2004

# Simulation studies of some new algebraic space-time codes

Geetali Bhatia
*Lehigh University*

Bhatia, Geetali M.

# Simulation Studies of Some New Algebraic Space-Time Codes

September 2004

# SIMULATION STUDIES OF SOME NEW ALGEBRAIC SPACE-TIME CODES

by

Geetali M. Bhatia

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Electrical Engineering

**Lehigh University**
**2004**

This thesis is accepted in partial fulfillment of the requirements for the degree of Master of Science.

AUGUST 12, 2004

(Date)

Rick Blum (Advisor)

Meghanad D. Wagh (Co-Advisor)

Donald Bolle, Chair

# Acknowledgements

I would like to thank Professor R. Blum for introducing me to the exciting field of MIMO systems and Space-Time Coding. I also thank him for suggesting the problem and also supporting me in summer. I also want to thank Professor Wagh for the discussions and help in formating the thesis.

I would like to take this opportunity to thank my parents, Dr. Manmohan Bhatia and Daisy Bhatia, and my brother, Haman, for all the encouragement and help they have provided me. They have been my solid rock of support during all my high as well as low periods.

Finally, I would like to thank my many friends at Lehigh, specially, Navneet, Prakash, Abhishek, Ashutosh, Venky and Sumit for helping make my stay and work here more enjoyable.

# Contents

**Acknowledgements** **iii**

**Abstract** **1**

**1 Performance of Algebraic Space-time Code** **3**

1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.2 Program to calculate the coding gain . . . . . . . . . . . . . . . . . 4

1.3 Number Theory . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

1.4 Unitary Transformation . . . . . . . . . . . . . . . . . . . . . . 8

1.5 System Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

1.6 Simulation results . . . . . . . . . . . . . . . . . . . . . . . . . 9

1.6.1 Numerical results . . . . . . . . . . . . . . . . . . . . . . 12

1.7 Conclusions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

**Bibliography** **15**

**Vita** **16**

# List of Figures

1.1 Plot of BER Vs SNR when $\lambda = \pi/2$ . . . . . . . . . . . . . . . . . . . . . 12
1.2 Plot of BER Vs SNR when $\lambda = \pi/4$ . . . . . . . . . . . . . . . . . . . . . 13
1.3 Plot of BER Vs SNR when $\lambda = \pi/8$ . . . . . . . . . . . . . . . . . . . . . 14

# Abstract

This thesis mainly focuses on obtaining the BER vs SNR plots for a code based on the unitary transformation proposed by Damen et al. They proposed a systematic construction of fully diverse unitary transformations carved from number rings. They concluded that given a multidimensional constellation from a number ring that is optimized for the additive white Gaussian noise channel, one can apply the unitary transformation proposed by them in order to make it suitable for fading channels. In this thesis, one such code is considered and the unitary transformation proposed by them is applied to get the BER vs SNR plots.

We conclude that the BER decreases with an increase in SNR. However, for low SNRs (<50), BER is least for $\lambda = \pi/2$ and highest for $\lambda = \pi/8$. Thus, $\lambda = \pi/2$ is closer to the optimum value of $\lambda$ than $\pi/4$ or $\pi/8$. Actually, it turns out that $\lambda = 0.521\pi$ is the optimum value.

# Chapter 1

# Performance of Algebraic Space-time Code

In 2002, Damen,Tewfik and Belfiore [1] proposed a new Space-Time code based on number theory. This new code was also inspired by previous work by Alamouti [2]. Damen et al. proposed a systematic construction of fully diverse unitary transformations carved from number rings. They conclude that given a multidimensional constellation from a number ring that is optimized for the additive white Gaussian noise channel, one can apply the unitary transformations proposed by them in order to make it suitable for fading channels. In this thesis, one such code is considered and the unitary transformation proposed by them is applied to get the BER vs SNR plots. Similar to their set up, we also assume two transmit and two recieve antennas.

Section 1.1 describes the code used in this study. Some number theoretic notations are explained in Section 1.3. The actual unitary transformation employed is shown in Section 1.4. The Simulation results listed in Section 1.6 are used to characterize bit error rate (BER) as a function of signal to noise ratio (SNR). Numerical results are provided in Subsection 1.6.1.

## 1.1 Introduction

Damen et al., proposed a Space Time code that can be described by the matrix:

$$B_{2,\phi} = \frac{1}{\sqrt{2}} \begin{pmatrix} s_1 + \phi s_2 & \theta.(s_3 + \phi s_4) \\ \theta.(s_3 - \phi s_4) & s_1 - \phi s_2 \end{pmatrix} \tag{1.1}$$

where $\theta^2 = \phi$, $\phi = e^{i\lambda}$ and $\lambda$ is the parameter to be optimized [1]. In Eq. (1.1), $(s_1, s_2)$ are the information bits to be sent. The communications system transmits the entries corresponding to the first column of $B_{2,\phi}$ during a given time slot by sending each element of the column from a given transmit antenna. The second column is sent during the next time slot. The main idea behind Eq. (1.1) was to introduce redundancy in the signal space (or signal space diversity) when the signal constellation is carved from some algebraic lattice. Signal Space Diversity is obtained by applying fully diverse unitary transformations to inputs drawn from lattice or multidimensional digital modulation signals carved from a number ring. The resulting constellations have the property that each point is uniquely determined by any of its components, thus retrieving the whole point if some of its components are lost in a deep fade.

## 1.2 Program to calculate the coding gain

This program is used to find the coding gain in Eq.1.1 for different values of lambdas.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define RANGE  1
```

```
//global data

double s[150][2], phi[3][2], term[2];
const int POINTS = 4*RANGE*RANGE;
double MySine[4], MyCosine[4];

//prototypes

void makedata(void);
void maketerm(int, int);  // index of s, phi-power
void makedata()
{
    int i, j, t=0;
    for (i= -RANGE; i<= RANGE; i++)
        for (j= -RANGE; j<= RANGE; j++)
        {
            s[t][0] = i*i - j*j;
            s[t++][1] = 2*i*j;
        }
}

void maketerm(int i9, int j9)
{

    term[0] = MyCosine[j9]*s[i9][0] - MySine[j9]*s[i9][1];
    term[1] = MyCosine[j9]*s[i9][1] + MySine[j9]*s[i9][0];
}
```

```
int main()
{
    int i1, i2, i3, i4;
    double mygain, temp[4][2], gain=100000;
    float lambda9;
    int answer[4];
    makedata();

    printf("\nPlease input a lambda: ");
    scanf("%f", &lambda9);
    for (i1=0; i1<4; i1++)
    {
        MyCosine[i1] = cos(lambda9*i1);
        MySine[i1] = sin(lambda9*i1);
    }
    makedata();
    for (i1= 0; i1 < POINTS; i1++)
    {
        maketerm(i1, 0);
        temp[0][0] = term[0];
        temp[0][1] = term[1];
        for (i3 = 0; i3 < POINTS; i3++)
        {
            maketerm(i3, 1);
            temp[1][0] = temp[0][0] - term[0];
            temp[1][1] = temp[0][1] - term[1];
            for (i2 = 0; i2 < POINTS; i2++)
            {
                maketerm(i2, 2);
                temp[2][0] = temp[1][0] - term[0];
```

```
                temp[2][1] = temp[1][1] - term[1];
                for (i4 = 0; i4 < POINTS; i4++)
                {
                    maketerm(i4, 3);
                    temp[3][0] = temp[2][0] + term[0];
                    temp[3][1] = temp[2][1] + term[1];
                    mygain = temp[3][0]*temp[3][0] +
                             temp[3][1]*temp[3][1];
                    if ((s[i1][0] != 0) || (s[i1][1] != 0) ||
                        (s[i2][0] != 0) || (s[i2][1] != 0) ||
                        (s[i3][0] != 0) || (s[i3][1] != 0) ||
                        (s[i4][0] != 0) || (s[i4][1] != 0))
                    {
                        if (mygain < gain)
                        {
                            gain = mygain;
                            answer[0] = i1;
                            answer[1] = i2;
                            answer[2] = i3;
                            answer[3] = i4;
                        }
                        if (mygain <= 0.00001)
                            i1=i1;
                    }
                }
            }
        }
    }
    printf("\nGain= %f\n", sqrt(gain)/2);
    return 0;
}
```

## 1.3 Number Theory

This section explains the notations of number theory used in the next section. We use the following notation throughout this Chapter.

1. Q: Field of rational numbers

2. C: Field of complex numbers

3. R: Field of real numbers

4. $w_n$: n-th primitive root of unity ($w_n = e^{\frac{2i\pi}{n}}$)

5. $Z[w_n]$: Ring of algebraic integers generated by cyclotomic polynomial with root $w_n$. Elements of $Z(w_n)$ can be described using the basis $1, w_n, {w_n}^2, \ldots\ldots, {w_n}^{n-1}$,i.e., by an expression

$$\sum_{i=0}^{n-1} x_i w_n^i \qquad \text{where} \quad x_i \in Z,$$

## 1.4 Unitary Transformation

Let R be a number ring and let $\phi_1, \ldots, \phi_m \in$ C with $|\phi_m| = 1$ .

Let $F_m$ be a mxm DFT matrix with entries $f_{kl} = \frac{1}{\sqrt{m}}({w_m}^{(k-1)(l-1)})$.

Then $U = {F_m}^H diag(\phi_1, \ldots\ldots, \phi_m)$ is fully diverse over R if $\phi_1 = 1$ , $\phi_2 = \phi^{\frac{1}{m}}$ .... and $\phi$ is chosen such that $1, \phi, \ldots, \phi^{m-1}$ are algebraically independent over R [3].

Such a choice of $\phi$ includes:

$\phi$ Transcendental: $\phi = e^{i\lambda}$ , $\lambda$ not equal to zero

$\phi$ algebraic of degree $>$ m over F, such that $1, \phi, \ldots, \phi^{m-1}$ is a basis or part of basis of $Q(\phi)$ over F.

## 1.5 System Model

Assume that there are two transmit and two receive antennas connected by a Rayleigh fading channel. Let $(s_1, s_2)$ denote the constellation signals to be transmitted.

The received signal matrix, Y,is given by

$$Y = \sqrt{\rho} H X + e \tag{1.2}$$

where H denotes the channel matrix, X is the transmitted signal set, e is the noise vector and $\rho$ is the signal to noise ratio.

The process of computing the bit error rate (BER) can be described as follows:

1. Calculate Y for given H, X, e and $\rho$ using Eq. (1.2).

2. Get the estimated signal, $\hat{s}$ , from Y by using Maximum likelihood (ML) decoding. This means that the point in the lattice closest to the point corresponding to the received signal is chosen.

3. Compare $\hat{s}$ with the original transmitted signal. If they don't match, it is an error.

4. Determine the BER using the following equation:

$$\begin{aligned} BER &= \frac{\text{No. of errors}}{\text{No. of iterations}} \\ &= \frac{\text{No. of iterations - No. of correct detections}}{\text{No. of iterations}} \end{aligned} \tag{1.3}$$

## 1.6 Simulation results

This section lists the MATLAB code prepared as per the procedure outlined in the previous section. In this program, information bits are generated randomly.

The channel and noise matrices are also selected in a random fashion. Then, using the randomly generated information bits, the transmitted signal is evaluated. The received signal is estimated based on the transmitted signal and channel and noise using Eq. (1.2). The estimated transmitted signal is obtained by using Maximum Likelihood (ML) detection at the output. he estimated signal is then compared with the actual transmitted signal. If they are different, it is an error. So, for a given value of SNR, the program was run for 10,000 iterations. Finally, the Bit Error Rate (BER) was calculated. Then, the plot of BER vs SNR (in dB) was plotted.

The program that does this is given below:

```
function test
counter = 0
lambda = input ('Enter the value of Lambda:  ')
SNR = input ('Enter the value of SNR in db:  ')
Rho = 10 ^ (SNR/10);
for (tt = 1:1:10000)
    S = randint (2,1);
    H = (1/sqrt(2))*[randn (2,2)+i*randn (2,2)];
    e = randn(2,1);
    X = abs (S(1)*S(1) - exp (i*lambda/2)*S(2)*S(2))*S;
    Y = sqrt (rho)*(H*X + e);
    P = (1/(2*pi))*exp((-1/2)*(((Y(1)-[H(1,1) H(1,2)]*X)
        *(Y(1)-[H(1,1) H(1,2)]*X)')+((Y(2)-[H(2,1) H(2,2)]
        *X)*(Y(2)-[H(2,1) H(2,2)]*X)')))
    A1=[0 0];
    A2=[0 1];
    A3=[1 0];
    A4=[1 1];
    G=[A1 A2 A3 A4];
```

```
    Q1=(((Y(1)-[H(1,1)H(1,2)]*A1')*(Y(1)-[H(1,1)H(1,2)]*A1').')+
       ((Y(2)-[H(2,1) H(2,2)]*A1')*(Y(2)-[H(2,1) H(2,2)]*A1').'));
    Q2=(((Y(1)-[H(1,1) H(1,2)]*A2')*(Y(1)-[H(1,1) H(1,2)]*A2').')+
       ((Y(2)-[H(2,1) H(2,2)]*A2')*(Y(2)-[H(2,1) H(2,2)]*A2').'));
    Q3=(((Y(1)-[H(1,1) H(1,2)]*A3')*(Y(1)-[H(1,1) H(1,2)]*A3').')+
       ((Y(2)-[H(2,1) H(2,2)]*A3')*(Y(2)-[H(2,1) H(2,2)]*A3').'));
    Q4=(((Y(1)-[H(1,1) H(1,2)]*A4')*(Y(1)-[H(1,1) H(1,2)]*A4').')+
       ((Y(2)-[H(2,1) H(2,2)]*A4')*(Y(2)-[H(2,1) H(2,2)]*A4').'));
    Q=[Q1,Q2,Q3,Q4];
    M=min(Q)
    if (M==Q1)
        D=A1;
    end
    if (M==Q2)
        D=A2;
    end
    if (M==Q3)
        D=A3;
    end
    if (M==Q4)
        D=A4;
    end
    D
    if (S(1)==D(1))
        counter=counter+1
    elseif (S(2)==D(2))
        counter=counter+1
    end
end
counter
```

### 1.6.1 Numerical results

The simulation results that are obtained after running the above program are shown. There are three plots, one for $\lambda = \pi/2$, one for $\lambda = \pi/4$ and the last one for $\lambda = \pi/8$. Each time, for a given value of SNR, the BER is calculated. The number of iterations for each value of SNR was 10,000.

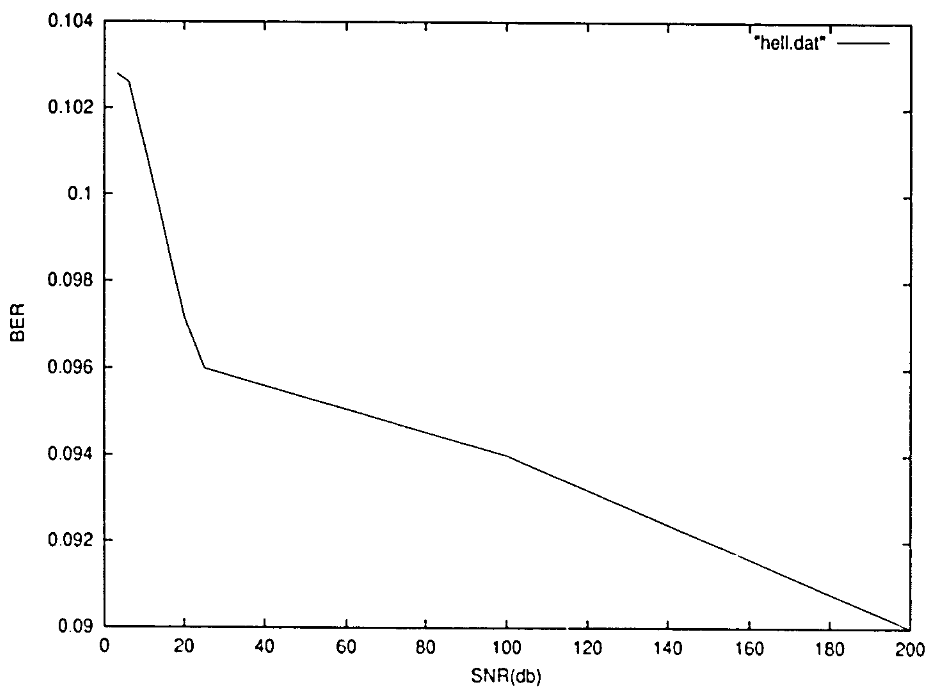


Figure 1.1: Plot of BER Vs SNR when $\lambda = \pi/2$

As can be seen from Figure 1.1, as SNR increases, BER drops in value. However, after a certain value of SNR (about 25 dB), the drop in BER is not as sharp as between 0 and 25 dB. This means that after a certain SNR value is reached, it doesn't help much to increase SNR.

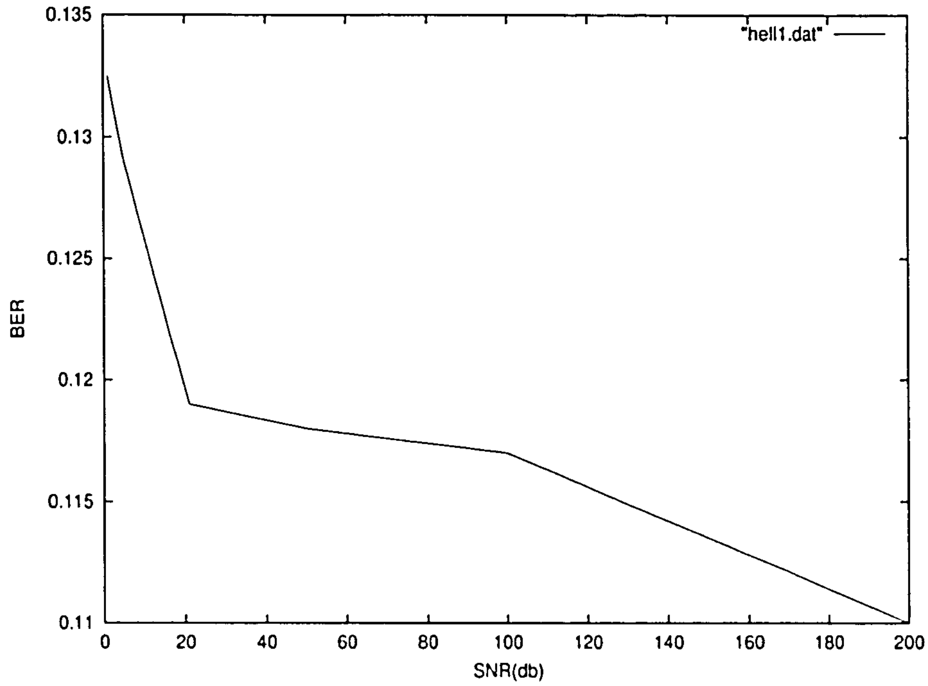


Figure 1.2: Plot of BER Vs SNR when $\lambda = \pi/4$

As is seen from Figure 1.2, again the BER vs SNR curve has a negative slope. However, this time after $SNR = 20dB$, the drop in BER is not as sharp as before.

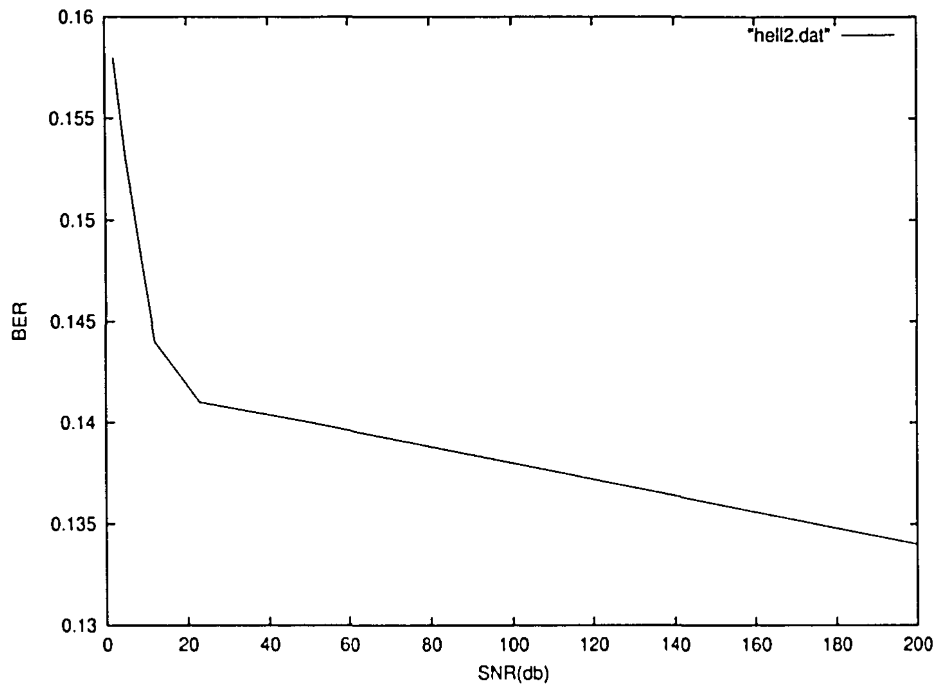


Figure 1.3: Plot of BER Vs SNR when $\lambda = \pi/8$

Finally, in Figure 1.3, at $SNR = 15dB$, the BER ceases to drop off sharply. On comparing these curves with those obtained by using the same code $B_{2,\phi}$ over an additive white Gaussian noise channel (AWGN), we see that in a Rayleigh fading channel with antenna diversity empoyed, the performance of the code is better.

## 1.7 Conclusions

For all three values of $\lambda$ used ($\pi/2, \pi/4, \pi/8$), the BER decreases with an increase in SNR. However, for low SNRs (<50), BER is least for $\lambda = \pi/2$ and highest for $\lambda = \pi/8$. Thus, $\lambda = \pi/2$ is closer to the optimum value of $\lambda$ than $\pi/4$ or $\pi/8$. Actually, it turns out that $\lambda = 0.521\pi$ is the optimum value.

# Bibliography


[1] M. O. Damen, A. Tewfik, and J. C. Belfiore, "A construction of a space-time code based on number theory," *IEEE Trans. Inform. Theory*, vol. 48, March 2002.

[2] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, Oct 1998.

[3] M. O. Damen, H. E. Gamal, and N. C. Beaulieu, "Systematic construction of full diversity algebraic constellations," *IEEE Trans. Inform. Theory*, vol. 49-12, Dec 2003.

# Vita

Geetali Bhatia was born on May 18, 1980 in Mumbai, India. She graduated from Atomic Energy Central School, Mumbai in 1997. She earned her B.S. in Electrical Engineering in 2001 from Victoria Jubilee Technical Institute (VJTI) affiliated with the Mumbai University.

Currently she is a graduate student in the department of Electrical and Computer Engineering of Lehigh University.

END OF TITLE