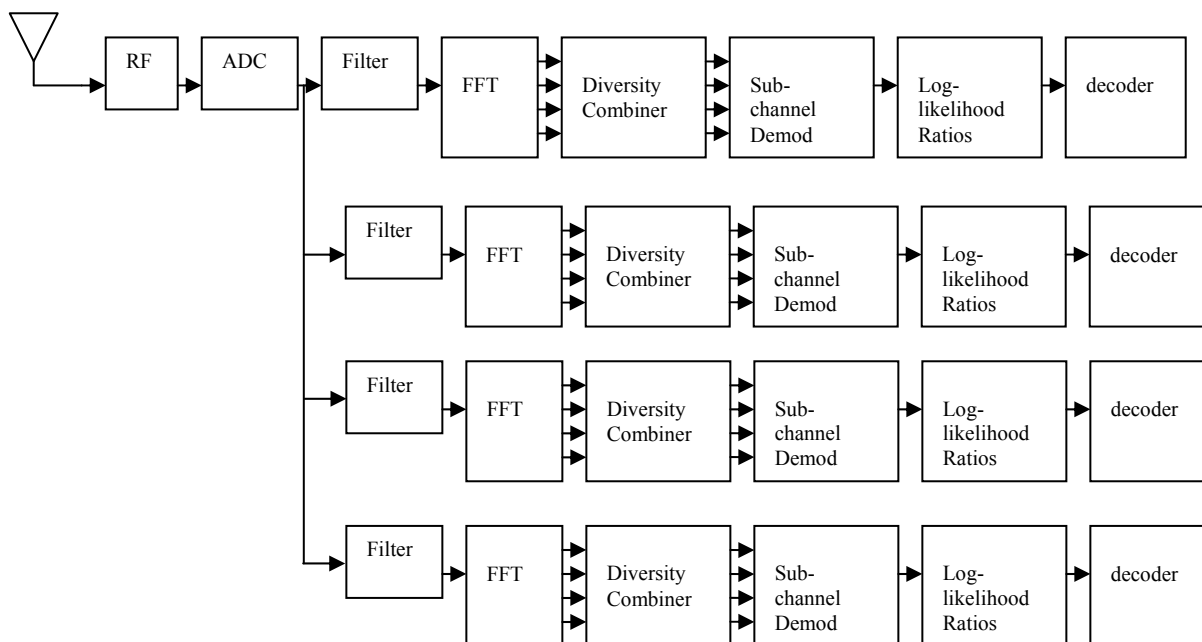


Figure 18. Demodulator for matrix  $E$ .

There are two transmitting antennas on the BS side and one receive antenna on the SS side. In this case, the receiver will have an even higher complexity compare to matrix  $D$  as it receives eight symbols in a time slot from four different frequency bands. To demodulate the received signal, four matched filters are used to separate the signals and process them separately, as seen in Figure 18.

## B. PERFORMANCE EVALUATION

Since the matrix  $B$  is non-orthogonal, simulation is required to evaluate its performance. MATLAB is used for the simulation, and one can use the ML decoding method to perform the simulation for both one and two receiving antennas. To perform the decoding, take an example from [6]. Consider a MIMO system with  $M_T$  transmit antennas and  $M_R$  receive antennas. Assume the information bit stream to be transmitted is encoded into ST codeword of dimension  $M_T \times T$ , where  $T$  is the block length, comprising of data symbols drawn from a unit average energy constellations. We denote the ST

codeword by  $S = [s_1 \ s_2 \ \dots \ s_T]$ , where  $s[k] = [s_1[k] \ \dots \ s_{M_T}[k]]^T$  is the transmitted vector symbol over the  $k$ th symbol period. The channel remains constant over the length of ST codeword. The signal model is

$$y[k] = \sqrt{\frac{E_s}{M_T}} HS[k] + n[k], \quad k = 1, 2, \dots, T, \quad (80)$$

where  $E_s$  is the total average energy available at the transmitter over a symbol period,  $H$  is the  $M_R \times M_T$  channel transfer function and  $n[k]$  is the noise vector. All  $T$  received vector symbols in the codeword may be stacked together so that

$$Y = \sqrt{\frac{E_s}{M_T}} HS + N \quad (81)$$

where  $Y = [y_1 \ y_2 \ \dots \ y_T]$  and  $N = [n_1 \ n_2 \ \dots \ n_T]$  are matrices of size  $M_R \times T$ .

To decode, assume the receiver uses a ML detection criterion based on perfect channel knowledge. The estimated codeword is

$$\begin{aligned} \hat{S} &= \arg \min_s \|Y - HS\|_F^2 \\ &= \arg \min_s \sum_{k=1}^T \left\| y[k] - \sqrt{\frac{E_s}{M_T}} HS[k] \right\|_F^2 \end{aligned} \quad (82)$$

where the minimization is performed over all admissible codewords  $s$ . An error occurs when the receiver mistakes a transmitted codeword for another codeword from the set of possible codewords.

Referring to (57), for a two receive antenna system, we see that the  $H$  matrix for the first two columns of matrix  $B$  is

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{12}^* & -h_{11}^* & h_{14}^* & -h_{13}^* \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{22}^* & -h_{21}^* & h_{24}^* & -h_{23}^* \end{bmatrix}. \quad (83)$$

The first and second columns of the matrix  $B$  shown in Equation (50) are used to transmits only  $s_1, s_2, s_3$  and  $s_4$  (transmit in time slots 1 and 2 only), while the third and fourth columns of the matrix  $B$  are used to transmit  $s_5, s_6, s_7$  and  $s_8$  (transmit in time slots 3 and 4 only). One can separate them and run the simulation for the first and second column. Since they do not transmit in the same time slot, they do not interfere with one another. Thus, the results will be the same. Therefore, the MATLAB simulation for the decoding is based on

$$\text{Minimum Distance} = \min \left\| y - \sqrt{\frac{E_s}{4}} Hs \begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \\ \hat{s}_4 \end{bmatrix} \right\|^2, \quad (84)$$

where  $y$  is the set of possible codewords and  $[\hat{s}_1 \ \hat{s}_2 \ \hat{s}_3 \ \hat{s}_4]^T$  is the received codeword.

Equation (84) is used to compute the minimum distance between the received signal and the constellations points in space by comparing it to every point and calculating the distance between them.

There are twelve matrix variations to perform the simulation. The twelve matrices can be generated from Equation (98) of the 802.16 standard shown in Appendix E. The twelve matrices are defined as  $B_{1-1}$  to  $B_{1-12}$ , and the corresponding  $H$  matrices are defined as  $H_{1-1}$  to  $H_{1-12}$ . Therefore,

$$B_{1-1} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \\ s_3 & -s_4^* \\ s_4 & s_3^* \end{bmatrix}, H_{1-1} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{12}^* & -h_{11}^* & h_{14}^* & -h_{13}^* \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{22}^* & -h_{21}^* & h_{24}^* & -h_{23}^* \end{bmatrix}, \quad (85)$$

$$B_{1-2} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \\ s_4 & s_3^* \\ s_3 & -s_4^* \end{bmatrix}, H_{1-2} = \begin{bmatrix} h_{11} & h_{12} & h_{14} & h_{13} \\ h_{12}^* & -h_{11}^* & h_{13}^* & -h_{14}^* \\ h_{21} & h_{22} & h_{24} & h_{23} \\ h_{22}^* & -h_{21}^* & h_{24}^* & -h_{23}^* \end{bmatrix}, \quad (86)$$

$$B_{1-3} = \begin{bmatrix} s_1 & -s_2^* \\ s_3 & -s_4^* \\ s_2 & s_1^* \\ s_4 & s_3^* \end{bmatrix}, H_{1-3} = \begin{bmatrix} h_{11} & h_{13} & h_{12} & h_{14} \\ h_{13}^* & -h_{11}^* & h_{14}^* & -h_{12}^* \\ h_{21} & h_{23} & h_{22} & h_{24} \\ h_{23}^* & -h_{21}^* & h_{24}^* & -h_{22}^* \end{bmatrix}, \quad (87)$$

$$B_{1-4} = \begin{bmatrix} s_1 & -s_2^* \\ s_4 & s_3^* \\ s_2 & s_1^* \\ s_3 & -s_4^* \end{bmatrix}, H_{1-4} = \begin{bmatrix} h_{11} & h_{14} & h_{12} & h_{13} \\ h_{13}^* & -h_{11}^* & h_{12}^* & -h_{14}^* \\ h_{21} & h_{24} & h_{22} & h_{23} \\ h_{23}^* & -h_{21}^* & h_{22}^* & -h_{24}^* \end{bmatrix}, \quad (88)$$

$$B_{1-5} = \begin{bmatrix} s_1 & -s_2^* \\ s_3 & -s_4^* \\ s_4 & s_3^* \\ s_2 & s_1^* \end{bmatrix}, H_{1-5} = \begin{bmatrix} h_{11} & h_{13} & h_{14} & h_{12} \\ h_{14}^* & -h_{11}^* & h_{13}^* & -h_{12}^* \\ h_{21} & h_{23} & h_{24} & h_{22} \\ h_{24}^* & -h_{21}^* & h_{23}^* & -h_{22}^* \end{bmatrix}, \quad (89)$$

$$B_{1-6} = \begin{bmatrix} s_1 & -s_2^* \\ s_4 & s_3^* \\ s_3 & -s_4^* \\ s_2 & s_1^* \end{bmatrix}, H_{1-6} = \begin{bmatrix} h_{11} & h_{14} & h_{13} & h_{12} \\ h_{14}^* & -h_{11}^* & h_{12}^* & -h_{13}^* \\ h_{21} & h_{24} & h_{23} & h_{22} \\ h_{24}^* & -h_{21}^* & h_{22}^* & -h_{23}^* \end{bmatrix}, \quad (90)$$

$$B_{1-7} = \begin{bmatrix} s_1 & -s_3^* \\ s_3 & s_1^* \\ s_2 & -s_4^* \\ s_4 & s_2^* \end{bmatrix}, H_{1-7} = \begin{bmatrix} h_{11} & h_{13} & h_{12} & h_{14} \\ h_{12}^* & h_{14}^* & -h_{11}^* & -h_{13}^* \\ h_{21} & h_{23} & h_{22} & h_{24} \\ h_{22}^* & h_{24}^* & -h_{21}^* & -h_{23}^* \end{bmatrix}, \quad (91)$$

$$B_{1-8} = \begin{bmatrix} s_1 & -s_3^* \\ s_3 & s_1^* \\ s_4 & s_2^* \\ s_2 & -s_4^* \end{bmatrix}, H_{1-8} = \begin{bmatrix} h_{11} & h_{13} & h_{14} & h_{12} \\ h_{12}^* & h_{13}^* & -h_{11}^* & -h_{14}^* \\ h_{21} & h_{23} & h_{24} & h_{22} \\ h_{22}^* & h_{23}^* & -h_{21}^* & -h_{24}^* \end{bmatrix}, \quad (92)$$

$$B_{1-9} = \begin{bmatrix} s_1 & -s_3^* \\ s_2 & -s_4^* \\ s_3 & s_1^* \\ s_4 & s_2^* \end{bmatrix}, H_{1-9} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{13}^* & h_{14}^* & -h_{11}^* & -h_{12}^* \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{23}^* & h_{24}^* & -h_{21}^* & -h_{22}^* \end{bmatrix}, \quad (93)$$

$$B_{1-10} = \begin{bmatrix} s_1 & -s_3^* \\ s_4 & s_2^* \\ s_3 & s_1^* \\ s_2 & -s_4^* \end{bmatrix}, H_{1-10} = \begin{bmatrix} h_{11} & h_{14} & h_{13} & h_{12} \\ h_{13}^* & h_{12}^* & -h_{11}^* & -h_{14}^* \\ h_{21} & h_{24} & h_{23} & h_{22} \\ h_{23}^* & h_{22}^* & -h_{21}^* & -h_{24}^* \end{bmatrix}, \quad (94)$$

$$B_{1-11} = \begin{bmatrix} s_1 & -s_3^* \\ s_2 & -s_4^* \\ s_4 & s_2^* \\ s_3 & s_1^* \end{bmatrix}, H_{1-11} = \begin{bmatrix} h_{11} & h_{12} & h_{14} & h_{13} \\ h_{14}^* & h_{13}^* & -h_{11}^* & -h_{12}^* \\ h_{21} & h_{22} & h_{24} & h_{23} \\ h_{24}^* & h_{23}^* & -h_{21}^* & -h_{22}^* \end{bmatrix}, \quad (95)$$

and

$$B_{1-12} = \begin{bmatrix} s_1 & -s_3^* \\ s_4 & s_2^* \\ s_2 & -s_4^* \\ s_3 & s_1^* \end{bmatrix}, H_{1-12} = \begin{bmatrix} h_{11} & h_{14} & h_{12} & h_{13} \\ h_{14}^* & h_{12}^* & -h_{11}^* & -h_{13}^* \\ h_{21} & h_{24} & h_{22} & h_{23} \\ h_{24}^* & h_{22}^* & -h_{21}^* & -h_{23}^* \end{bmatrix}. \quad (96)$$

The corresponding  $H$  matrix is presented beside the matrix  $B$  and is shown from (85) to (96). One can use the twelve matrix variations to perform the simulation. The  $H$  matrix is generated using the same method illustrated in Equation (57). The first four columns of the  $H$  matrix in Equation (57) are used in this case as we demodulate only four symbols  $s_1, s_2, s_3$  and  $s_4$ . The MATLAB code is given in Appendix B. The channel used is a Rayleigh channel, and the noise is white Gaussian noise with 0 dB variance.

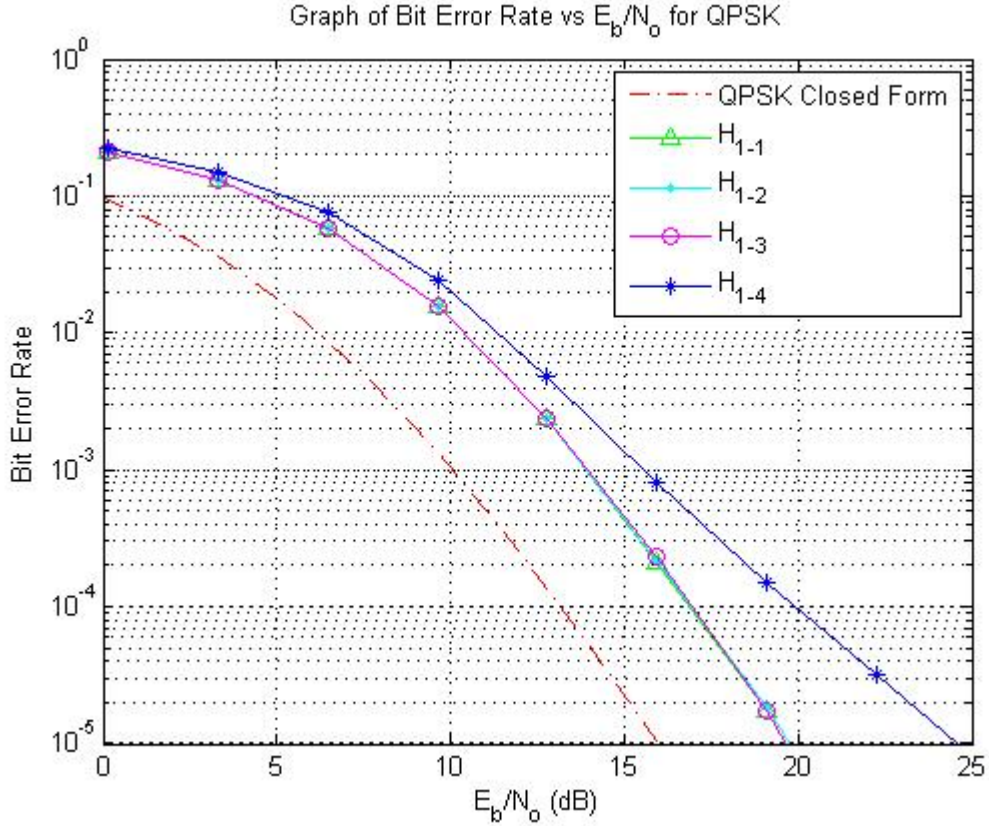


Figure 19. Simulation results for  $H_{1-1}, H_{1-2}, H_{1-3}, H_{1-4}$  and QPSK closed form solution.

From Figure 19, the simulations for matrix  $H_{1-1}, H_{1-2}, H_{1-3}$  are relatively close and have a 4 dB drop in performance when compared to the QPSK closed form solution curve. This is expected since the code rate has increase from one to two.  $H_{1-4}$  has relatively poor performance compared to the rest. We will avoid using this variation of the matrix for transmission.

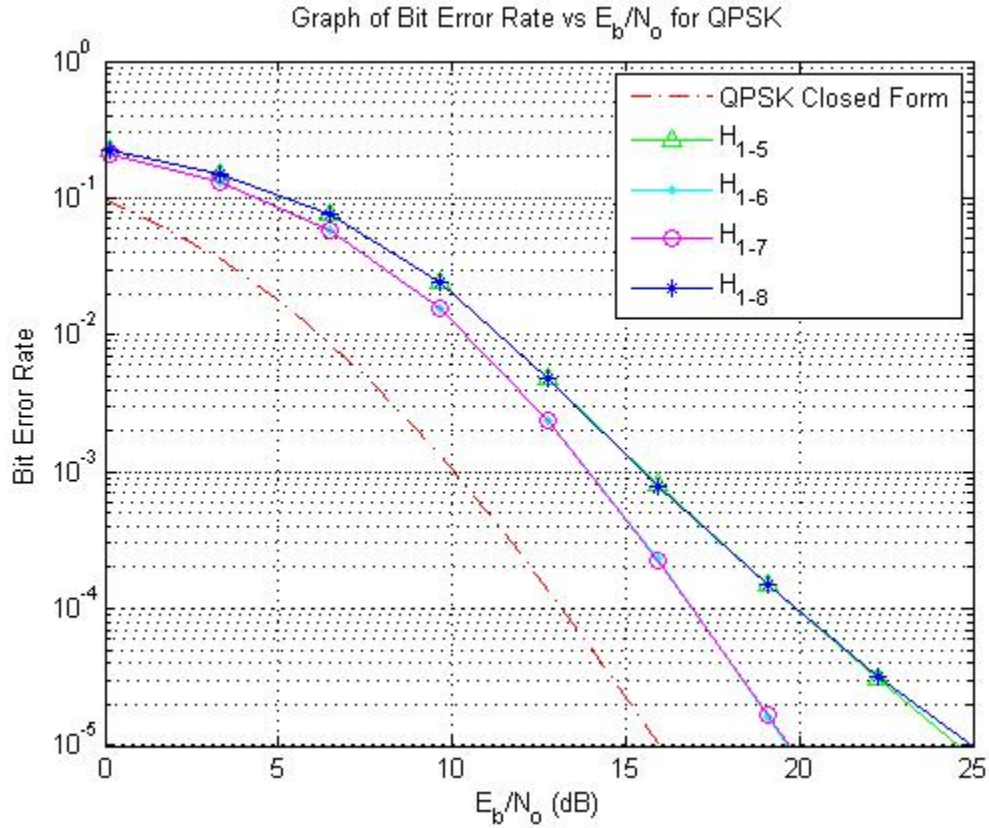


Figure 20. Simulation results for  $H_{1-5}$ ,  $H_{1-6}$ ,  $H_{1-7}$ ,  $H_{1-8}$  and QPSK closed form solution.

In Figure 20, the simulations for  $H_{1-6}$  and  $H_{1-7}$  are relatively close and have a 4 dB drop in performance when compared to the QPSK closed form solution curve.  $H_{1-5}$  and  $H_{1-8}$  have relatively poor performance compared to the rest. We will avoid using these two variations of the matrix for transmission.

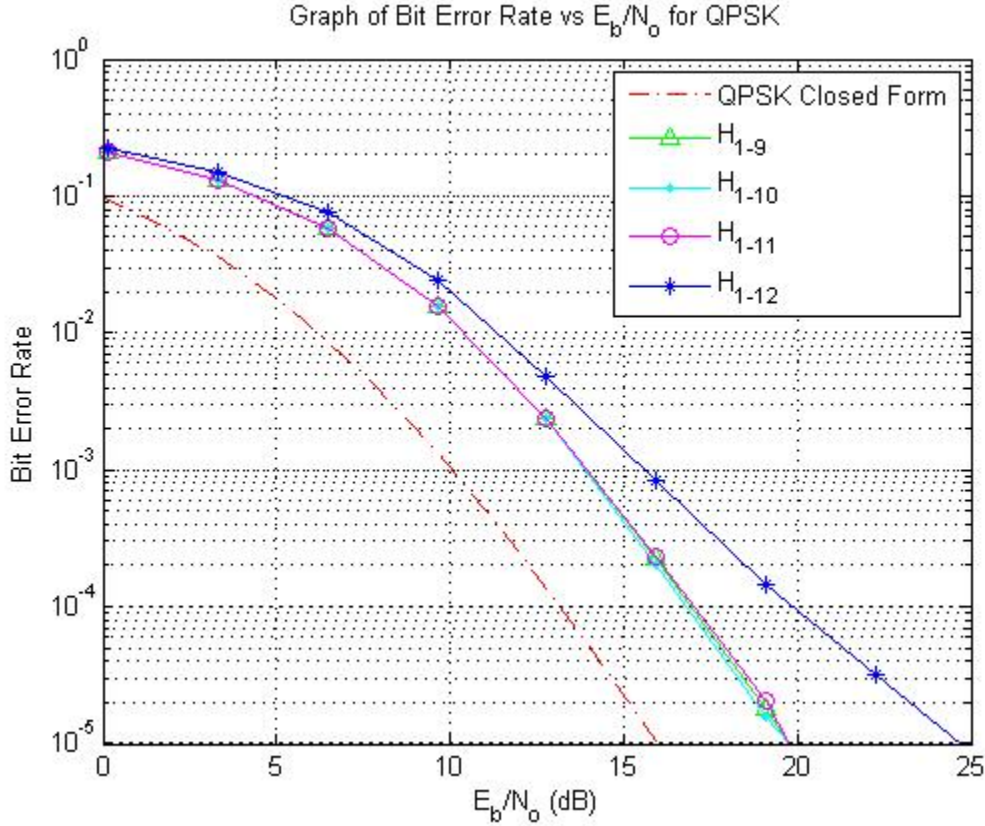


Figure 21. Simulation results for  $H_{1-9}$ ,  $H_{1-10}$ ,  $H_{1-11}$ ,  $H_{1-12}$  and QPSK closed form solution.

In Figure 21, the simulation results for  $H_{1-9}$ ,  $H_{1-10}$  and  $H_{1-11}$  are relatively close and have a 4 dB drop in performance when compared to the QPSK closed form solution curve.  $H_{1-12}$  has relatively poor performance compared to the rest. We will avoid using this variation of the matrix for transmission.

From the simulation results, we can see that  $H_{1-4}$ ,  $H_{1-5}$ ,  $H_{1-8}$  and  $H_{1-12}$  have poor performance compared to  $H_{1-1}$ ,  $H_{1-2}$ ,  $H_{1-3}$ ,  $H_{1-6}$ ,  $H_{1-7}$ ,  $H_{1-9}$ ,  $H_{1-10}$ , and  $H_{1-11}$ . We conclude that we should use these eight matrices for transmission.

Now we should compare the performance using only one receiving antenna. The  $H$  matrix used is from Equation (55). The MATLAB code is attached in Appendix C. The result is illustrated in Figure 22.



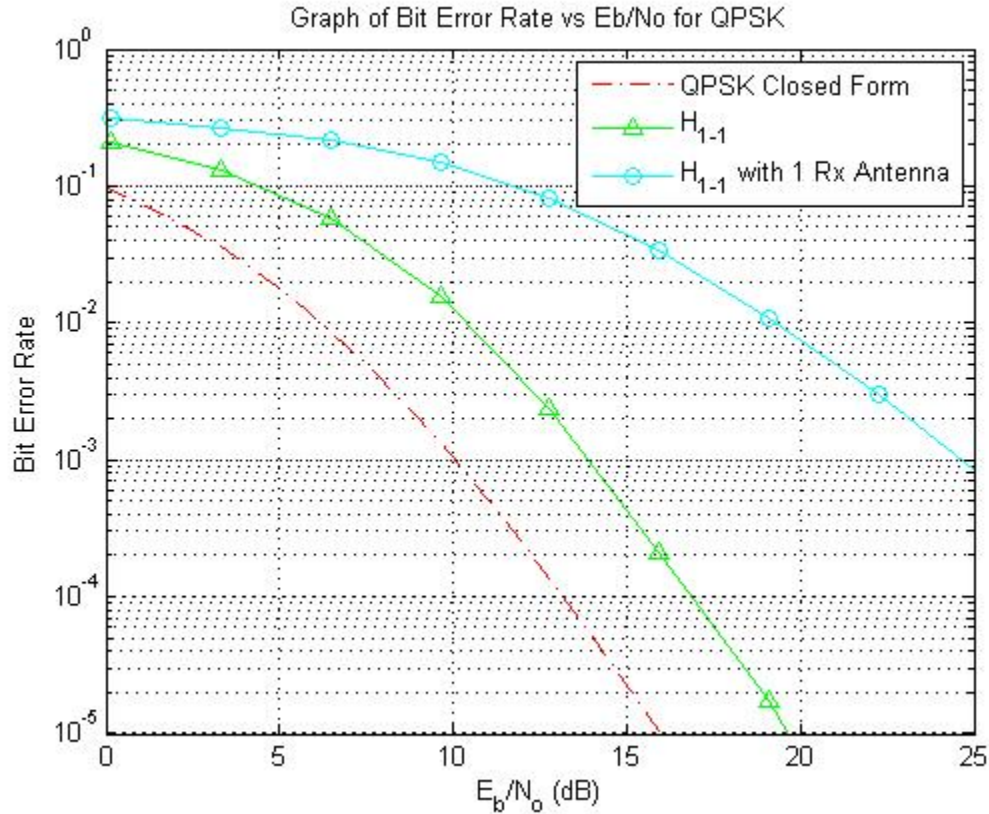


Figure 22. Simulation results for  $H_{1-1}$ ,  $H_{1-1}$  with only one receiving antenna and QPSK closed form solution.

The poor performance when using only one receiving antenna is illustrated in Figure 22. There is an 11 dB drop in performance compared to the design with two receiving antennas.

Next, we evaluate the performance for 16QAM. Due to the large amount of time required to run all twelve  $H$  matrices, only the first four  $H$  matrices were simulated. The MATLAB code is attached in Appendix D. The simulated results are illustrated in Figure 23.

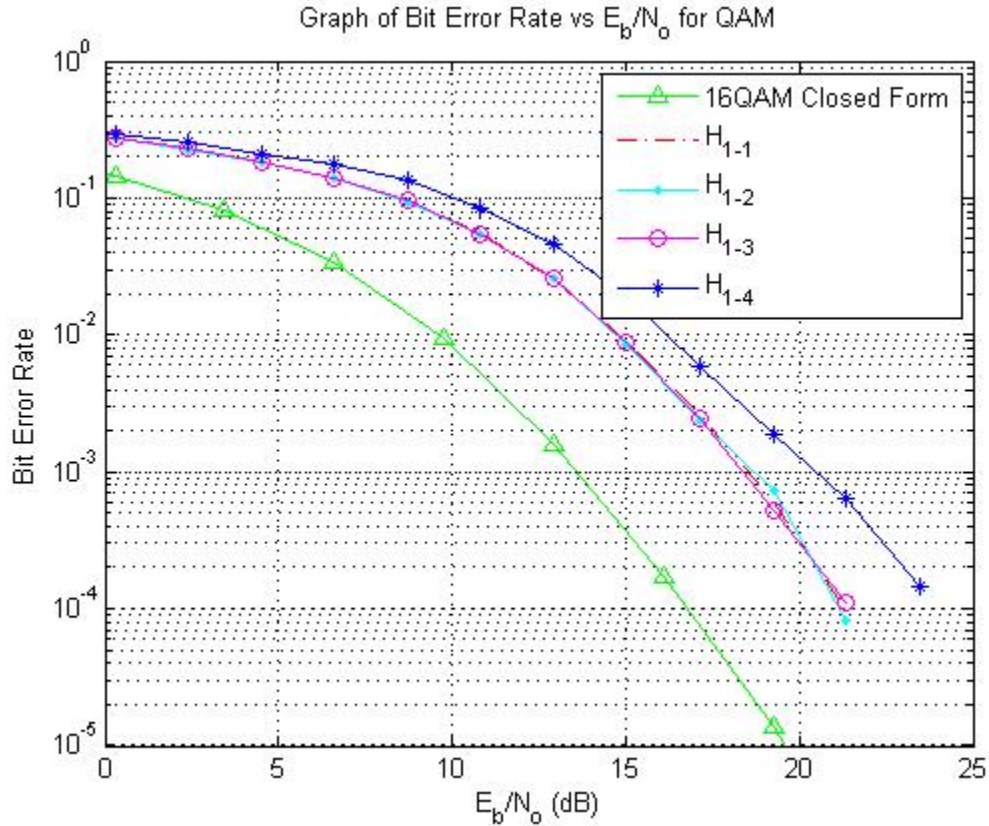


Figure 23. Simulation results for  $H_{1-1}$ ,  $H_{1-2}$ ,  $H_{1-3}$ ,  $H_{1-4}$  and 16QAM closed form solution.

The 16QAM also has a 4 dB drop in performance when compared to the closed form solution, illustrated in Figure 23. Similar to QPSK, the simulation results for  $H_{1-1}$ ,  $H_{1-2}$ ,  $H_{1-3}$  are similar, but  $H_{1-4}$  has poorer performance compared to  $H_{1-1}$ ,  $H_{1-2}$ ,  $H_{1-3}$ . We can now confirm that  $H_{1-4}$  has a poorer performance and should not be used in the transmission process.

### C. CHAPTER SUMMARY

Matrix  $B$  is non-orthogonal. Therefore, programs using MATLAB is required to perform simulations to evaluate the performance. It is clear that with two receiving antennas, one can improve the performance by almost 11 dB. The code rate is two for matrix  $B$  instead of one for matrix  $A$ . Therefore, there is a 4 dB drop in performance when compared to the closed form solution. We discovered that four matrices out of twelve matrices have poorer performance, and these four variations should not be used for transmission. In addition, four frequency bands can be used to transmit the eight symbols at a code rate of 4 with the performance of the closed form solution. The implementation of the modulator and demodulator for matrix  $B$  and its variations were also discussed in this chapter.

## V. CONCLUSION AND FUTURE WORKS

### A. CONCLUSION

This research was focused on STCs and evaluated the use of matrix  $A$  and  $B$  given in the 802.16 standard. The main point of the research was to use MRC to demodulate the transmitted signal.

It has been proven that since the matrix  $A$  and its variations are quasi-orthogonal, the noise is independent; therefore, we can demodulate the transmitted signal using the MRC decision statistics. The performance of the STBC for matrix  $A$  was evaluated using closed form equation.

Using an adaptive system that combines all the variations of the original matrix  $A$  and keeping it quasi-orthogonal, we can achieve superior performance by having the receiver send information to the transmitter in order for the transmitter to make changes to the STC. In addition, one can utilize more frequency bands to transmit the symbols using half the time slots and half the number of antennas used in the standard. The code rate can then be improved from one to four by using four frequency bands.

Use of matrix  $B$ , which is non-orthogonal, results in a 4 dB drop in performance compared to the closed form solution for QPSK and 16QAM. Also, increasing the number of receiving antennas enables a better performance and is necessary, as it provides the receiver with more information to accurately determine the transmitted symbol.

The variations of the matrix  $B$  for the twelve cases determined by the 802.16 standard were simulated. Out of the twelve matrices, four were found to have poorer performance than the other eight and should not be used for transmission.

### B. FUTURE WORK

Due to the large amount of time required to run the MATLAB simulations, we did not simulate the performance of 64QAM or all twelve cases for 16QAM. Due to time

constraints, only four cases for the 16QAM were simulated. More data points need to be included in the MATLAB to generate more accurate results, and 64QAM should be investigated in detailed.

Future researchers can also generate more variations of the  $H$  matrix and evaluate the performance obtained by using these variations. In addition, the MATLAB code can also be modified to perform a simulation when more receiving antennas are used.

## APPENDIX A

```
%This MATLAB Code plots the closed form solution for QPSK and QAM
%This MATLAB code is provided by MAJ Ng Wei Gee

EsN0_lower = 0;
EsN0_upper = 60;
num_EsN0_points = 20;
EsN0_dB = linspace(EsN0_lower,EsN0_upper,num_EsN0_points) ; % multiple
Eb/N0 values

EbN0_dB_QPSK = EsN0_dB - 10*log10(2);
EbN0_dB_QAM16 = EsN0_dB - 10*log10(4);
EbN0_dB_QAM64 = EsN0_dB - 10*log10(6);

EbN0_dB = [EsN0_dB; EbN0_dB_QPSK; EbN0_dB_QAM16; EbN0_dB_QAM64];

% calculate theoretical QAM curves
M=16;
L=4;
iSNR = 3*10.^(EbN0_dB(1,:)/10)/L/2/(M-1);
mu = sqrt(iSNR./(1+iSNR));
sumterm = 0;
for l=0:L-1
    sumterm = sumterm + nchoosek(L-1+l, l)*((1+mu)/2).^l;
end
qam16BerTheory = (4-4/sqrt(M))/log2(M)*((1-mu)/2).^L .* sumterm;
semilogy (EbN0_dB_QAM16, qam16BerTheory, 'y')
hold on

% QPSK
L=4;
iSNR = 10.^(EbN0_dB(1,:)/10)/L/2;
mu = sqrt(iSNR./(1+iSNR));
sumterm = 0;
for l=0:L-1
    sumterm = sumterm + nchoosek(L-1+l, l)*((1+mu)/2).^l;
end
qpskBerTheory = ((1-mu)/2).^L .* sumterm;
semilogy (EbN0_dB_QPSK, qpskBerTheory, 'r')
hold on
```

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX B

```
%This MATLAB Code computes the BER for H1 to H12 matrix for QPSK
%A large portion of this MATLAB code is provided by MAJ Ng Wei Gee

clear
close all

qpsk_map = [ 0.7071 + 0.7071i
            -0.7071 + 0.7071i
             0.7071 - 0.7071i
            -0.7071 - 0.7071i]

bit_diff_map = [ 0 1 2 1 2 3 2 1 ]

N = linspace (1e3,2e3, 20); % number of bits or symbols
EsN0_lower = 0;
EsN0_upper = 60;
num_EsN0_points = 20;
num_points_per_iter = 4;
num_iter=ceil (N/num_points_per_iter/2);

constellation_map = qpsk_map;
c_factor = sqrt(2);
angle = 31.7 / 180 * pi();

% initialise variables
EsN0_dB = linspace(EsN0_lower,EsN0_upper,num_EsN0_points) ; % multiple
Eb/N0 values
num_bit_errors_1 = zeros(1, num_EsN0_points);
num_sym_errors_1 = zeros(1, num_EsN0_points);
num_bit_errors_2 = zeros(1, num_EsN0_points);
num_sym_errors_2 = zeros(1, num_EsN0_points);
num_bit_errors_3 = zeros(1, num_EsN0_points);
num_sym_errors_3 = zeros(1, num_EsN0_points);
num_bit_errors_4 = zeros(1, num_EsN0_points);
num_sym_errors_4 = zeros(1, num_EsN0_points);
num_bit_errors_5 = zeros(1, num_EsN0_points);
num_sym_errors_5 = zeros(1, num_EsN0_points);
num_bit_errors_6 = zeros(1, num_EsN0_points);
num_sym_errors_6 = zeros(1, num_EsN0_points);
num_bit_errors_7 = zeros(1, num_EsN0_points);
num_sym_errors_7 = zeros(1, num_EsN0_points);
num_bit_errors_8 = zeros(1, num_EsN0_points);
num_sym_errors_8 = zeros(1, num_EsN0_points);
num_bit_errors_9 = zeros(1, num_EsN0_points);
num_sym_errors_9 = zeros(1, num_EsN0_points);
num_bit_errors_10 = zeros(1, num_EsN0_points);
num_sym_errors_10 = zeros(1, num_EsN0_points);
num_bit_errors_11 = zeros(1, num_EsN0_points);
num_sym_errors_11 = zeros(1, num_EsN0_points);
num_bit_errors_12 = zeros(1, num_EsN0_points);
```



```

num_sym_errors_12 = zeros(1, num_EsN0_points);

tic
for j_iter = 1:num_EsN0_points
for i_iter= 1:num_iter(j_iter)

    % Transmitter
    symbol_orig = randi([0 3],4,1) ;
    s = constellation_map(symbol_orig+1);

    n = 1/sqrt(2)*(randn(4*2,1) + 1i*randn(4*2,1)); % white
gaussian noise, 0dB variance

    h11 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h12 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h13 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h14 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h21 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h22 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h23 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h24 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel

    % instantaneous noise factor
iNoiseFactor = 2*sqrt(2)*10^(-EsN0_dB(j_iter)/20);

    % Channel and noise Noise addition for QO
H1 = [
        h11 h12 h13 h14
        conj(h12) -conj(h11) conj(h14) -conj(h13)
        h21 h22 h23 h24
        conj(h22) -conj(h21) conj(h24) -conj(h23)];

H2 = [
        h11 h12 h14 h13
        conj(h12) -conj(h11) conj(h13) -conj(h14)
        h21 h22 h24 h23
        conj(h22) -conj(h21) conj(h23) -conj(h24)];

H3 = [
        h11 h13 h12 h14
        conj(h13) -conj(h11) conj(h14) -conj(h12)
        h21 h23 h22 h24
        conj(h23) -conj(h21) conj(h24) -conj(h22)];

H4 = [
        h11 h14 h12 h13
        conj(h13) -conj(h11) conj(h12) -conj(h14)
        h21 h24 h22 h23

```

```

conj(h23)  -conj(h21) conj(h22)  -conj(h24)];

H5 = [
      h11  h13  h14  h12
conj(h14) -conj(h11) conj(h13)  -conj(h12)
h21  h23  h24  h22
conj(h24) -conj(h21) conj(h23)  -conj(h22)];

H6 = [
      h11  h14  h13  h12
conj(h14) -conj(h11) conj(h12)  -conj(h13)
h21  h24  h23  h22
conj(h24) -conj(h21) conj(h22)  -conj(h23)];

H7 = [
      h11  h13  h12  h14
conj(h12) conj(h14) -conj(h11)  -conj(h13)
h21  h23  h22  h24
conj(h22) conj(h24) -conj(h21)  -conj(h23)];

H8 = [
      h11  h13  h14  h12
conj(h12) conj(h13) -conj(h11)  -conj(h14)
h21  h23  h24  h22
conj(h22) conj(h23) -conj(h21)  -conj(h24)];

H9 = [
      h11  h12  h13  h14
conj(h13) conj(h14) -conj(h11)  -conj(h12)
h21  h22  h23  h24
conj(h23) conj(h24) -conj(h21)  -conj(h22)];

H10 = [
      h11  h14  h13  h12
conj(h13) conj(h12) -conj(h11)  -conj(h14)
h21  h24  h23  h22
conj(h23) conj(h22) -conj(h21)  -conj(h24)];

H11 = [
      h11  h12  h14  h13
conj(h14) conj(h13) -conj(h11)  -conj(h12)
h21  h22  h24  h23
conj(h24) conj(h23) -conj(h21)  -conj(h22)];

H12 = [
      h11  h14  h12  h13
conj(h14) conj(h12) -conj(h11)  -conj(h13)
h21  h24  h22  h23
conj(h24) conj(h22) -conj(h21)  -conj(h23)];

y1 = H1*s + iNoiseFactor*n(1:num_points_per_iter);
y2 = H2*s + iNoiseFactor*n(1:num_points_per_iter);
y3 = H3*s + iNoiseFactor*n(1:num_points_per_iter);
y4 = H4*s + iNoiseFactor*n(1:num_points_per_iter);
y5 = H5*s + iNoiseFactor*n(1:num_points_per_iter);
y6 = H6*s + iNoiseFactor*n(1:num_points_per_iter);
y7 = H7*s + iNoiseFactor*n(1:num_points_per_iter);
y8 = H8*s + iNoiseFactor*n(1:num_points_per_iter);
y9 = H9*s + iNoiseFactor*n(1:num_points_per_iter);

```

```

y10 = H10*s + iNoiseFactor*n(1:num_points_per_iter);
y11 = H11*s + iNoiseFactor*n(1:num_points_per_iter);
y12 = H12*s + iNoiseFactor*n(1:num_points_per_iter);

min_dist1=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter1=[s1 s2 s3 s4]';
                s_iterate1= constellation_map(s_iter1+1);
                dist1 = sum(abs(y1 - H1*s_iterate1).^2);
                if dist1<min_dist1
                    min_dist1=dist1;
                    current_best_shat1 = s_iter1;
                end
            end
        end
    end
end

min_dist2=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter2=[s1 s2 s3 s4]';
                s_iterate2= constellation_map(s_iter2+1);
                dist2 = sum(abs(y2 - H2*s_iterate2).^2);
                if dist2<min_dist2
                    min_dist2=dist2;
                    current_best_shat2 = s_iter2;
                end
            end
        end
    end
end

min_dist3=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter3=[s1 s2 s3 s4]';
                s_iterate3= constellation_map(s_iter3+1);
                dist3 = sum(abs(y3 - H3*s_iterate3).^2);
                if dist3<min_dist3
                    min_dist3=dist3;
                    current_best_shat3 = s_iter3;
                end
            end
        end
    end
end

```

```

min_dist4=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter4=[s1 s2 s3 s4]';
                s_iterate4= constellation_map(s_iter4+1);
                dist4 = sum(abs(y4 - H4*s_iterate4).^2);
                if dist4<min_dist4
                    min_dist4=dist4;
                    current_best_shat4 = s_iter4;
                end
            end
        end
    end
end

min_dist5=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter5=[s1 s2 s3 s4]';
                s_iterate5= constellation_map(s_iter5+1);
                dist5 = sum(abs(y5 - H5*s_iterate5).^2);
                if dist5<min_dist5
                    min_dist5=dist5;
                    current_best_shat5 = s_iter5;
                end
            end
        end
    end
end

min_dist6=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter6=[s1 s2 s3 s4]';
                s_iterate6= constellation_map(s_iter6+1);
                dist6 = sum(abs(y6 - H6*s_iterate6).^2);
                if dist6<min_dist6
                    min_dist6=dist6;
                    current_best_shat6 = s_iter6;
                end
            end
        end
    end
end

```

```

min_dist7=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter7=[s1 s2 s3 s4]';
                s_iterate7= constellation_map(s_iter7+1);
                dist7 = sum(abs(y7 - H7*s_iterate7).^2);
                if dist7<min_dist7
                    min_dist7=dist7;
                    current_best_shat7 = s_iter7;
                end
            end
        end
    end
end
end

```

```

min_dist8=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter8=[s1 s2 s3 s4]';
                s_iterate8= constellation_map(s_iter8+1);
                dist8 = sum(abs(y8 - H8*s_iterate8).^2);
                if dist8<min_dist8
                    min_dist8=dist8;
                    current_best_shat8 = s_iter8;
                end
            end
        end
    end
end
end

```

```

min_dist9=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter9=[s1 s2 s3 s4]';
                s_iterate9= constellation_map(s_iter9+1);
                dist9 = sum(abs(y9 - H9*s_iterate9).^2);
                if dist9<min_dist9
                    min_dist9=dist9;
                    current_best_shat9 = s_iter9;
                end
            end
        end
    end
end
end

```

```

min_dist10=1e3;

```

```

for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter10=[s1 s2 s3 s4]';
                s_iterate10= constellation_map(s_iter10+1);
                dist10 = sum(abs(y10 - H10*s_iterate10).^2);
                if dist10<min_dist10
                    min_dist10=dist10;
                    current_best_shat10 = s_iter10;
                end
            end
        end
    end
end

min_dist11=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter11=[s1 s2 s3 s4]';
                s_iterate11= constellation_map(s_iter11+1);
                dist11 = sum(abs(y11 - H11*s_iterate11).^2);
                if dist11<min_dist11
                    min_dist11=dist11;
                    current_best_shat11 = s_iter11;
                end
            end
        end
    end
end

min_dist12=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter12=[s1 s2 s3 s4]';
                s_iterate12= constellation_map(s_iter12+1);
                dist12 = sum(abs(y12 - H12*s_iterate12).^2);
                if dist12<min_dist12
                    min_dist12=dist12;
                    current_best_shat12 = s_iter12;
                end
            end
        end
    end
end

% count bit errors

```

```

        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat1+1)) * c_factor/2;
        num_bit_errors_1(j_iter) = num_bit_errors_1(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat2+1)) * c_factor/2;
        num_bit_errors_2(j_iter) = num_bit_errors_2(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat3+1)) * c_factor/2;
        num_bit_errors_3(j_iter) = num_bit_errors_3(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat4+1)) * c_factor/2;
        num_bit_errors_4(j_iter) = num_bit_errors_4(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat5+1)) * c_factor/2;
        num_bit_errors_5(j_iter) = num_bit_errors_5(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat6+1)) * c_factor/2;
        num_bit_errors_6(j_iter) = num_bit_errors_6(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat7+1)) * c_factor/2;
        num_bit_errors_7(j_iter) = num_bit_errors_7(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat8+1)) * c_factor/2;
        num_bit_errors_8(j_iter) = num_bit_errors_8(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

```

```

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat9+1)) * c_factor/2;
        num_bit_errors_9(j_iter) = num_bit_errors_9(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat10+1)) * c_factor/2;
        num_bit_errors_10(j_iter) = num_bit_errors_10(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat11+1)) * c_factor/2;
        num_bit_errors_11(j_iter) = num_bit_errors_11(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

        % count bit errors
        dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat12+1)) * c_factor/2;
        num_bit_errors_12(j_iter) = num_bit_errors_12(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

end
end
N = num_iter * num_points_per_iter;

simBer_1 = num_bit_errors_1./N/2; % simulated ber
simBer_2 = num_bit_errors_2./N/2; % simulated ber
simBer_3 = num_bit_errors_3./N/2; % simulated ber
simBer_4 = num_bit_errors_4./N/2; % simulated ber
simBer_5 = num_bit_errors_5./N/2; % simulated ber
simBer_6 = num_bit_errors_6./N/2; % simulated ber
simBer_7 = num_bit_errors_7./N/2; % simulated ber
simBer_8 = num_bit_errors_8./N/2; % simulated ber
simBer_9 = num_bit_errors_9./N/2; % simulated ber
simBer_10 = num_bit_errors_10./N/2; % simulated ber
simBer_11 = num_bit_errors_11./N/2; % simulated ber
simBer_12 = num_bit_errors_12./N/2; % simulated ber

```



THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX C

```
%This MATLAB Code computes using H1 with only 1 receiving antenna for
%QPSK
%A large portion of this MATLAB code is provided by MAJ Ng Wei Gee

clear
close all

qpsk_map = [ 0.7071 + 0.7071i
            -0.7071 + 0.7071i
             0.7071 - 0.7071i
            -0.7071 - 0.7071i]

bit_diff_map = [ 0 1 2 1 2 3 2 1 ]

N = linspace (1e6,2e7, 20); % number of bits or symbols
EsN0_lower = 0;
EsN0_upper = 60;
num_EsN0_points = 20;
num_points_per_iter = 4;
num_iter=ceil (N/num_points_per_iter/2);

constellation_map = qpsk_map;
c_factor = sqrt(2);
angle = 31.7 / 180 * pi();

% initialise variables
EsN0_dB = linspace(EsN0_lower,EsN0_upper,num_EsN0_points) ; % multiple
Eb/N0 values
num_bit_errors_1 = zeros(1, num_EsN0_points);
num_sym_errors_1 = zeros(1, num_EsN0_points);

tic
for j_iter = 1:num_EsN0_points
for i_iter= 1:num_iter(j_iter)

    % Transmitter
    symbol_orig = randi([0 3],4,1) ;
    s = constellation_map(symbol_orig+1);

    n = 1/sqrt(2)*(randn(4*2,1) + 1i*randn(4*2,1)); % white
gaussian noise, 0dB variance

    h11 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h12 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
    h13 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
```

```

        h14 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
        h21 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
        h22 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
        h23 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
        h24 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel

% instantaneous noise factor
iNoiseFactor = 2*sqrt(2)*10^(-EsN0_dB(j_iter)/20);

% Channel and noise Noise addition for QO
H1 = [
        h11 h12 h13 h14
        conj(h12) -conj(h11) conj(h14) -conj(h13)
        0 0 0 0
        0 0 0 0];

y1 = H1*s + iNoiseFactor*n(1:num_points_per_iter);

min_dist1=1e3;
for s1 = 0:3
    for s2 = 0:3
        for s3 = 0:3
            for s4 = 0:3
                s_iter1=[s1 s2 s3 s4]';
                s_iteratel= constellation_map(s_iter1+1);
                dist1 = sum(abs(y1 - H1*s_iteratel).^2);
                if dist1<min_dist1
                    min_dist1=dist1;
                    current_best_shat1 = s_iter1;
                end
            end
        end
    end
end

% count bit errors
dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat1+1)) * c_factor/2;
num_bit_errors_1(j_iter) = num_bit_errors_1(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

end
end
N = num_iter * num_points_per_iter;

simBer_1 = num_bit_errors_1./N/2; % simulated ber

```

## APPENDIX D

```
%This MATLAB Code computes using H1 to H4 with 2 receiving antenna for  
%16QAM, due to the large amount of time needed for this code to run,  
%the H matrix is run separately  
%A large portion of this MATLAB code is provided by MAJ Ng Wei Gee
```

```
clear  
close all
```

```
qam16_map = [0.316227766016838 + 0.316227766016838i  
0.316227766016838 + 0.948683298050514i  
0.316227766016838 - 0.316227766016838i  
0.316227766016838 - 0.948683298050514i  
0.948683298050514 + 0.316227766016838i  
0.948683298050514 + 0.948683298050514i  
0.948683298050514 - 0.316227766016838i  
0.948683298050514 - 0.948683298050514i  
-0.316227766016838 + 0.316227766016838i  
-0.316227766016838 + 0.948683298050514i  
-0.316227766016838 - 0.316227766016838i  
-0.316227766016838 - 0.948683298050514i  
-0.948683298050514 + 0.316227766016838i  
-0.948683298050514 + 0.948683298050514i  
-0.948683298050514 - 0.316227766016838i  
-0.948683298050514 - 0.948683298050514i]
```

```
bit_diff_map = [ 0 1 2 1 2 3 2 1 ]
```

```
N = linspace (1e4,3e5, 20); % number of bits or symbols  
m=4  
EsN0_lower = 0;  
EsN0_upper = 40;  
num_EsN0_points = 20;  
num_points_per_iter = 4;  
num_iter=ceil (N/num_points_per_iter/m);
```

```
constellation_map = qam16_map;  
c_factor = sqrt(10);  
angle = 0.5*atan(0.5);
```

```
% initialise variables
```

```
EsN0_dB = linspace(EsN0_lower,EsN0_upper,num_EsN0_points) ; % multiple  
Eb/N0 values  
num_bit_errors = zeros(1, num_EsN0_points);  
num_sym_errors = zeros(1, num_EsN0_points);
```

```
tic  
for j_iter = 1:num_EsN0_points  
for i_iter= 1:num_iter(j_iter)
```

```

% Transmitter
symbol_orig = randi([0 15],4,1) ;
s = constellation_map(symbol_orig+1);

n = 1/sqrt(2)*(randn(4*2,1) + 1i*randn(4*2,1)); % white
gaussian noise, 0dB variance

channel
h11 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h12 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h13 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h14 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h21 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h22 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h23 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel
h24 = 1/sqrt(2)*(randn(1,1) + 1i*randn(1,1)); % Rayleigh
channel

% instantaneous noise factor
iNoiseFactor = 2*sqrt(2)*10^(-EsN0_dB(j_iter)/20);

% Channel for H1
H = [
      h11 h12 h13 h14
      -conj(h12) conj(h11) -conj(h14) conj(h13)
h21 h22 h23 h24
      -conj(h22) conj(h21) -conj(h24) conj(h23)];

%
% Channel for H2
%
% H = [
%       h11 h12 h14 h13
%       conj(h12) -conj(h11) conj(h13) -conj(h14)
%       h21 h22 h24 h23
%       conj(h22) -conj(h21) conj(h23) -conj(h24)];
%
% Channel for H3
%
% H = [
%       h11 h13 h12 h14
%       conj(h13) -conj(h11) conj(h14) -conj(h12)
%       h21 h23 h22 h24
%       conj(h23) -conj(h21) conj(h24) -conj(h22)];
%
% Channel for H4
%
% H = [
%       h11 h14 h12 h13
%       conj(h13) -conj(h11) conj(h12) -conj(h14)
%       h21 h24 h22 h23
%       conj(h23) -conj(h21) conj(h22) -conj(h24)];

y = H*s + iNoiseFactor*n(1:num_points_per_iter);

```

```

min_dist=1e3;
for s1 = 0:15
    for s2 = 0:15
        for s3 = 0:15
            for s4 = 0:15
                s_iter=[s1 s2 s3 s4]';
                s_iterate= constellation_map(s_iter+1);
                dist = sum(abs(y - H*s_iterate).^2);
                if dist<min_dist
                    min_dist=dist;
                    current_best_shat = s_iter;
                end
            end
        end
    end
end
% count bit errors
dummy = (constellation_map(symbol_orig+1)-
constellation_map(current_best_shat+1)) * c_factor/2;
num_bit_errors(j_iter) = num_bit_errors(j_iter) +
sum(bit_diff_map(round(abs(real(dummy)))+1)) +
sum(bit_diff_map(round(abs(imag(dummy)))+1));

end
end
N = num_iter * num_points_per_iter;

simBer = num_bit_errors./N/m; % simulated ber

```

THIS PAGE INTENTIONALLY LEFT BLANK

## APPENDIX E

$$A_1 = \begin{bmatrix} S_1 & -S^*_2 & 0 & 0 \\ S_2 & S^*_1 & 0 & 0 \\ 0 & 0 & S_3 & -S^*_4 \\ 0 & 0 & S_4 & S^*_3 \end{bmatrix}, A_2 = \begin{bmatrix} S_1 & -S^*_2 & 0 & 0 \\ 0 & 0 & S_3 & -S^*_4 \\ S_2 & S^*_1 & 0 & 0 \\ 0 & 0 & S_4 & S^*_3 \end{bmatrix}, A_3 = \begin{bmatrix} S_1 & -S^*_2 & 0 & 0 \\ 0 & 0 & S_3 & -S^*_4 \\ 0 & 0 & S_4 & S^*_3 \\ S_2 & S^*_1 & 0 & 0 \end{bmatrix} \quad (97)$$

The mapping of subscript  $k$  to determine the Matrix  $A_k$  is given by the following formula:

$$k = \text{mod}(\text{floor}(\text{logical\_data\_subcarrier\_number\_for\_first\_tone\_of\_code} - 1)/2), 3) + 1$$

where

$$\text{logical\_data\_subcarrier\_number\_for\_first\_tone\_of\_code} = 1, 2, 3, \dots, \text{Total \# of data subcarriers.}$$

The permuted matrix- $B$  (over two OFDMA symbols and two subcarriers where the first two columns refer to the first subcarrier over two OFDMA symbols and the last two columns refer to the second subcarrier over two OFDMA symbols) for 4-Tx-Rate 2 is given in six permuted matrices as shown in Equation (98).

$$B_1 = \begin{bmatrix} S_1 & -S^*_2 & S_3 & -S^*_4 \\ S_2 & S^*_1 & S_5 & -S^*_6 \\ S_3 & -S^*_4 & S_7 & S^*_5 \\ S_4 & S^*_3 & S_9 & S^*_8 \end{bmatrix}, B_2 = \begin{bmatrix} S_1 & -S^*_2 & S_3 & -S^*_4 \\ S_2 & S^*_1 & S_5 & -S^*_6 \\ S_4 & S^*_3 & S_8 & S^*_7 \\ S_3 & -S^*_4 & S_7 & S^*_5 \end{bmatrix}, B_3 = \begin{bmatrix} S_1 & -S^*_2 & S_3 & -S^*_4 \\ S_3 & -S^*_4 & S_7 & S^*_5 \\ S_2 & S^*_1 & S_5 & -S^*_6 \\ S_4 & S^*_3 & S_8 & S^*_7 \end{bmatrix} \quad (98)$$

$$B_4 = \begin{bmatrix} S_1 & -S^*_2 & S_3 & -S^*_4 \\ S_4 & S^*_3 & S_8 & S^*_7 \\ S_2 & S^*_1 & S_5 & -S^*_6 \\ S_3 & -S^*_4 & S_7 & S^*_5 \end{bmatrix}, B_5 = \begin{bmatrix} S_1 & -S^*_2 & S_3 & -S^*_4 \\ S_3 & -S^*_4 & S_7 & S^*_5 \\ S_4 & S^*_3 & S_8 & S^*_7 \\ S_2 & S^*_1 & S_5 & -S^*_6 \end{bmatrix}, B_6 = \begin{bmatrix} S_1 & -S^*_2 & S_3 & -S^*_4 \\ S_4 & S^*_3 & S_8 & S^*_7 \\ S_3 & -S^*_4 & S_7 & S^*_5 \\ S_2 & S^*_1 & S_5 & -S^*_6 \end{bmatrix}$$

The mapping of subscript  $k$  to determine the Matrix  $B_k$  is given by the following formula:

$$k = \text{mod}(\text{floor}(\text{logical\_data\_subcarrier\_number\_for\_first\_tone\_of\_code} - 1)/2), 6) + 1$$

where

$$\text{logical\_data\_subcarrier\_number\_for\_first\_tone\_of\_code} = 1, 2, 3, \dots, \text{Total \# of data subcarriers.}$$

### 8.4.8.3.5.1 Enhanced 4-Tx Matrix A with antenna grouping

For 4-Tx antenna BS, transmission Matrix  $A$  in 8.4.8.3.5 may be employed with adaptive antenna grouping, which is fed back from MS.

When MS reports 0b101110 on its CQICH, then BS shall group antenna 0 and 1 for the first subcarrier and antenna 2 and 3 for the second subcarrier. In matrix form, it shall be read as shown in Equation (99).



THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF REFERENCES

- [1] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block code from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, Oct. 1999.
- [2] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Sel. Areas Comm.*, vol. 16, Oct 1998.
- [3] B. Vucetic and J. Yuan, *Space-Time Coding*, Wiley, 2003.
- [4] T. Ha, *Theory and Design of Digital Communications Systems*, Cambridge University Press, 2011.
- [5] IEEE Standard For Local and metropolitan area networks, *Part 16: Air Interface for Broadband Wireless Access Systems*, *IEEE Std 802.16*, 2009.
- [6] R. Prasad, *OFDM for Wireless Communications Systems*, Artech House, 2004.
- [7] A. Paulraj, R. Nabar and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, 2003.
- [8] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj and H. Vincent Poor, *MIMO Wireless Communications*, Cambridge University Press, 2007.
- [9] H. Jafarkhani, "A quasi orthogonal space-time block codes," *IEEE Trans. Comm.*, vol. 49, Jan 2001.
- [10] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inf. Theory*, vol. 44, Mar. 1998.
- [11] B. Lu and X. Wang, "Space-Time code design in OFDM systems," in *Proc. IEEE GLOBECOM'00*, Nov. 2000.
- [12] Z. Liu, G. B. Giannakis, S. Barbarossa, and A. Scaglione, "Tramitter-antennae spece-time block coding for generalized OFDM in the presence of unknown multipath," *IEEE J. Sel. Areas Comm.*, vol. 19, Jul. 2001.
- [13] J. Yuan, B. Vucetic, Z. Chen, and W. Firmanto, "Performance of Space-Time coding on fading channel," in *Proc. IEEE ISIT'01*, Jun. 2001.
- [14] Y. Gong and K. B. Letaief, "Space-frequency-time coded OFDM for broadband wireless communications," in *IEEE Global Telecommun. Conf*, vol. 1, Nov. 2001.

THIS PAGE INTENTIONALLY LEFT BLANK

## **INITIAL DISTRIBUTION LIST**

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California