ABSTRACT

NEW HYBRID AUTOMATIC REPEAT REQUEST (HARQ) SCHEME FOR A 4x4 MIMO SYSTEM, BASED ON THE EXTENDED ALAMOUTI QUASI-ORTHOGONAL SPACE-TIME BLOCK CODING (Q-STBC), IN INVARIANT AND VARIANT FADING CHANNEL

by

Jordi Ferrer Torras

A new Hybrid Automatic Repeat reQuest (HARQ) combining scheme for a 4x4 Multiple Input Multiple Output (MIMO) system in invariant and variant fading channel conditions is proposed and analized. Based on the Extended Alamouti Quasi-orthogonal Space-Time Block Coding (Q-STBC), the use of the so-called Alternative Matrices for transmission, depending on the Channel State Information (CSI) received as feedback, is compared to other existing solutions.

Sign changes and permutations in the retransmission sequences allow reducing the interference while exploiting the spatial diversity to introduce some gain in the signal power. The best transmission order is selected by the Determinant Criterion, which optimizes the SNR in each receiver antenna to minimize the Bit Error Rate (BER) and maximize the throughput.

Studying the performance of a priori different alternatives, both analytically and empirically, several equivalents are found. Finally, the simulation results show that the proposed scheme achieves an improvement for the case of an invariant channel, but not for the time varying model, where the Auto-Regressive of order 1 (AR-1) is chosen for simplicity.

NEW HYBRID AUTOMATIC REPEAT REQUEST (HARQ) SCHEME FOR A 4x4 MIMO SYSTEM, BASED ON THE EXTENDED ALAMOUTI QUASI-ORTHOGONAL SPACE-TIME BLOCK CODING (Q-STBC), IN INVARIANT AND VARIANT FADING CHANNEL

by
Jordi Ferrer Torras

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Telecommunications

Department of Electrical and Computer Engineering

May 2006

Blank Page

APPROVAL PAGE

NEW HYBRID AUTOMATIC REPEAT REQUEST (HARQ) SCHEME FOR A 4x4 MIMO SYSTEM, BASED ON THE EXTENDED ALAMOUTI QUASI-ORTHOGONAL SPACE-TIME BLOCK CODING (Q-STBC), IN INVARIANT AND VARIANT FADING CHANNEL

Jordi Ferrer Torras

Dr. Yehesked Bar-Ness, Thesis Advisor — Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Alexander M. Haimovich, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Jordi Ferrer Torras

Degree: Master of Science

Date: May 2006

Undergraduate and Graduate Education:

- Master of Science in Telecommunications,
  New Jersey Institute of Technology, Newark, NJ, 2006

- Bachelor of Science in Telecommunications,
  Escola Técnica Superior de Telecomunicacions de Barcelona (ETSETB) at Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, 2004

Major: Telecommunications

To my beloved parents, Jordi and Montserrat

ACKNOWLEDGMENT

First and foremost, I would specially like to thank my parents and all my dear family (grandmothers, uncles, aunts and cousins). Without their love, the American Dream would not have been possible.

I would also like to express my deepest appreciation to Dr. Yeheskel Bar-Ness and Ms. Elisa Sayrol who gave me this great opportunity to live an unforgettable experience in the United States. Special thanks to my other two committee members for contributing with their expert insight and approval of my Master's Thesis: Professor Alex Haimovich and Dr. Ali Abdi.

To all my closest friends that have shared so many good moments with me during these two wonderful years in NJIT, thank you very much for your unconditional support: Gerard, Muñoz, Mike, Sebastian, Emanuele, Pere, Sander, Angèl-lica, Sonia, Alexandra, Jennifer, Anne-Katrin, Grace and Anna. Furthermore, despite the distance I really appreciate the eternal friendship from all my Bon Soleil's classmates (Jean Paul, Joël, Dario, etc.) and the brilliant *Falcons* I met in the Technical University of Catalunya; thank you so much! I wish to thank all my teammates as well, both in Barcelona and Plainfield/Garden City, because life is not only studying. My special gratitude goes to Matthijs, Gonzalo, Buendi, Cata, Viran, Saba, Marc, Frank and Fitzy.

Finally, I am also very grateful to all my labmates in CWCSPR with a special mention to Guillem, Jordi, Aitor, Wendy, Igor, Krithika, Hong-Li, and Razvan. Last but not least, thanks to all the people who have somehow also played an important role in my recent graduation: Marlene, Adish, Dr. Kane, Ms. Gonzalez, Mr. Grundy, Ms. Small and so many others. Once again, thanks a lot to all of you.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF ACRONYMS

ACK – Acknowledgment

AR-1 – AutoRegressive model of order 1

ARQ – Automatic Repeat Request

BER – Bit Error Rate

CRC – Cyclic Redundancy Check

CSI – Channel State Information

D-STTD – Double Space-Time Transmit Diversity

EAC – Extended Alamouti Coding

HARQ – Hybrid Automatic Repeat reQuest

LZF – Linear Zero Forcing

MIMO – Multiple Input Multiple Output

MMSE – Minimum Mean Square Error

NACK – Negative Acknowledgment

QPSK – Quadrature Phase Shift Keying

Q-STBC – Quasi-orthogonal Space-Time Block Coding

SNR – Signal to Noise Ratio

STBC – Space Time Block Coding

ZF – Zero Forcing

CHAPTER 1

INTRODUCTION

1.1 Motivation

This Master's Thesis is a follow-up from the previous work of another NJIT student coming from Barcelona: Guillem Ernest Malagarriga. He developed in [1] a new HARQ scheme termed Multiple Alamouti Coding for an NxM antenna MIMO system and established a selection algorithm named the Determinant Criterion to get an optimal retransmission order. The goal is to find a better solution for the case of 4x4 antennas using the Extended Alamouti Quasi-orthogonal Space-Time Block Coding originally proposed in [2] and the mentioned algorithm to compare it with other schemes, both in invariant and variant fading channel conditions.

It will be shown how, choosing among several transmission matrices with some Channel State Information feedback from the receiver, the performance of the system can be considerably improved in throughput and reliability, even beating for a time invariant channel the best scheme proposed so far.

1.2 Background Information

In Wireless Communications, there is nowadays an endless quest for increased capacity and improved quality. Within this area, during the last years the research community has been studying how to use multiple antennas in a more intelligent way.

Perhaps the most striking characteristic of MIMO systems is their ability to exploit, rather than combat, multipath propagation [3], where the spatial dimension allows the improvement of the wireless data link performance.

On regular radio communications, multiplexing implies the appearance of some interference; however, with multiple antennas the additional pathways are used to transmit more information and then recombine the signal at the receiver. MIMO systems provide a substantial capacity gain over conventional single antenna systems, along with more reliable communication.

One of the most popular transmission techniques for multiple transmit antennas is the Alamouti's scheme [4], also known as Space-Time Block Coding (STBC). At a given symbol period t_1, two signals are simultaneously transmitted from two antennas; during the next one ($t_2 = t_1 + T$), the transmitted signals are switched, conjugated and have a sign change in one of them (no matter which one), as it is shown in Figure 1.1:

$$\begin{array}{cc} & \begin{array}{cc} t_1 & t_2 \end{array} \\ \begin{array}{c} \text{Antenna 1} \\ \text{Antenna 2} \end{array} & \begin{bmatrix} s_1 & s_2^* \\ s_2 & -s_1^* \end{bmatrix} \end{array}$$

Figure 1.1 Alamouti's scheme for N=2 transmit antennas.

This paradigm for communication over Rayleigh fading channels has an orthogonal structure that provides full diversity and full transmission rate, but this is not possible for more than two antennas. For instance, in [5] several complex orthogonal codes for N=3 and 4 can be found, giving full diversity but low rates (½ and ¾).

This thesis will evaluate a Quasi-Orthogonal STBC [6] called Extended Alamouti, which provides rate one but partial diversity, as a potential HARQ solution for a 4x4 MIMO system. Hybrid Automatic Repeat ReQuest is a variation of the conventional ARQ error control method, giving much better performance although with higher complexity. Instead of discarding erroneous packets and asking for retransmission, it stores and smartly combines them. Among all the HARQ existing techniques [7], this thesis will focus on the Pre-Combining scheme where the retransmitted packets are combined at the symbol-level, and the cumulative interference is removed using a Linear Zero Forcing (LZF) or a Minimum Mean-Square Error (MMSE) equalizer.

1.3 Previous Work

At the begining of the year 2005, Dr. Yeheskel Bar-Ness developed in [8] an analytical nomenclature to show how an orthogonal code, termed Multiple Alamouti, could be used as a valid scheme for HARQ in MIMO systems. The new suggested method in Figure 1.2 was characterized by a better error performance than former systems but also by a low transmission rate (*3/4* for 3 Tx antennas, and *4/7* for 4, if all the retransmissions were needed), resulting in a poor throughput, especially in the first retransmissions.

3 Element Transmitter

$s^{(1)}$	$s^{(2)}$	$s^{(3)}$	$s^{(4)}$
s_1	$-s_2^*$	s_3^*	0
s_2	s_1^*	0	$-s_3^*$
s_3	0	$-s_1^*$	s_2^*

4 Element Transmitter

$s^{(1)}$	$s^{(2)}$	$s^{(3)}$	$s^{(4)}$	$s^{(5)}$	$s^{(6)}$	$s^{(7)}$
s_1	$-s_2^*$	0	0	s_3^*	0	$-s_4^*$
s_2	s_1^*	$-s_3^*$	0	0	s_4^*	0
s_3	0	s_2^*	$-s_4^*$	$-s_1^*$	0	0
s_4	0	0	s_3^*	0	$-s_2^*$	s_1^*

Figure 1.2 Multiple Alamouti Coding scheme for 3x3 and 4x4 MIMO systems.

Based on that scheme, G. E. Malagarriga found in [1] a criterion to determine the best retransmission order, through the minimization of the Bit Error Rate (BER) or the maximization of the Signal to Noise Ratio (SNR) in each receiver branch. The Determinant of the resulting matched filtered and combined cross-correlation matrix with channel coefficients was the key factor to achieve good and consistent results.

In the middle of March 2005, LG Electronics Inc. and Nortel Networks submitted a proposal [9] for patent to the IEEE 802.16e group with a scheme that was also referred as D-STTD (Double Space-Time Transmit Diversity). They were basically using all the antennas in the 3x3/4x4 cases, respectively, with three alternatives in couples (conjugated signals for odd retransmissions and non-conjugated for the even ones) shown as follows in Figure 1.3:

3 Element Transmitter

Initial trans. | Odd retrans. | Even retrans.

$$s^{(1)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \quad s_{ALT1}^{(odd)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ s_3^* \end{bmatrix} \quad s_{ALT1}^{(even)} = \begin{bmatrix} s_3 \\ s_1 \\ s_2 \end{bmatrix}$$

$$s_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix} \quad s_{ALT2}^{(even)} = \begin{bmatrix} s_2 \\ s_3 \\ s_1 \end{bmatrix}$$

$$s_{ALT3}^{(odd)} = \begin{bmatrix} s_1^* \\ -s_3^* \\ s_2^* \end{bmatrix} \quad s_{ALT3}^{(even)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$$

4 Element Transmitter

Initial trans. | Odd retrans. | Even retrans.

$$s^{(1)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \quad s_{ALT1}^{(odd)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ -s_4^* \\ s_3^* \end{bmatrix} \quad s_{ALT1}^{(even)} = \begin{bmatrix} s_3 \\ -s_4 \\ -s_1 \\ s_2 \end{bmatrix}$$

$$s_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ -s_4^* \\ s_1^* \\ s_2^* \end{bmatrix} \quad s_{ALT2}^{(even)} = \begin{bmatrix} s_2 \\ s_1 \\ s_4 \\ s_3 \end{bmatrix}$$

$$s_{ALT3}^{(odd)} = \begin{bmatrix} -s_4^* \\ -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix} \quad s_{ALT3}^{(even)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

Figure 1.3 LG Electronics Inc. D-STTD scheme for 3x3 and 4x4 MIMO systems.

By the end of May 2005, Samsung Corp. proposed in [10] a hybrid combination between the LG solution and the Multiple Alamouti Coding, trying to exploit their particular strengths. This last also called NJIT scheme had good performance at later retransmission stage when completely orthogonalized, but a weak combining effect at early retransmissions due to its inefficiency: only two out of four antennas used to send information. On the other side, the D-STTD type retransmission (or LG) had better performance at early stage; however, it had cross interference terms yielding performance degradation at later retransmissions. In Figure 1.4 the whole signal set organization can be observed:

	$\mathbf{s}_1^{(i)}$	$\mathbf{s}_2^{(i)}$	$\mathbf{s}_3^{(i)}$	$\mathbf{s}_4^{(i)}$	$\mathbf{s}_5^{(i)}$	$\mathbf{s}_6^{(i)}$	$\mathbf{s}_7^{(i)}$	$\mathbf{s}_8^{(i)}$	$\mathbf{s}_9^{(i)}$
Tx1	$-s_2^*$	$-s_4^*$	$-s_3^*$	$-s_2^*$	0	$-s_4^*$	0	$-s_3^*$	0
Tx2	s_1^*	s_3^*	$-s_4^*$	s_1^*	0	0	s_3^*	0	$-s_4^*$
Tx3	$-s_4^*$	$-s_2^*$	s_1^*	0	$-s_4^*$	0	$-s_2^*$	s_1^*	0
Tx4	s_3^*	s_1^*	s_2^*	0	s_3^*	s_1^*	0	0	s_2^*
	(1,2)	(1,4)	(1,3)	(1,2)	(3,4)	(1,4)	(2,3)	(1,3)	(2,4)
	(3,4)	(2,3)	(2,4)	X	X	X	X	X	X

Figure 1.4 Samsung Corp. Hybrid ARQ scheme for a 4x4 MIMO system.

Nevertheless, according to the results found by G. Malagarriga in [1] using the same Determinant Criterion to maximize the SNR in all the schemes, the LG solution was still getting the best performance for the case with 4x4 antennas in a time invariant channel, as it can be shown in Figure 1.5:

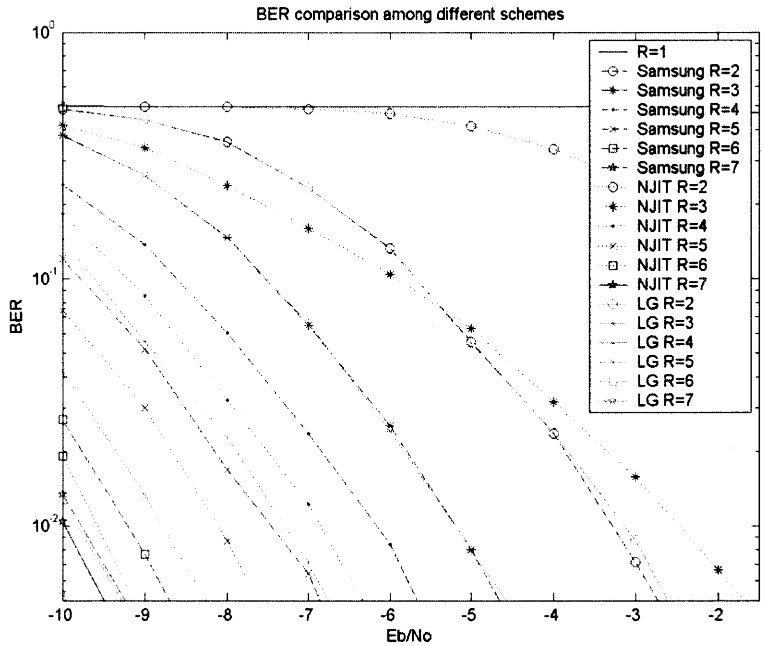

Figure 1.5 BER performance comparison among different schemes for 4x4 MIMO.

CHAPTER 2

HARQ FOR 4X4 MIMO SYSTEM USING EXTENDED ALAMOUTI CODING AND THE DETERMINANT CRITERION IN INVARIANT CHANNEL

2.1 System Model

As mentioned in the Introduction, this research is an extension of a previous work so the transmission/reception model, which is depicted in Figure 2.1, keeps the same structure as in [1], except for the proper customization of the Extended Alamouti Coding (EAC) in a 4x4 MIMO system. All the data is also summarized in Table 2.1.

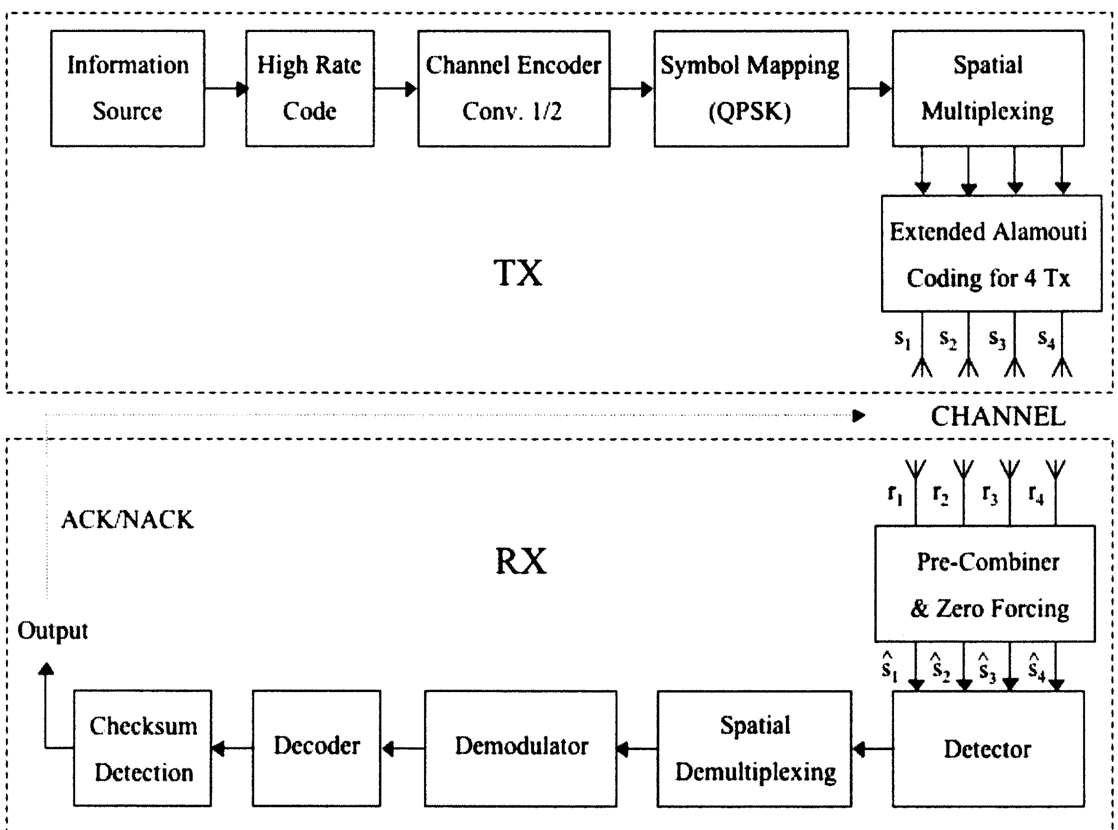

Figure 2.1 Transmitter and receiver structure for a 4x4 MIMO system with EAC.

Table 2.1 Numerical features of the Transmitter/Receiver blocks.

BLOCK	FEATURE
Information Source	522 Info Bits
High Rate Coder/Checksum Detection	16 CRC Bits
Channel Encoder/Decoder	½ Convolutional Code
Symbol Mapping/Demodulator	QPSK 2 bits/symbol
Spatial Multiplexing/Demultiplexing	540 symbols [(522+ 16+ 2)*2/2]
Extended Alamouti Coding/Pre-Combiner	s_i = 540/4 = 135 symbols

The Invariant Channel will be considered the same as well; that is, Rayleigh Fading Channel.

2.2 Analytical Description

The Extended Alamouti Block Coding scheme applied to HARQ is represented in the transmission matrix S_1 (2.1), where $s^{(1)}$ is the initial transmission and $s^{(2)}$, $s^{(3)}$ and $s^{(4)}$ are the potentially needed retransmissions.

$$\begin{matrix} \quad s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} \end{matrix}$$

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \quad (2.1)$$

The essential idea is a clear "alamoutisation" of basic (2x2) Alamouti codes:

$$S_1 = \begin{bmatrix} A & B^* \\ B & -A^* \end{bmatrix} \quad (2.2)$$

where $$A = \begin{bmatrix} s_1 & s_2^* \\ s_2 & -s_1^* \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} s_3 & s_4^* \\ s_4 & -s_3^* \end{bmatrix} \tag{2.3}$$

For a four-element transmitter four-element receiver system, the received signal from the first transmission can be modeled with:

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \quad \text{or} \quad \mathrm{r}^{(1)} = \mathrm{H}\,\mathrm{s}^{(1)} = \mathrm{n}^{(1)} \tag{2.4}$$

where r_i and n_i with $i = 1,2,3,4$ are, respectively, the received signal and noise on the ith receiver antenna; s_j with $j = 1,2,3,4$ is the transmitted signal on the jth transmitter antenna; and h_{ij} is the channel gain of the wireless link from the jth transmitter antenna to the ith receiver antenna.

After matched filtering at the receiver,

$$\mathrm{x}^{(1)} = \mathrm{H}^{\dagger}\mathrm{H}\,\mathrm{s}^{(1)} + \mathrm{H}^{\dagger}\,\mathrm{n}^{(1)} = \mathrm{C}\,\mathrm{s}^{(1)} + \mathrm{H}^{\dagger}\,\mathrm{n}^{(1)} \tag{2.5}$$

where $(\bullet)^{\dagger}$ is the operation of conjugate transpose;

$$\mathrm{C} = \mathrm{H}^{\dagger}\mathrm{H} = \begin{bmatrix} h_{11}^* & h_{21}^* & h_{31}^* & h_{41}^* \\ h_{12}^* & h_{22}^* & h_{32}^* & h_{42}^* \\ h_{13}^* & h_{23}^* & h_{33}^* & h_{43}^* \\ h_{14}^* & h_{24}^* & h_{34}^* & h_{44}^* \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \tag{2.6a}$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \tag{2.6b}$$

At the second instant, the transmitted signal is: $$s^{(2)} = \begin{bmatrix} s_2^* \\ -s_1^* \\ s_4^* \\ -s_3^* \end{bmatrix}. \quad (2.7)$$

Hence,

$$r^{(2)} = H\, s^{(2)} + n^{(2)} = H\, \mathcal{J}_1\, s^{(1)*} + n^{(2)} \quad (2.8)$$

where

$$\mathcal{J}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \quad (2.9)$$

Therefore,

$$r^{(2)*} = H^*\, \mathcal{J}_1\, s^{(1)} + n^{(2)*} \quad (2.10)$$

and multiplying by $(H^* \mathcal{J}_1)^\dagger = \mathcal{J}_1^T H^T$ after the matched filter,

$$x^{(2)} = \mathcal{J}_1^T H^T H^* \mathcal{J}_1\, s^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} \quad (2.11)$$

Combining $x^{(1)}$ and $x^{(2)}$,

$$\hat{s}_{1,2} = (C + \mathcal{J}_1^T C^* \mathcal{J}_1)\, s^{(1)} + H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} \quad (2.12)$$

where

$$C^* = H^T H^* = \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* & a_{14}^* \\ a_{12} & a_{22} & a_{23}^* & a_{24}^* \\ a_{13} & a_{23} & a_{33} & a_{34}^* \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \quad (2.13)$$

Then,

$$\mathcal{J}_1^{T}\, C^{*}\, \mathcal{J}_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^{*} & a_{13}^{*} & a_{14}^{*} \\ a_{12} & a_{22} & a_{23}^{*} & a_{24}^{*} \\ a_{13} & a_{23} & a_{33} & a_{34}^{*} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \qquad (2.14a)$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -a_{12}^{*} & a_{11} & -a_{14}^{*} & a_{13}^{*} \\ -a_{22} & a_{12} & -a_{24}^{*} & a_{23}^{*} \\ -a_{23} & a_{13} & -a_{34}^{*} & a_{33} \\ -a_{24} & a_{14} & -a_{44} & a_{34} \end{bmatrix} \qquad (2.14b)$$

$$= \begin{bmatrix} a_{22} & -a_{12} & a_{24}^{*} & -a_{23}^{*} \\ -a_{12}^{*} & a_{11} & -a_{14}^{*} & a_{13}^{*} \\ a_{24} & -a_{14} & a_{44} & -a_{34} \\ -a_{23} & a_{13} & -a_{34}^{*} & a_{33} \end{bmatrix} \qquad (2.14c)$$

so as result, the from now on called resulting matched filtered and combined cross-correlation matrix C_x:

$$C_x = C + \mathcal{J}_1^{T}\, C^{*}\, \mathcal{J}_1 = \begin{bmatrix} a_{11} + a_{22} & 0 & a_{13} + a_{24}^{*} & a_{14} - a_{23}^{*} \\ 0 & a_{22} + a_{11} & a_{23} - a_{14}^{*} & a_{24} + a_{13}^{*} \\ a_{13}^{*} + a_{24} & a_{23}^{*} - a_{14} & a_{33} + a_{44} & 0 \\ a_{14}^{*} - a_{23} & a_{24}^{*} + a_{13} & 0 & a_{44} + a_{33} \end{bmatrix} \qquad (2.15)$$

and to recover the signal $s^{(1)}$ that was sent, only Zero Forcing at the receiver is needed:

$$\hat{s} = (C_x)^{-1}\, \hat{s}_{1,2} \qquad (2.16a)$$

$$= s^{(1)} + (C + \mathcal{J}_1^{T}\, C^{*}\, \mathcal{J}_1)^{-1}\, H^{\dagger}\, n^{(1)} + (C + \mathcal{J}_1^{T}\, C^{*}\, \mathcal{J}_1)^{-1}\, \mathcal{J}_1^{T}\, H^{T}\, n^{(2)*} \qquad (2.16b)$$

In case of a third transmission, $s^{(3)}$ will be sent:

$$s^{(3)} = \begin{bmatrix} -s_3^{*} \\ -s_4^{*} \\ s_1^{*} \\ s_2^{*} \end{bmatrix} = \mathcal{J}_2\, s^{(1)\,*} \qquad (2.17)$$

with

$$\mathcal{J}_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.18}$$

Then,

$$x^{(3)} = \mathcal{J}_2^{\mathrm{T}}\, H^{\mathrm{T}} H^{*}\, \mathcal{J}_2\, s^{(1)} + \mathcal{J}_2^{\mathrm{T}}\, H^{\mathrm{T}}\, n^{(3)*} \tag{2.19}$$

so adding $x^{(3)}$,

$$\hat{s}_{1,2,3} = (C + \mathcal{J}_1^{\mathrm{T}}\, C^{*}\, \mathcal{J}_1 + \mathcal{J}_2^{\mathrm{T}}\, C^{*}\, \mathcal{J}_2)\, s^{(1)} + H^{\dagger} n^{(1)} + \mathcal{J}_1^{\mathrm{T}}\, H^{\mathrm{T}}\, n^{(2)*} + \mathcal{J}_2^{\mathrm{T}}\, H^{\mathrm{T}}\, n^{(3)*} \tag{2.20}$$

in which

$$\mathcal{J}_2^{\mathrm{T}}\, C^{*}\, \mathcal{J}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^{*} & a_{13}^{*} & a_{14}^{*} \\ a_{12} & a_{22} & a_{23}^{*} & a_{24}^{*} \\ a_{13} & a_{23} & a_{33} & a_{34}^{*} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.21a}$$

$$= \begin{bmatrix} a_{33} & a_{34}^{*} & -a_{13} & -a_{23} \\ a_{34} & a_{44} & -a_{14} & -a_{24} \\ -a_{13}^{*} & -a_{14}^{*} & a_{11} & a_{12}^{*} \\ -a_{23}^{*} & -a_{24}^{*} & a_{12} & a_{22} \end{bmatrix} \tag{2.21b}$$

Hence, now

$$C_x = C + \mathcal{J}_1^{\mathrm{T}}\, C^{*}\, \mathcal{J}_1 + \mathcal{J}_2^{\mathrm{T}}\, C^{*}\, \mathcal{J}_2 \tag{2.22a}$$

$$= \begin{bmatrix} a_{11} + a_{22} + a_{33} & a_{34}^{*} & a_{24}^{*} & -a_{14} - a_{23} - a_{23}^{*} \\ a_{34} & a_{11} + a_{22} + a_{44} & a_{23} + a_{14} + a_{14}^{*} & -a_{13}^{*} \\ a_{24} & a_{14} + a_{14}^{*} + a_{23}^{*} & a_{11} + a_{33} + a_{44} & -a_{12}^{*} \\ -a_{23} - a_{23}^{*} - a_{14}^{*} & -a_{13} & -a_{12} & a_{22} + a_{33} + a_{44} \end{bmatrix} \tag{2.22b}$$

Finally, if a fourth transmission (or third retransmission) is needed,

$$\mathrm{s}^{(4)} = \begin{bmatrix} s_4 \\ -s_3 \\ -s_2 \\ s_1 \end{bmatrix} = \mathcal{J}_3\, \mathrm{s}^{(1)} \tag{2.23}$$

with

$$\mathcal{J}_3 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.24}$$

Therefore, now

$$\mathrm{r}^{(4)} = \mathrm{H}\, \mathcal{J}_3\, \mathrm{s}^{(1)} + \mathrm{n}^{(4)} \tag{2.25}$$

and multiplying by $(\mathrm{H}\,\mathcal{J}_3)^{\dagger} = \mathcal{J}_3^{\mathrm{T}}\, \mathrm{H}^{\dagger}$ to matched filter:

$$\mathrm{x}^{(4)} = \mathcal{J}_3^{\mathrm{T}}\, \mathrm{H}^{\mathrm{T}}\, \mathrm{H}\, \mathcal{J}_3\, \mathrm{s}^{(1)} + \mathcal{J}_3^{\mathrm{T}}\, \mathrm{H}^{\dagger}\, \mathrm{n}^{(4)} \tag{2.26}$$

By adding $\mathrm{x}^{(4)}$ to the former combined signals,

$$\hat{\mathrm{s}}_{1,2,3,4} = (\mathrm{C} + \mathcal{J}_1^{\mathrm{T}}\mathrm{C}^{*}\mathcal{J}_1 + \mathcal{J}_2^{\mathrm{T}}\mathrm{C}^{*}\mathcal{J}_2 + \mathcal{J}_3^{\mathrm{T}}\mathrm{C}\mathcal{J}_3)\, \mathrm{s}^{(1)} + \mathrm{H}^{\dagger}\, \mathrm{n}^{(1)} + \mathcal{J}_1^{\mathrm{T}}\mathrm{H}^{\mathrm{T}}\mathrm{n}^{(2)*} + \mathcal{J}_2^{\mathrm{T}}\mathrm{H}^{\mathrm{T}}\mathrm{n}^{(3)*} + \mathcal{J}_3^{\mathrm{T}}\, \mathrm{H}^{\dagger}\, \mathrm{n}^{(4)}$$

Since

$$\mathcal{J}_3^{\mathrm{T}}\mathrm{C}\mathcal{J}_3 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^{*} & a_{22} & a_{23} & a_{24} \\ a_{13}^{*} & a_{23}^{*} & a_{33} & a_{34} \\ a_{14}^{*} & a_{24}^{*} & a_{34}^{*} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.27a}$$

$$= \begin{bmatrix} a_{44} & -a_{34}^{*} & -a_{24}^{*} & a_{14}^{*} \\ -a_{34} & a_{33} & a_{23}^{*} & -a_{13}^{*} \\ -a_{24} & a_{23} & a_{22} & -a_{12}^{*} \\ a_{14} & -a_{13} & -a_{12} & a_{11} \end{bmatrix} \tag{2.27b}$$

the resulting matched filtered and combined cross-correlation matrix is:

$$C_{X1} = C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2 + \mathcal{J}_3^T C \mathcal{J}_3 \tag{2.28a}$$

$$= \begin{bmatrix} a_{11}+a_{22}+a_{33}+a_{44} & 0 & 0 & a_{14}+a_{14}^*-a_{23}-a_{23}^* \\ 0 & a_{11}+a_{22}+a_{33}+a_{44} & a_{23}+a_{23}^*-a_{14}-a_{14}^* & 0 \\ 0 & a_{23}+a_{23}^*-a_{14}-a_{14}^* & a_{11}+a_{22}+a_{33}+a_{44} & 0 \\ a_{14}+a_{14}^*-a_{23}-a_{23}^* & 0 & 0 & a_{11}+a_{22}+a_{33}+a_{44} \end{bmatrix}$$

$$= A \begin{bmatrix} 1 & 0 & 0 & X_1 \\ 0 & 1 & -X_1 & 0 \\ 0 & -X_1 & 1 & 0 \\ X_1 & 0 & 0 & 1 \end{bmatrix} \tag{2.28b}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{2.29}$$

and

$$X_1 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \tag{2.30}$$

The matrix in (2.28b) proves that the Extended Alamouti is a Q-STBC (Quasi-orthogonal Space-Time Block Code). As a trade-off between the LG solution and the Multi-Alamouti Coding, the proposed scheme will use the four antennas keeping part of the orthogonality, so also cancelling some interference.

2.3 Selection Algorithm

EA-QSTBC:

$$\begin{matrix} & s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} \end{matrix}$$

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \tag{2.31}$$

Analizing the transmission matrix in (2.31), the possible retransmission orders in terms of HARQ are the following:

$$\text{a)} \quad s^{(1)} \quad s^{(2)} \quad s^{(3)} \quad s^{(4)} \tag{2.32}$$

$$\text{b)} \quad s^{(1)} \quad s^{(3)} \quad s^{(2)} \quad s^{(4)} \tag{2.33}$$

$$\text{c)} \quad s^{(1)} \quad s^{(2)} \quad s^{(4)} \quad s^{(3)} \tag{2.34}$$

$$\text{d)} \quad s^{(1)} \quad s^{(3)} \quad s^{(4)} \quad s^{(2)} \tag{2.35}$$

Note that $s^{(4)}$ will never be the first retransmission, because without the conjugated signals it's not cancelling any interference with $s^{(1)}$, due to the lack of "Alamoutization".

So far, the best order selection algorithm found is the Determinant Criterion used in [1], where the SNR in each of the receiver antennas is maximized. As it was done for the Multiple Alamouti Coding, it first has to be checked if $R_{NN} = \sigma^2 (C_x^{-1})^\dagger$, where σ^2 is the noise variance, and C_x the resulting matched filtered and combined cross-correlation matrix analized in the former section.

Since for Extended Alamouti information in all the antennas is sent, there is no worry about the Power Normalization, and the signal power has just to be considered equal to 1. Hence, in that case the SNR will directly be the inverse of the noise power, determined through its autocorrelation matrix. If the expression mentioned above can be proved, the Determinant Criterion will keep properly doing its function: to select the retransmission that minimizes the BER maximizing the SNR.

The main doubt is focused on the last retransmission $s^{(4)}$, where there are non-conjugated signals, so for simplicity the case when $s^{(1)}$, $s^{(2)}$ and $s^{(4)}$ are sent has been taken, but it can also be easily shown with the rest of combinations. Then, after combining and zero forcing:

$$\hat{s}_{1,2,,4} = s^{(1)} + (C_x)^{-1} (H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} + \mathcal{J}_3^T H^\dagger n^{(4)}) \qquad (2.36)$$

with

$$C_x = (C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_3^T C \mathcal{J}_3) \qquad (2.37)$$

Now,

$$R_{NN} = E\{[C_x^{-1}(H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} + \mathcal{J}_3^T H^\dagger n^{(4)})][(n^{(4)} H^\dagger \mathcal{J}_3^T + n^{(2)*} H^T \mathcal{J}_1^T + n^{(1)} H^\dagger)C_x^{-1}]^\dagger\}$$

$$= E\{[C_x^{-1}(H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} + \mathcal{J}_3^T H^\dagger n^{(4)})][(n^{(4)\,T} H \mathcal{J}_3^* + n^{(2)\,T} H^* \mathcal{J}_1^* + n^{(1)\dagger} H) (C_x^{-1})^\dagger]\}$$

$$= \sigma^2 C_x^{-1} H^\dagger H (C_x^{-1})^\dagger + \sigma^2 C_x^{-1}(\mathcal{J}_1^T H^T H \mathcal{J}_1^*)(C_x^{-1})^\dagger + \sigma^2 C_x^{-1}(\mathcal{J}_3^T H^\dagger H \mathcal{J}_3^*)(C_x^{-1})^\dagger$$

$$= \sigma^2 C_x^{-1} (C + \mathcal{J}_1^T C^* \mathcal{J}_1^* + \mathcal{J}_3^T C \mathcal{J}_3^*)(C_x^{-1})^\dagger$$

$$= \sigma^2 (C_x^{-1})^\dagger \qquad (2.38)$$

Note that $(C + \mathcal{J}_1^T C^* \mathcal{J}_1^* + \mathcal{J}_3^T C \mathcal{J}_3^*) = C_x$ because $\mathcal{J}_1^* = \mathcal{J}_1$ and $\mathcal{J}_3^* = \mathcal{J}_3$ (all real values); also, $n^{(i)}$ with i=1,2,3,4 are always assumed uncorrelated and their variance equal to σ^2.

After showing that the Determinant Criterion is valid for the Extended Alamouti Coding, the simulations can be performed. However, it has to be kept in mind that, although not the optimal algorithm, it's the best one found so far.

2.4 Simulation Results

The data and matlab code based on [1] have been customized for the LG/Samsung schemes and the Extended Alamouti Coding. Note that for the latter, after the third retransmission Zero Forcing has to be done as well in this case because the resulting matrix is not completely orthogonal so there is still some interference from the other branches of the channel.

The Bit Error Rates versus E_b/N_0 are depicted in Figure 2.2 for R=2, 3, 4, 5 and 6. R=1 would represent the first regular transmission and R=4 shows the result after completing the four transmissions (or three retransmissions). For R=5 and R=6 in Extended Alamouti the whole algorithm is restarted with the initial transmission again.

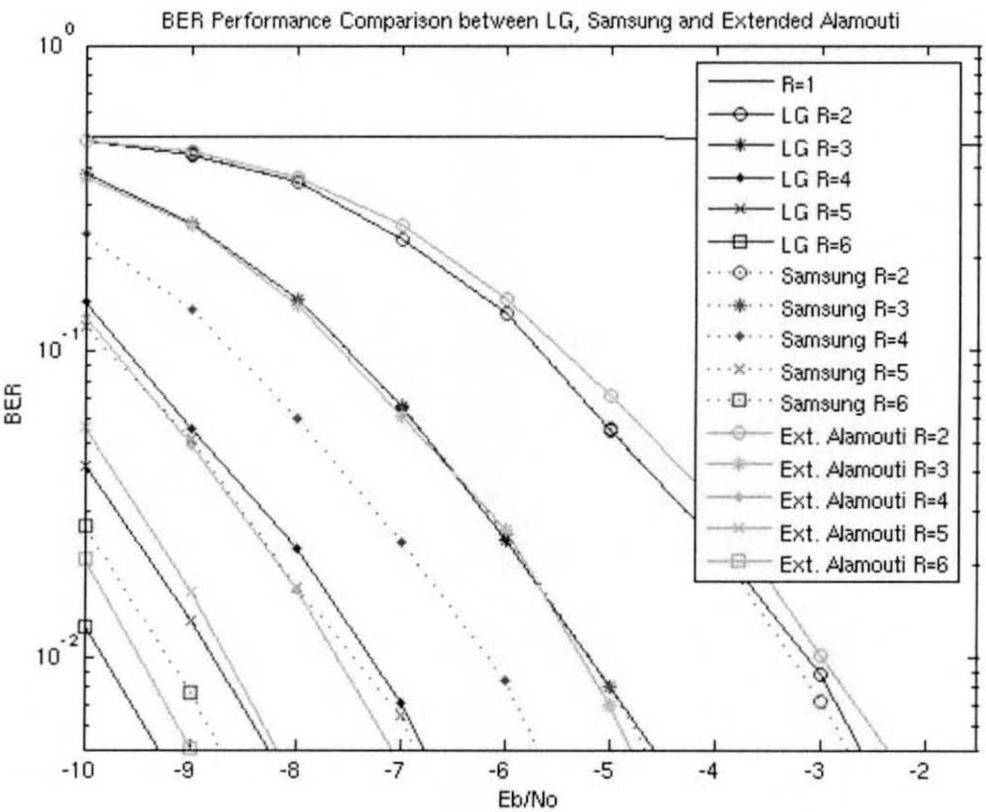

Figure 2.2 BER performance comparison between LG, Samsung and Ext. Alamouti.

Figure 2.3 compares the performance of the three systems in terms of throughput for only R=2, 3, 4, because it's enough to get the idea.

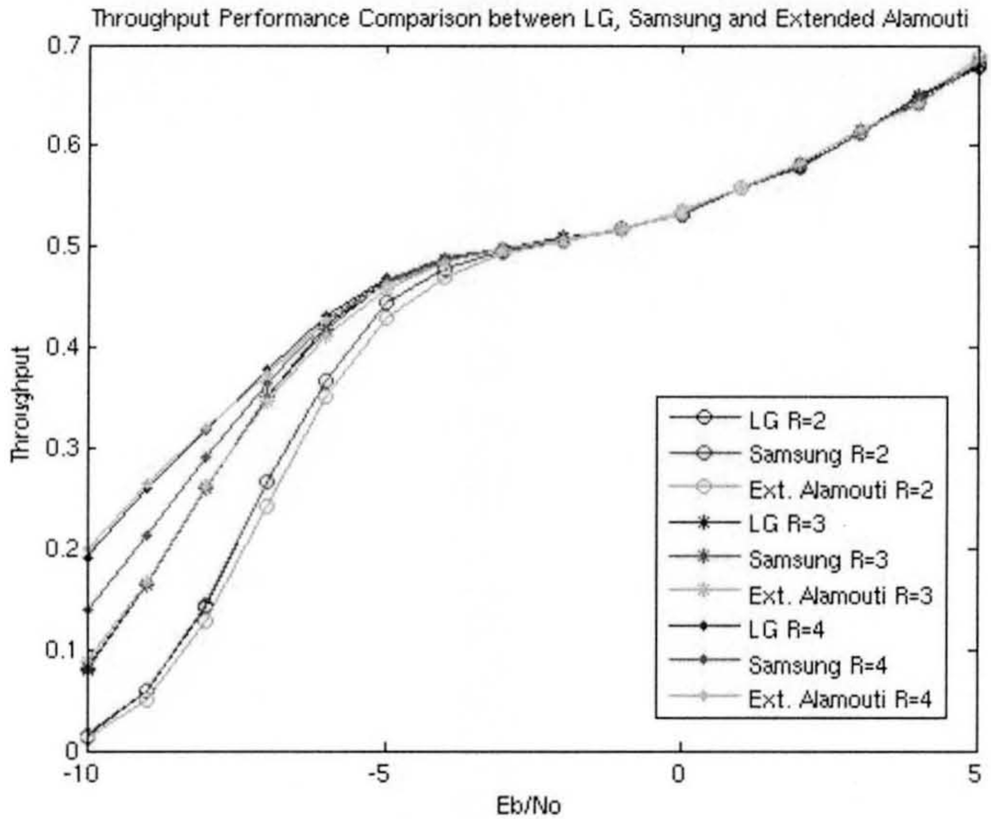

Figure 2.3 Throughput performance comparison between LG, Samsung and EAC.

The conclusion from the results of the simulations is that the Extended Alamouti solution behaves pretty similarly to the best scheme found so far (LG). In throughput, it's nearly the same because it also uses the four antennas in each retransmission (the Samsung's scheme throughput decreases after the two first retransmissions because of the Multiple Alamouti Coding at later stage). In BER performance, EAC is a little bit worse than LG for R=2 (since it's cyclic, for R=5 as well), but better for R=4 and the same for R=3. Let's see if a new modification that would allow an improvement can be found.

CHAPTER 3

HARQ FOR 4X4 MIMO SYSTEM USING ALTERANTIVE MATRICES FROM AN EAC AND PERFORMANCE COMPARISON WITH OTHER SCHEMES

3.1 Extended Alamouti with One Alternative Matrix

Looking at the results of the Extended Alamouti Coding applied to the HARQ for MIMO systems in the Chapter 2, the conclusion was that only the fourth transmission was better than the LG solution. The idea now is to check if what is called the Alternative Matrix used in [11] would give a better performance in all the retransmissions.

3.3.1 Analytical Description

So far, the Extended Alamouti transmission matrix was used as it follows:

$$\begin{matrix} s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} \end{matrix}$$

$$\mathrm{S}_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.1.1)$$

Now, from the mentioned paper [11], an Alternative Matrix which only differs from (3.1) with the sign of the transmitted signals in the first raw exists, that is:

$$\begin{matrix} s^{(1)'} & s^{(2)'} & s^{(3)'} & s^{(4)'} \end{matrix}$$

$$\mathrm{S}_2 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.1.2)$$

If the new development with the $\mathcal{J}$ nomenclature is done, the best transmission matrix will have to be determined from the begining because $\mathcal{J}_0$ is not going to be the assumed identity matrix anymore (as it usually was):

With

$$s^{(1)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \qquad \text{and} \qquad s^{(1)'} = \begin{bmatrix} -s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} \tag{3.1.3}$$

if the new alternative is transmitted, the received signal is the following:

$$r^{(1)'} = H\, s^{(1)'} + n^{(1)} \tag{3.1.4a}$$

$$= H\, \mathcal{J}_0\, s^{(1)} + n^{(1)} \tag{3.1.4b}$$

where

$$\mathcal{J}_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1.5}$$

So after matched filtering at the receiver,

$$x^{(1)'} = \mathcal{J}_0^{T}\, H^{\dagger} H\, \mathcal{J}_0\, s^{(1)} + \mathcal{J}_0^{T}\, H^{\dagger}\, n^{(1)} \tag{3.1.6a}$$

$$= \mathcal{J}_0^{T}\, C\, \mathcal{J}_0\, s^{(1)} + \mathcal{J}_0^{T}\, H^{\dagger}\, n^{(1)} \tag{3.1.6b}$$

At the second instant, the signal $s^{(2)'} = \begin{bmatrix} -s_2^* \\ -s_1^* \\ s_4^* \\ -s_3^* \end{bmatrix}$ is transmitted, hence

$$r^{(2)'} = H\, s^{(2)'} + n^{(2)} = H\, \mathcal{J}_1\, s^{(1)*} + n^{(2)} \tag{3.1.7}$$

where

$$\mathcal{J}_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{3.1.8}$$

Therefore,

$$r^{(2),*} = H^* \mathcal{J}_1 s^{(1)} + n^{(2)*} \tag{3.1.9}$$

and after matched filtering,

$$x^{(2),} = \mathcal{J}_1^{T} H^{T} H^* \mathcal{J}_1 s^{(1)} + \mathcal{J}_1^{T} H^{T} n^{(2)*} \tag{3.1.10}$$

so adding both terms to get the output for decision

$$\hat{s} = (\mathcal{J}_0^{T} C \mathcal{J}_0 + \mathcal{J}_1^{T} C^* \mathcal{J}_1) s^{(1)} + \mathcal{J}_0^{T} H^{\dagger} n^{(1)} + \mathcal{J}_1^{T} H^{T} n^{(2)*} \tag{3.1.11}$$

Now,

$$\mathcal{J}_0^{T} C \mathcal{J}_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1.12a}$$

$$= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -a_{11} & -a_{12} & -a_{13} & -a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \tag{3.1.12b}$$

$$= \begin{bmatrix} a_{11} & -a_{12} & -a_{13} & -a_{14} \\ -a_{12}^* & a_{22} & a_{23} & a_{24} \\ -a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ -a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \tag{3.1.12c}$$

$$\mathcal{J}_1^{T} C^* \mathcal{J}_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* & a_{14}^* \\ a_{12} & a_{22} & a_{23}^* & a_{24}^* \\ a_{13} & a_{23} & a_{33} & a_{34}^* \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{3.1.13a}$$

$$= \begin{bmatrix} a_{22} & a_{12} & a_{24}^* & -a_{23}^* \\ a_{12}^* & a_{11} & a_{14}^* & -a_{13}^* \\ a_{24} & a_{14} & a_{44} & -a_{34} \\ -a_{23} & -a_{13} & -a_{34}^* & a_{33} \end{bmatrix} \tag{3.1.13b}$$

$$\mathcal{J}_0^{\mathsf{T}}\,\mathrm{C}\,\mathcal{J}_0 + \mathcal{J}_1^{\mathsf{T}}\,\mathrm{C}^*\,\mathcal{J}_1 = \begin{bmatrix} a_{11} & -a_{12} & -a_{13} & -a_{14} \\ -a_{12}^* & a_{22} & a_{23} & a_{24} \\ -a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ -a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} + \begin{bmatrix} a_{22} & a_{12} & a_{24}^* & -a_{23}^* \\ a_{12}^* & a_{11} & a_{14}^* & -a_{13}^* \\ a_{24} & a_{14} & a_{44} & -a_{34} \\ -a_{23} & -a_{13} & -a_{34}^* & a_{33} \end{bmatrix} \tag{3.1.14a}$$

$$= \begin{bmatrix} a_{11}+a_{22} & 0 & a_{24}^*-a_{13} & -a_{14}-a_{23}^* \\ 0 & a_{11}+a_{22} & a_{23}+a_{14}^* & a_{24}-a_{13}^* \\ a_{24}-a_{13}^* & a_{14}+a_{23}^* & a_{33}+a_{44} & 0 \\ -a_{23}-a_{14}^* & a_{24}^*-a_{13} & 0 & a_{33}+a_{44} \end{bmatrix} \tag{3.1.14b}$$

Similarly,

$$\mathcal{J}_0^{\mathsf{T}}\,\mathrm{C}\,\mathcal{J}_0 + \mathcal{J}_1^{\mathsf{T}}\,\mathrm{C}^*\,\mathcal{J}_1 + \mathcal{J}_2^{\mathsf{T}}\,\mathrm{C}^*\,\mathcal{J}_2 = \tag{3.1.15}$$

$$= \begin{bmatrix} a_{11}+a_{22}+a_{33} & a_{34}^* & a_{24}^* & -a_{14}-a_{23}-a_{23}^* \\ a_{34} & a_{11}+a_{22}+a_{44} & a_{23}+a_{14}+a_{14}^* & -a_{13}^* \\ a_{24} & a_{14}+a_{14}^*+a_{23}^* & a_{11}+a_{33}+a_{44} & -a_{12}^* \\ -a_{23}-a_{23}^*-a_{14}^* & -a_{13} & -a_{12} & a_{22}+a_{33}+a_{44} \end{bmatrix}$$

Finally,

$$\mathrm{C_X} = \mathcal{J}_0^{\mathsf{T}}\,\mathrm{C}\,\mathcal{J}_0 + \mathcal{J}_1^{\mathsf{T}}\,\mathrm{C}^*\,\mathcal{J}_1 + \mathcal{J}_2^{\mathsf{T}}\,\mathrm{C}^*\,\mathcal{J}_2 + \mathcal{J}_3^{\mathsf{T}}\,\mathrm{C}\,\mathcal{J}_3 = \tag{3.1.16}$$

$$= \begin{bmatrix} a_{11}+a_{22}+a_{33}+a_{44} & 0 & 0 & -a_{14}-a_{23}^*-a_{23}-a_{14}^* \\ 0 & a_{11}+a_{22}+a_{33}+a_{44} & a_{14}+a_{23}^*+a_{23}+a_{14}^* & 0 \\ 0 & a_{23}+a_{14}^*+a_{14}+a_{23}^* & a_{11}+a_{22}+a_{33}+a_{44} & 0 \\ -a_{14}-a_{23}^*-a_{23}-a_{14}^* & 0 & 0 & a_{11}+a_{22}+a_{33}+a_{44} \end{bmatrix}$$

where

$$\mathcal{J}_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{3.1.17}$$

and

$$J_3 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.1.18}$$

It can be shown that an entire family of EA-STBCs derived by sign changes and, alternatively, permutations of the transmit antennas behave equivalently in terms of their nearly-orthogonality. However, for a fixed channel, the BER performance will be different depending on the so-called off main-diagonal interference X_i:

$$C_{Xi} = A \begin{bmatrix} 1 & 0 & 0 & X_i \\ 0 & 1 & -X_i & 0 \\ 0 & -X_i & 1 & 0 \\ X_i & 0 & 0 & 1 \end{bmatrix} \quad \text{for i=1,2 in these two examples of EA-STBCs} \tag{3.1.19}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.1.20}$$

and

$$X_1 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \qquad \text{if } S_1 \text{ is sent} \tag{3.1.21}$$

$$X_2 = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \qquad \text{if } S_2 \text{ is sent} \tag{3.1.22}$$

Since the term in the diagonal A (signal power) will be the same, taking the lowest interference the BER performance will be improved. Therefore, in the simulations, before even sending the first transmission (in practice, probably through a training signal to know the channel conditions) the matrix that minimizes $|X|$ will be checked and then its transmission order will be decided with the Determinant Criterion.

3.1.2 Simulations Results:

To see the performance of this proposal called Alternative Matrix, it is compared in the following figures with both the previous simple Extended Alamouti and the LG solution, which was the best one in all former comparisons.

As always, the BER and Throughput Performance are depicted versus the E_b/N_0 (in dB) for the cases of R=2, 3, 4, 5 and 6 to see the evolution in all the retransmissions, even after one cylce is completed in the Extended Alamouti Coding.

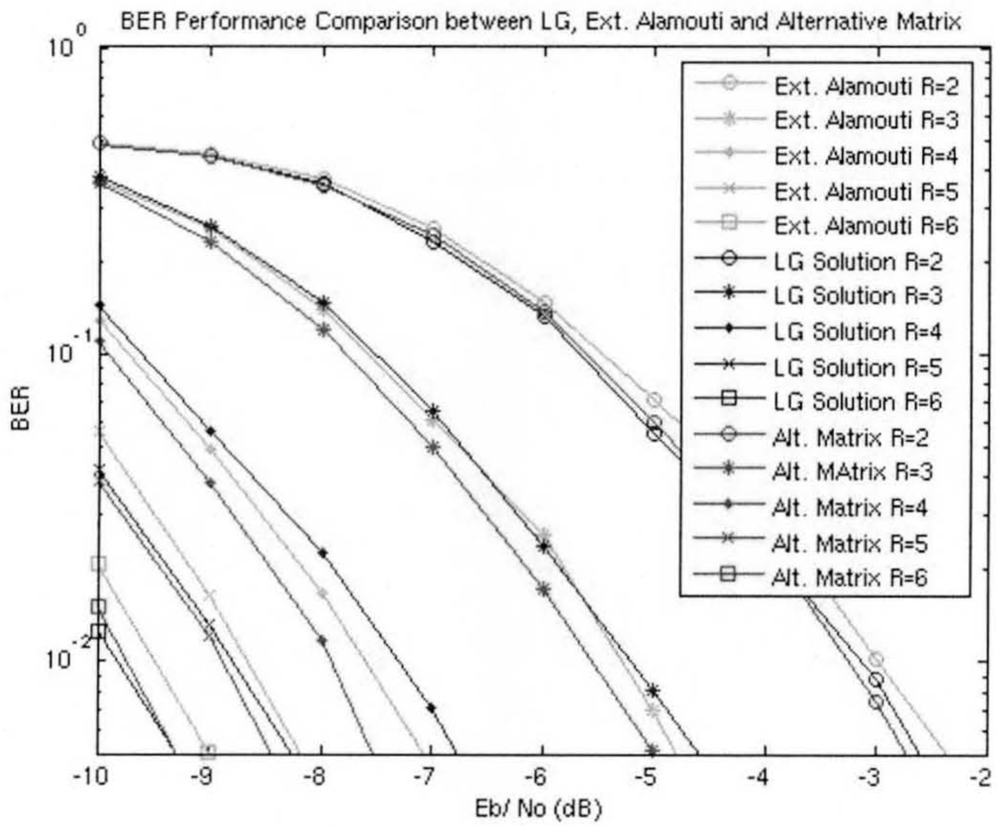

Figure 3.1 BER performance comparison between Extended Alamouti, LG and Alternative Matrix.

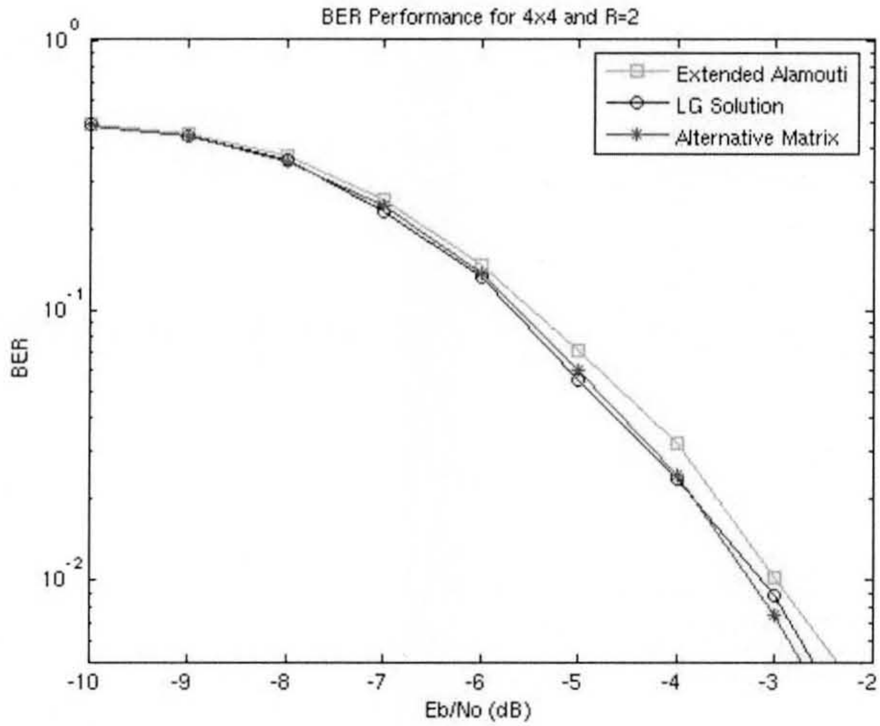

Figure 3.2 BER performance for 4x4 and R=2.

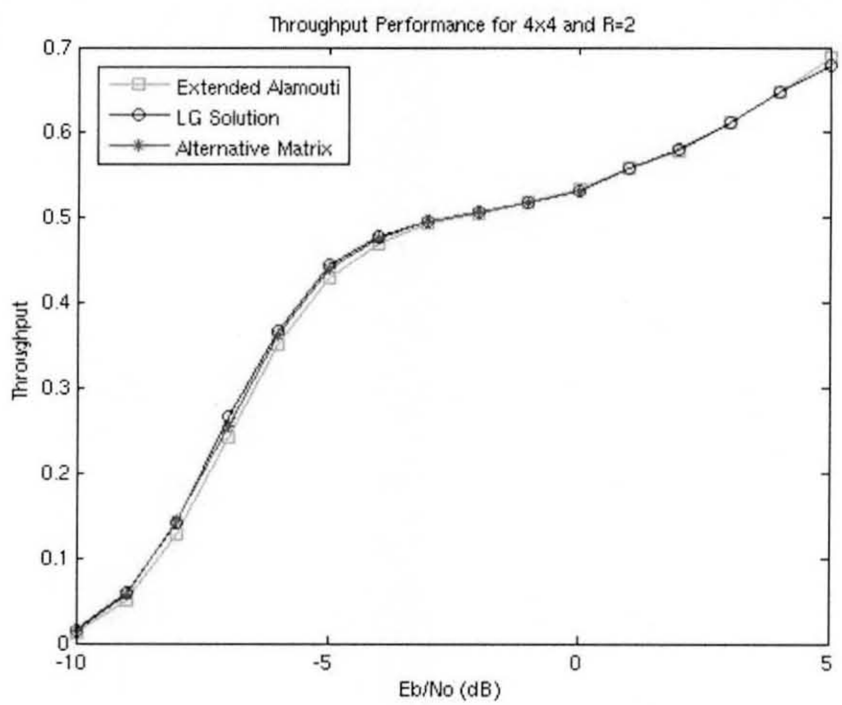

Figure 3.3 Throughput performance for 4x4 and R=2.

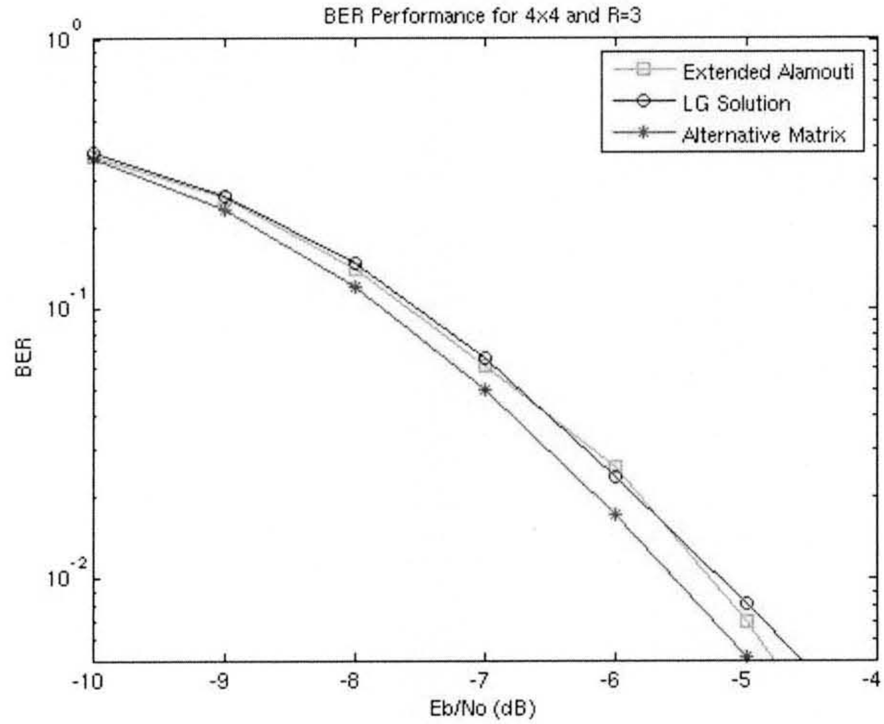

Figure 3.4 BER performance for 4x4 and R=3.

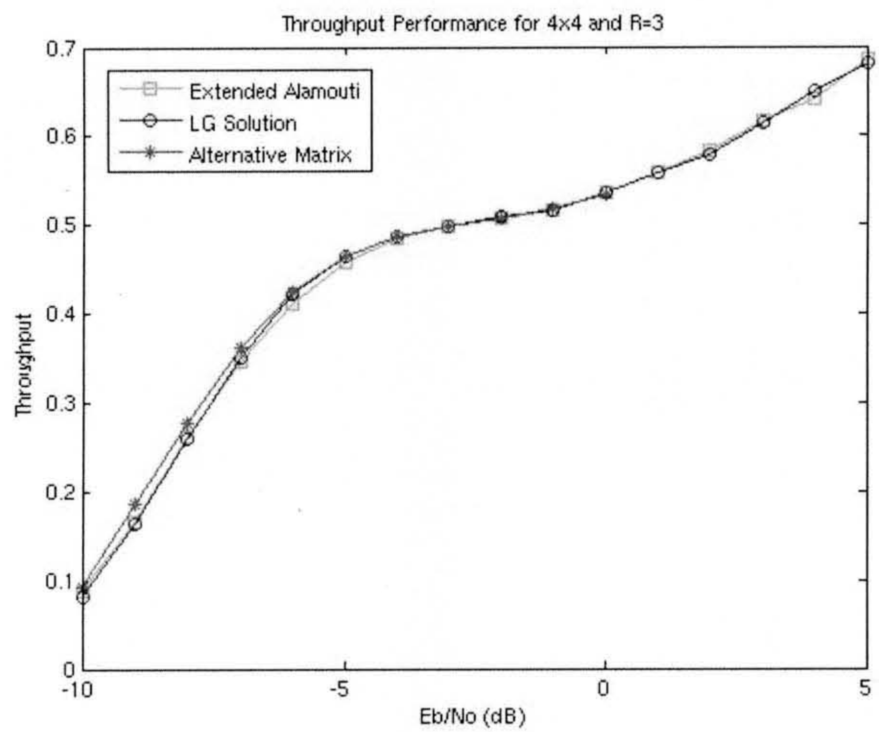

Figure 3.5 Throughput performance for 4x4 and R=3.

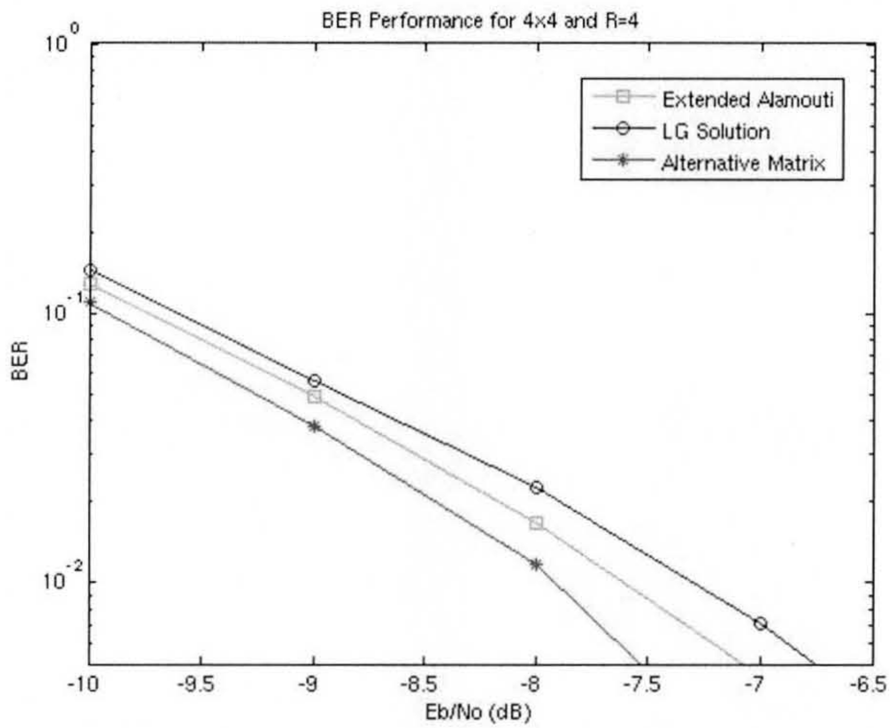

Figure 3.6 BER performance for 4x4 and R=4.

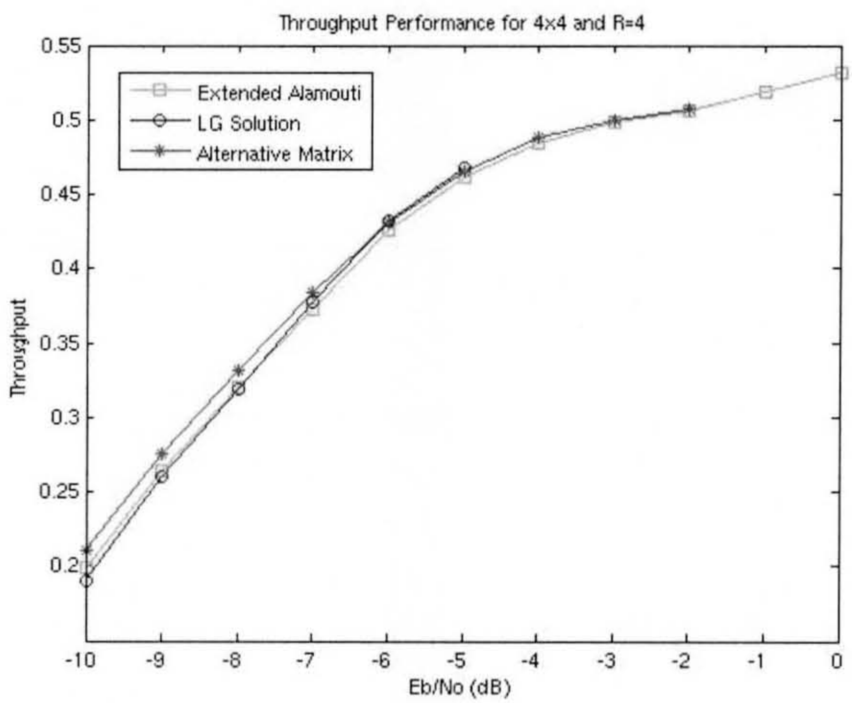

Figure 3.7 Throughput performance for 4x4 and R=4.

After having done the simulations for all the solutions with the Determinant Criterion, which is the best one found so far, the Alternative Matrix should be seriously considered because it performs better than all the rest for R=3, 4, 5 and pretty similarly to the LG solution for R=2 and R=6. This is due to the fact that the system is actually optimized to minimize the interference at the completion of the cylce (R=4), when it will be quasi-orthogonal.

Right now, the Alternative Matrix used in [11] has just been tried, but it exists a whole family of EA-STBC's that can be used, playing with the sign changes and the retransmission order, so that the system could even be a little bit more improved. However, as compensation, instead of only one bit of feedback, as many bits as necessary to code all the combinations would then be needed.

The strength of this proposed solution for flat channel is that it partially keeps the analytical "beauty" with the quasi-orthogonality, without losing in performance, specially in throughput, since all the antennas are used in every retransmission (whereas the Multiple Alamouti doesn't, for instance).

3.1.3 Clarifying Extension

In the previous section, it has been shown how using an alternative matrix can improve the performance of the system.

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad S_2 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.1.23)$$

To better understand its behaviour, Figures 3.8-10 depict the BER performance of the different cases where either "only S1" (formerly called "Extended Alamouti"), "only S2" (a new case expected to be similar to the former) or the proposed scheme choosing the best (with the lowest interference at the fourth transmission) between S1 or S2 (formerly called "Alternative Matrix") is sent, plus the LG scheme for comparison purpose.

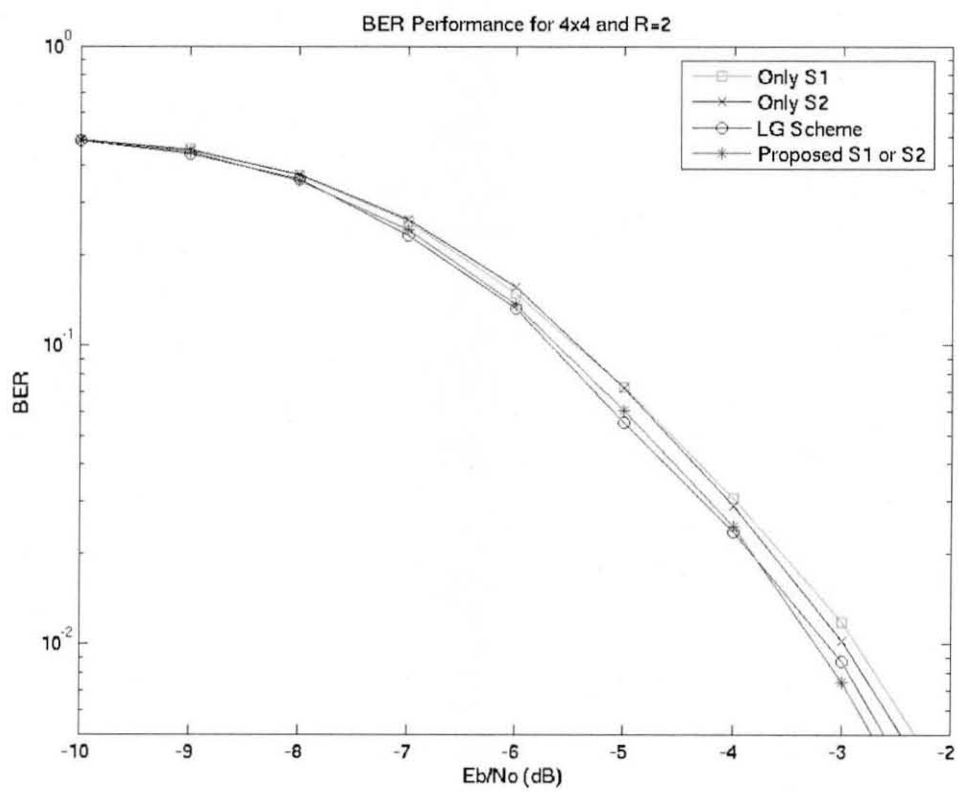

Figure 3.8 BER performance for 4x4 and R=2 (II).

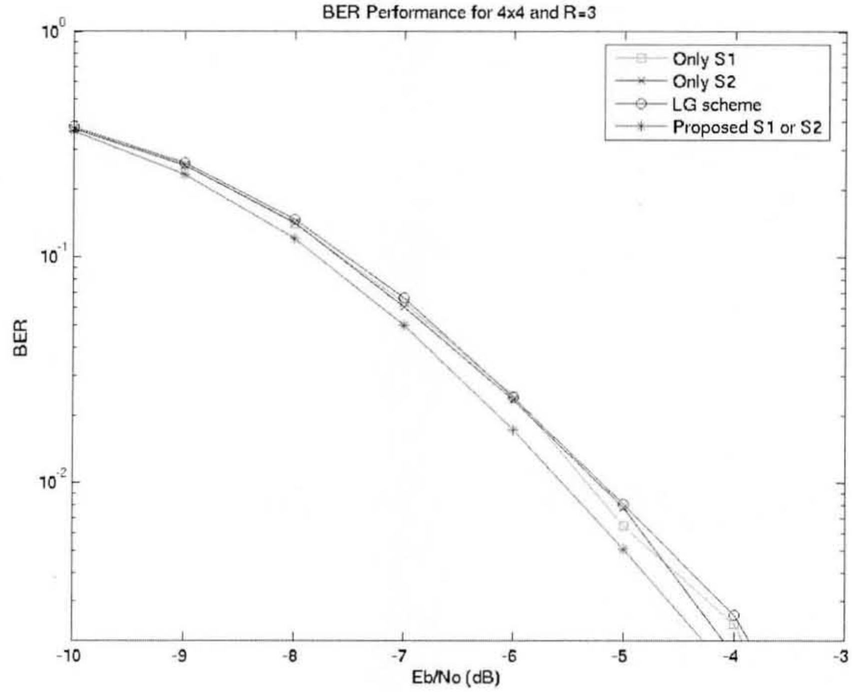

Figure 3.9 BER performance for 4x4 and R=3 (II).

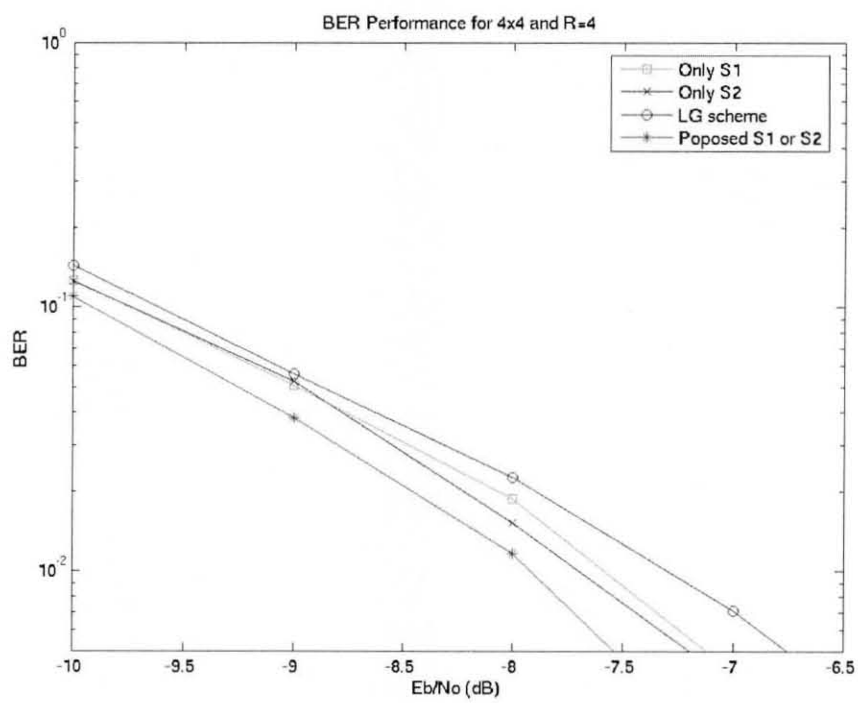

Figure 3.10 BER performance for 4x4 and R=4 (II).

3.2 Extended Alamouti with Sign Changes in Alternative Matrices

From the previous figures it can be noticed that by taking the best of two alternatives is always better than sending only one. Therefore, as it was said in the last section, there is an entire family of EA-STBC's coming from sign changes in different raws to keep the nearly-orthogonality. As a next step, let's analize all these possible combinations.

3.2.1 Analytical Description

Starting again from the development of two Aternative Matrices:

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \quad \text{and} \quad S_2 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \tag{3.2.1}$$

The resulting matrix after the fourth transmission was:

$$G_i = A \begin{bmatrix} 1 & 0 & 0 & X_i \\ 0 & 1 & -X_i & 0 \\ 0 & -X_i & 1 & 0 \\ X_i & 0 & 0 & 1 \end{bmatrix} \quad \text{with} \quad A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.2.2}$$

with

$$X_1 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_1 \text{ is sent} \tag{3.2.3}$$

and

$$X_2 = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_2 \text{ is sent} \tag{3.2.4}$$

Now, as new combinations:

$$S_3 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix}$$ with a sign change in all the second raw. (3.2.5)

$$S_4 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix}$$ with a sign change in all the third raw. (3.2.6)

$$S_5 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix}$$ with a sign change in all the fourth raw. (3.2.7)

So, if the resulting matrix for all of them is calculated, almost exactly the same nearly-orthogonality appears with a predictable issue:

$$X_3 = \frac{2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_3 \text{ is sent} \tag{3.2.8}$$

$$X_4 = \frac{2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_4 \text{ is sent} \tag{3.2.9}$$

$$X_5 = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_5 \text{ is sent} \tag{3.2.10}$$

where

$$X_3 = X_4 \tag{3.2.11}$$

and

$$X_5 = X_2. \tag{3.2.12}$$

This fact means that after the fourth transmission, the BER performance of the system should be nearly the same for the cases when for instance S_2 or S_5 is sent (also S_3 or S_4).

If more sign changes in several raws are made at the same time, it can be shown as an example:

$$S_6 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \text{ with a sign change in the first and the second raw.} \quad (3.2.13)$$

Now, the resulting matrix is:

$$X_6 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_6 \text{ is sent} \quad (3.2.14)$$

where

$$X_6 = X_1 \quad (3.2.15)$$

Right now, several options can be tried with Matlab simulations, but a better criterion to decide among all the possible Alternatives is needed because the "after the fourth transmission" resulting matrix doesn't help to distinguish from several alternatives. Anyway, the system is getting improved and the combinations are increasing, so this line of study looks like being promising in terms of getting the best performance for a 4x4 MIMO scheme.

All the possible Alternative Matrices with sign changes are the following:

$$s^{(1)} \quad s^{(2)} \quad s^{(3)} \quad s^{(4)}$$

Original Extended Alamouti:

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.2.16)$$

1-raw sign change: (1st)

$$S_2 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.2.17)$$

1-raw sign change: (2nd)

$$S_3 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.2.18)$$

1-raw sign change: (3rd)

$$S_4 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad (3.2.19)$$

1-raw sign change: (4th)

$$S_5 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \qquad (3.2.20)$$

2-raws sign change: (1st & 2nd) $$S_6 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \quad (3.2.21)$$

2-raws sign change: (3rd & 4th) $$S_7 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.22)$$

2-raws sign change: (1st & 4th) $$S_8 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.23)$$

2-raws sign change: (2nd & 3rd) $$S_9 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \quad (3.2.24)$$

2-raws sign change: (1st & 3rd) $$S_{10} = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \quad (3.2.25)$$

2-raws sign change: (2nd & 4th) $$S_{11} = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.26)$$

3-raws sign change: (1st, 2nd & 3rd) $$S_{12}=\begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \quad (3.2.27)$$

3-raws sign change: (2nd, 3rd & 4th) $$S_{13}=\begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.28)$$

3-raws sign change: (1st, 2nd & 4th) $$S_{14}=\begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.29)$$

3-raws sign change: (1st, 3rd & 4th) $$S_{15}=\begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.30)$$

and finally, 4-raws sign change: (1st, 2nd, 3rd & 4th)

$$S_{16}=\begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} \quad (3.2.31)$$

Let's remember that the resulting matrix after the fourth transmission was:

$$C_{Xi} = A \begin{bmatrix} 1 & 0 & 0 & X_i \\ 0 & 1 & -X_i & 0 \\ 0 & -X_i & 1 & 0 \\ X_i & 0 & 0 & 1 \end{bmatrix} \quad \text{with} \quad A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.2.32}$$

and

$$X_1 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_1 \text{ is sent,} \tag{3.2.33}$$

$$X_2 = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_2 \text{ is sent,} \tag{3.2.34}$$

$$X_3 = \frac{2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_3 \text{ is sent,} \tag{3.2.35}$$

$$X_4 = \frac{2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_4 \text{ is sent,} \tag{3.2.36}$$

$$X_5 = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_5 \text{ is sent,} \tag{3.2.37}$$

$$X_6 = \frac{-2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_6 \text{ is sent,} \tag{3.2.38}$$

$$X_7 = \frac{-2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_7 \text{ is sent,} \tag{3.2.39}$$

$$X_8 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_8 \text{ is sent,} \tag{3.2.40}$$

$$X_9 = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_9 \text{ is sent,} \tag{3.2.41}$$

$$X_{10} = \frac{-2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_{10} \text{ is sent,} \tag{3.2.42}$$

$$X_{11} = \frac{-2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_{11} \text{ is sent,} \tag{3.2.43}$$

$$X_{12} = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_{12} \text{ is sent,} \tag{3.2.44}$$

$$X_{13} = \frac{-2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_{13} \text{ is sent,} \tag{3.2.45}$$

$$X_{14} = \frac{2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_{14} \text{ is sent,} \tag{3.2.46}$$

$$X_{15} = \frac{2\mathrm{Re}(a_{14} + a_{23})}{A} \quad \text{if } S_{15} \text{ is sent,} \tag{3.2.47}$$

$$X_{16} = \frac{2\mathrm{Re}(a_{14} - a_{23})}{A} \quad \text{if } S_{16} \text{ is sent,} \tag{3.2.48}$$

where

$$X_1 = X_8 = X_9 = X_{16}, \tag{3.2.49}$$

$$X_2 = X_5 = X_{12} = X_{13}, \tag{3.2.50}$$

$$X_3 = X_4 = X_{14} = X_{15}, \tag{3.2.51}$$

and

$$X_6 = X_7 = X_{10} = X_{11}. \tag{3.2.52}$$

If the Mathematics behind are more deeply analized, it can be shown that the equivalency comes from the symmetries created and that all the matrices with the same determinant have the same result at the end of the fourth transmission.

For instance,

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} = \begin{bmatrix} A & B^* \\ B & -A^* \end{bmatrix} \tag{3.2.53}$$

and

$$S_{16} = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ -s_2 & s_1^* & -s_4^* & s_3 \\ -s_3 & -s_4^* & s_1^* & s_2 \\ -s_4 & s_3^* & s_2^* & -s_1 \end{bmatrix} = \begin{bmatrix} -A & -B^* \\ -B & A^* \end{bmatrix} \tag{3.2.54}$$

where

$$\det(S_1) = -AA^*-BB^* \quad \text{and} \quad \det(S_{16}) = -AA^*-BB^* \qquad (3.2.55)$$

So, as it can be seen operating by blocks, $\det(S_1) = \det(S_{16})$, therefore $X_1 = X_{16}$.

With Matlab, it is calculated: $\det(S_8)=$ (3.2.56)

$s_1^2(s_1^*)^2 + 2s_1s_2(s_1^*)(s_2^*) - (s_4^*)^2s_1^2+2(s_2^*)s_1(s_4^*)s_3 + 2s_1(s_3^*)(s_4^*)s_2 + 2s_1(s_1^*)(s_3^*)s_3 + (s_2^*)^2s_2^2 + 2s_2(s_4^*)s_4(s_2^*) - (s_3^*)^2s_2^2 + 2s_4(s_3^*)s_2(s_1^*) - s_3^2(s_2^*)^2 + 2s_3(s_1^*)s_4(s_2^*) + (s_3^*)^2s_3^2 + 2(s_3^*)(s_4^*)s_3s_4 - s_4^2(s_1^*)^2 + s_4^2(s_4^*)^2$

but it can be actually noticed that:

$$\det(S_1) = \det(S_{16}) = \det(S_8) = \det(S_9) = \det(S_6) = \det(S_7) = \det(S_{10}) = \det(S_{11})$$

$$\neq \det(S_{2,3,4,5,12,13,14,15}), \qquad (3.2.57)$$

whereas

$$\det(S_2) = \det(S_5) = \det(S_{12}) = \det(S_{13}) = \det(S_3) = \det(S_4) = \det(S_{14}) = \det(S_{15})$$

$$\neq \det(S_{1,6,7,8,9,10,11,16}). \qquad (3.2.58)$$

This result is crucial because it shows that in fact, at the end, there are only two alternatives!!! The rest of combinations are somehow related to the original couple.

3.2.2 Simulation Results

Let's have a look to the BER perfromance of the independents figures for S_6, S_7, S_8, S_9 compared to the election between the best of them for only R=2, which is already going to give an idea of their equivalency:

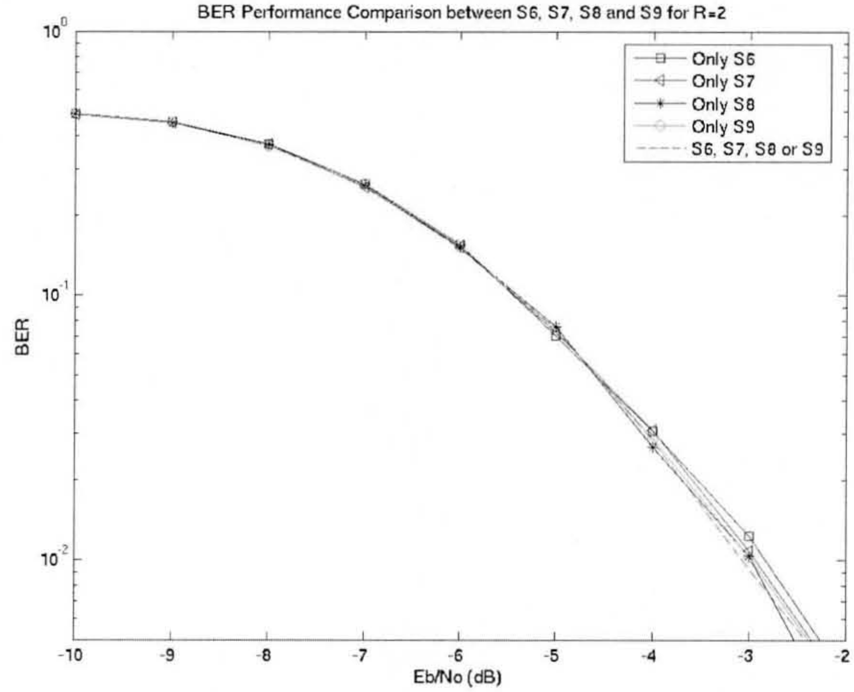

Figure 3.11 BER performance comparison between S6, S7, S8 and S9 for R=2.

It can be observed how all of them have nearly the same performance, only a little bit different around BER = 10^{-2} because of the number of samples (5000) in the Montecarlo simulation.

Comparing them with the figures in the former section, where it was analized Only S_1, Only S_2, and the proposed S_1 or S_2, it can be shown that there is no improvement, because they are equivalent, so in conclusion, the selection is not among the 16 alternatives but only between two of them (for example S_1 or S_2).

Thinking about the reason why this happens, it's pretty logical because the interference in the non-main diagonal will be the same in absolute value for all those equivalent cases. It doesn't really matter if X_i is positive or negative, since they both appear in different places.

3.3 Extended Alamouti with Permutations of Alternative Matrices

In this section a couple of permutations of Alternative Matrices termed S_{1B} and S_{1C} are analyzed, still for the invariant fading channel.

3.3.1 Analytical Description

The structure of these new permutations is the following:

$$S_{1B} = \begin{bmatrix} s_1 & s_4^* & s_3^* & s_2 \\ s_2 & -s_3^* & s_4^* & -s_1 \\ s_3 & s_2^* & -s_1^* & -s_4 \\ s_4 & -s_1^* & -s_2^* & s_3 \end{bmatrix} \tag{3.3.1}$$

$$S_{1C} = \begin{bmatrix} s_1 & s_3^* & s_3^* & -s_1 \\ s_2 & -s_4^* & s_4^* & s_2 \\ s_3 & -s_1^* & -s_1^* & -s_3 \\ s_4 & s_2^* & -s_2^* & s_4 \end{bmatrix} \tag{3.3.2}$$

Let's start with S_{1B} remembering how, for a 4x4 antennas MIMO system, the received signal can be modeled with:

$$r^{(1)} = H\, s^{(1)} + n^{(1)} \tag{3.3.3}$$

After matched filtering at the receiver,

$$x^{(1)} = H^{\dagger} H\, s^{(1)} + H^{\dagger} n^{(1)} = C\, s^{(1)} + H^{\dagger} n^{(1)} \tag{3.3.4}$$

where $(\bullet)^{\dagger}$ is the operation of conjugate transpose; $C = H^{\dagger} H$.

At the second instant, the signal $s^{(2)} = \begin{bmatrix} s_4^* \\ -s_3^* \\ s_2^* \\ -s_1^* \end{bmatrix}$ is transmitted, hence

$$r^{(2)} = H\, s^{(2)} + n^{(2)} = H\, \mathcal{J}_1\, s^{(1)*} + n^{(2)} \tag{3.3.5}$$

where

$$\mathcal{J}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \quad (3.3.6)$$

Therefore,

$$r^{(2)*} = H^* \mathcal{J}_1 s^{(1)} + n^{(2)*} \quad (3.3.7)$$

After the matched filter, multiplying again by $(H^*\mathcal{J}_1)^\dagger = \mathcal{J}_1^T H^T$

$$x^{(2)} = \mathcal{J}_1^T H^T H^* \mathcal{J}_1 \ s^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} \quad (3.3.8)$$

Combining $x^{(1)}$ and $x^{(2)}$,

$$\hat{s}_{1,2} = (C + \mathcal{J}_1^T C^* \mathcal{J}_1)\, s + H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} \quad (3.3.9)$$

Then,

$$\mathcal{J}_1^T C^* \mathcal{J}_1 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* & a_{14}^* \\ a_{12} & a_{22} & a_{23}^* & a_{24}^* \\ a_{13} & a_{23} & a_{33} & a_{34}^* \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \quad (3.3.10a)$$

$$= \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -a_{14}^* & a_{13}^* & -a_{12}^* & a_{11} \\ -a_{24}^* & a_{23}^* & -a_{22} & a_{12} \\ -a_{34}^* & a_{33} & -a_{23} & a_{13} \\ -a_{44} & a_{34} & -a_{24} & a_{14} \end{bmatrix} \quad (3.3.10b)$$

$$= \begin{bmatrix} a_{44} & -a_{34} & a_{24} & -a_{14} \\ -a_{34}^* & a_{33} & -a_{23} & a_{13} \\ a_{24}^* & -a_{23}^* & a_{22} & -a_{12} \\ -a_{14}^* & a_{13}^* & -a_{12}^* & a_{11} \end{bmatrix} \quad (3.3.10c)$$

As resulting matrix,

$$C_x = C + \mathcal{J}_1^T C^* \mathcal{J}_1 = \begin{bmatrix} a_{11}+a_{44} & a_{12}-a_{34} & a_{13}+a_{24} & 0 \\ a_{12}^*-a_{34}^* & a_{22}+a_{33} & 0 & a_{24}+a_{13} \\ a_{13}^*+a_{24}^* & 0 & a_{33}+a_{22} & a_{34}-a_{12} \\ 0 & a_{24}^*+a_{13}^* & a_{34}^*-a_{12}^* & a_{44}+a_{11} \end{bmatrix} \tag{3.3.11}$$

For the third transmission,

$$s^{(3)} = \begin{bmatrix} s_3^* \\ s_4^* \\ -s_1^* \\ -s_2^* \end{bmatrix} = \mathcal{J}_2\, s^{(1)*} \tag{3.3.12}$$

with

$$\mathcal{J}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{3.3.13}$$

Then, after matched filtering

$$x^{(3)} = \mathcal{J}_2^T H^T H^* \mathcal{J}_2\, s^{(1)} + \mathcal{J}_2^T H^T n^{(3)*} \tag{3.3.14}$$

Combining $x^{(3)}$,

$$\hat{s}_{1,2,3} = (C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2)s + H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} + \mathcal{J}_2^T H^T n^{(3)*} \tag{3.3.15}$$

in which

$$\mathcal{J}_2^T C^* \mathcal{J}_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* & a_{14}^* \\ a_{12} & a_{22} & a_{23}^* & a_{24}^* \\ a_{13} & a_{23} & a_{33} & a_{34}^* \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{3.3.16a}$$

$$= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -a_{13}^* & -a_{14}^* & a_{11} & a_{12}^* \\ -a_{23}^* & -a_{24}^* & a_{12} & a_{22} \\ -a_{33} & -a_{34}^* & a_{13} & a_{23} \\ -a_{34} & -a_{44} & a_{14} & a_{24} \end{bmatrix} \tag{3.3.16b}$$

$$= \begin{bmatrix} a_{33} & a_{34}^* & -a_{13} & -a_{23} \\ a_{34} & a_{44} & -a_{14} & -a_{24} \\ -a_{13}^* & -a_{14}^* & a_{11} & a_{12}^* \\ -a_{23}^* & -a_{24}^* & a_{12} & a_{22} \end{bmatrix} \qquad (3.3.16c)$$

Therefore,

$$C_X = C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2 \qquad (3.3.17a)$$

$$= \begin{bmatrix} a_{11} + a_{33} + a_{44} & a_{12} - a_{34} + a_{34}^* & a_{24} & -a_{23} \\ a_{12}^* + a_{34} - a_{34}^* & a_{22} + a_{33} + a_{44} & -a_{14} & a_{13} \\ a_{24}^* & -a_{14}^* & a_{11} + a_{33} + a_{22} & a_{34} + a_{12}^* - a_{12} \\ -a_{23}^* & a_{13}^* & a_{34}^* + a_{12} - a_{12}^* & a_{44} + a_{22} + a_{11} \end{bmatrix} \qquad (3.3.17b)$$

At the fourth transmission,

$$s^{(4)} = \begin{bmatrix} s_2 \\ -s_1 \\ -s_4 \\ s_3 \end{bmatrix} = \mathcal{J}_3 \, s^{(1)} \qquad (3.3.18)$$

with

$$\mathcal{J}_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (3.3.19)$$

However, now

$$r^{(4)} = H \, \mathcal{J}_3 \, s^{(1)} + n^{(4)} \qquad (3.3.20)$$

After the matched filter, multiplying by $(H \mathcal{J}_3) = \mathcal{J}_3^T H$

$$x^{(4)} = \mathcal{J}_2^T H^{\dagger} H \; \mathcal{J}_2 \, s^{(1)} + \mathcal{J}_3^T H^{\dagger} \, n^{(4)} \qquad (3.3.21)$$

and combining $x^{(4)}$,

$$\hat{s}_{1,2,3,4} = (C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2 + \mathcal{J}_3^T C \, \mathcal{J}_3) \, s + H^{\dagger} n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} +$$

$$+ \mathcal{J}_2^T H^T n^{(3)*} + \mathcal{J}_3^T H^{\dagger} \, n^{(4)} \qquad (3.3.22)$$

Since

$$\mathcal{J}_3^{\mathrm{T}}\,\mathrm{C}\,\mathcal{J}_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.3.23a}$$

$$= \begin{bmatrix} a_{22} & -a_{12}^* & -a_{24} & a_{23} \\ -a_{12} & a_{11} & a_{14} & -a_{13} \\ -a_{24}^* & a_{14}^* & a_{44} & -a_{34}^* \\ a_{23}^* & -a_{13}^* & -a_{34} & a_{33} \end{bmatrix} \tag{3.3.23b}$$

the resulting matrix is

$$\mathrm{C_X} = \mathrm{C} + \mathcal{J}_1^{\mathrm{T}}\,\mathrm{C}^*\,\mathcal{J}_1 + \mathcal{J}_2^{\mathrm{T}}\,\mathrm{C}^*\,\mathcal{J}_2 + \mathcal{J}_3^{\mathrm{T}}\,\mathrm{C}\,\mathcal{J}_3 \tag{3.3.24a}$$

$$= \begin{bmatrix} a_{11}+a_{22}+a_{33}+a_{44} & a_{12}-a_{12}^*-a_{34}+a_{34}^* & 0 & 0 \\ a_{12}^*-a_{12}+a_{34}-a_{34}^* & a_{11}+a_{22}+a_{33}+a_{44} & 0 & 0 \\ 0 & 0 & a_{11}+a_{22}+a_{33}+a_{44} & a_{12}^*-a_{12}+a_{34}-a_{34}^* \\ 0 & 0 & a_{12}-a_{12}^*-a_{34}+a_{34}^* & a_{11}+a_{22}+a_{33}+a_{44} \end{bmatrix}$$

That is

$$\mathrm{C_{X1B}} = \mathrm{A} \begin{bmatrix} 1 & X_{1B} & 0 & 0 \\ -X_{1B} & 1 & 0 & 0 \\ 0 & 0 & 1 & -X_{1B} \\ 0 & 0 & X_{1B} & 1 \end{bmatrix} \tag{3.3.24b}$$

with

$$\mathrm{A} = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.25}$$

and

$$X_{1\mathrm{B}} = \frac{a_{12} - a_{12}^* - a_{34} + a_{34}^*}{A} \tag{3.3.26}$$

So, for S_{1B} there is a very similar structure as seen so far, with a nearly orthogonal matrix at the end of the cycle, but the interference X_{1B} is different (both in situation and components) in this case.

For S_{1C}, at the second instant, the signal $s^{(2)} = \begin{bmatrix} s_3^* \\ -s_4^* \\ -s_1^* \\ s_2^* \end{bmatrix}$ is transmitted, hence

$$r^{(2)} = H\, s^{(2)} + n^{(2)} = H\, \mathcal{J}_1\, s^{(1)*} + n^{(2)} \tag{3.3.27}$$

where

$$\mathcal{J}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.3.28}$$

Therefore,

$$r^{(2)*} = H^* \mathcal{J}_1\, s^{(1)} + n^{(2)*} \tag{3.3.29}$$

After the matched filter, multiplying by $(H^*\mathcal{J}_1)^\dagger = \mathcal{J}_1^T H^T$

$$x^{(2)} = \mathcal{J}_1^T H^T H^*\, \mathcal{J}_1\, s^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} \tag{3.3.30}$$

Combining $x^{(1)}$ and $x^{(2)}$,

$$\hat{s}_{1,2} = (C + \mathcal{J}_1^T C^*\, \mathcal{J}_1)\, s + H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} \tag{3.3.31}$$

Then,

$$\mathcal{J}_1^T C^* \mathcal{J}_1 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* & a_{14}^* \\ a_{12} & a_{22} & a_{23}^* & a_{24}^* \\ a_{13} & a_{23} & a_{33} & a_{34}^* \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.3.32a}$$

$$= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -a_{13}^* & a_{14}^* & a_{11} & -a_{12}^* \\ -a_{23}^* & a_{24}^* & a_{12} & -a_{22} \\ -a_{33} & a_{34}^* & a_{13} & -a_{23} \\ -a_{34} & a_{44} & a_{14} & -a_{24} \end{bmatrix} \tag{3.3.32b}$$

$$= \begin{bmatrix} a_{33} & -a_{34}^* & -a_{13} & a_{23} \\ -a_{34} & a_{44} & a_{14} & -a_{24} \\ -a_{13}^* & a_{14}^* & a_{11} & -a_{12}^* \\ a_{23}^* & -a_{24}^* & -a_{12} & a_{22} \end{bmatrix} \tag{3.3.32c}$$

so as resulting matrix,

$$C_X = C + \mathcal{J}_1^T C^* \mathcal{J}_1 = \begin{bmatrix} a_{11}+a_{33} & a_{12}-a_{34}^* & 0 & a_{14}+a_{23} \\ a_{12}^*-a_{34} & a_{22}+a_{44} & a_{23}+a_{14} & 0 \\ 0 & a_{23}^*+a_{14}^* & a_{33}+a_{11} & a_{34}-a_{12}^* \\ a_{23}^*+a_{14}^* & 0 & a_{34}^*-a_{12} & a_{44}+a_{22} \end{bmatrix} \tag{3.3.33}$$

If the third transmission is needed,

$$s^{(3)} = \begin{bmatrix} s_3^* \\ s_4^* \\ -s_1^* \\ -s_2^* \end{bmatrix} = \mathcal{J}_2\, s^{(1)*} \tag{3.3.34}$$

with

$$\mathcal{J}_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{3.3.35}$$

Then,

$$x^{(3)} = \mathcal{J}_2^T H^T H^* \mathcal{J}_2\, s^{(1)} + \mathcal{J}_2^T H^T n^{(3)*} \tag{3.3.36}$$

and combining,

$$\hat{s}_{1,2,3} = (C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2)s + H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} + \mathcal{J}_2^T H^T n^{(3)*} \tag{3.3.37}$$

in which

$$\mathcal{J}_2^T C^* \mathcal{J}_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* & a_{14}^* \\ a_{12} & a_{22} & a_{23}^* & a_{24}^* \\ a_{13} & a_{23} & a_{33} & a_{34}^* \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \tag{3.3.38a}$$

$$= \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -a_{13}^* & -a_{14}^* & a_{11} & a_{12}^* \\ -a_{23}^* & -a_{24}^* & a_{12} & a_{22} \\ -a_{33} & -a_{34}^* & a_{13} & a_{23} \\ -a_{34} & -a_{44} & a_{14} & a_{24} \end{bmatrix} \tag{3.3.38b}$$

$$= \begin{bmatrix} a_{33} & a_{34}^* & -a_{13} & -a_{23} \\ a_{34} & a_{44} & -a_{14} & -a_{24} \\ -a_{13}^* & -a_{14}^* & a_{11} & a_{12}^* \\ -a_{23}^* & -a_{24}^* & a_{12} & a_{22} \end{bmatrix} \tag{3.3.38c}$$

Therefore,

$$C_X = C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2$$

$$= \begin{bmatrix} a_{11} + a_{33} + a_{33} & a_{12} & -a_{13} & a_{14} \\ a_{12}^* & a_{22} + a_{44} + a_{44} & a_{23} & -a_{24} \\ -a_{13}^* & a_{23}^* & a_{11} + a_{33} + a_{11} & a_{34} \\ a_{14}^* & -a_{24}^* & a_{34}^* & a_{44} + a_{22} + a_{22} \end{bmatrix} \tag{3.3.39}$$

At the fourth transmission,

$$s^{(4)} = \begin{bmatrix} -s_1 \\ s_2 \\ -s_3 \\ s_4 \end{bmatrix} = \mathcal{J}_3 \, s^{(1)} \tag{3.3.40}$$

with

$$\mathcal{J}_3 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3.41}$$

However, now

$$r^{(4)} = H \, \mathcal{J}_3 \, s^{(1)} + n^{(4)} \tag{3.3.42}$$

After the matched filter, multiplying by $(H \mathcal{J}_3) = \mathcal{J}_3^T H$

$$x^{(4)} = \mathcal{J}_2^T H^\dagger H \;\mathcal{J}_2 \, s^{(1)} + \mathcal{J}_3^T H^\dagger n^{(4)} \tag{3.3.43}$$

and combining $x^{(4)}$,

$$\hat{s}_{1,2,3,4} = (C + \mathcal{J}_1^T C^* \mathcal{J}_1 + \mathcal{J}_2^T C^* \mathcal{J}_2 + \mathcal{J}_3^T C \, \mathcal{J}_3) \, s +$$

$$+ H^\dagger n^{(1)} + \mathcal{J}_1^T H^T n^{(2)*} + \mathcal{J}_2^T H^T n^{(3)*} + \mathcal{J}_3^T H^\dagger n^{(4)} \tag{3.3.44}$$

Since

$$\mathcal{J}_3^{T} C\, \mathcal{J}_3 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^{*} & a_{22} & a_{23} & a_{24} \\ a_{13}^{*} & a_{23}^{*} & a_{33} & a_{34} \\ a_{14}^{*} & a_{24}^{*} & a_{34}^{*} & a_{44} \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.3.45a}$$

$$= \begin{bmatrix} a_{11} & -a_{12} & a_{13} & -a_{14} \\ -a_{12}^{*} & a_{22} & -a_{23} & a_{24} \\ a_{13}^{*} & -a_{23}^{*} & a_{33} & -a_{34} \\ -a_{14}^{*} & a_{24}^{*} & -a_{34}^{*} & a_{44} \end{bmatrix} \tag{3.3.45b}$$

the resulting matrix is

$$C_{X1C} = C + \mathcal{J}_1^{T} C^{*} \mathcal{J}_1 + \mathcal{J}_2^{T} C^{*} \mathcal{J}_2 + \mathcal{J}_3^{T} C\, \mathcal{J}_3 \tag{3.3.46a}$$

$$= \begin{bmatrix} a_{11}+a_{11}+a_{33}+a_{33} & 0 & 0 & 0 \\ 0 & a_{22}+a_{22}+a_{44}+a_{44} & 0 & 0 \\ 0 & 0 & a_{33}+a_{33}+a_{11}+a_{11} & 0 \\ 0 & 0 & 0 & a_{44}+a_{44}+a_{22}+a_{22} \end{bmatrix}$$

$$= \begin{bmatrix} A_{13} & 0 & 0 & 0 \\ 0 & A_{24} & 0 & 0 \\ 0 & 0 & A_{13} & 0 \\ 0 & 0 & 0 & A_{24} \end{bmatrix} \tag{3.3.46b}$$

with

$$A_{13} = a_{11} + a_{11} + a_{33} + a_{33} \tag{3.3.47}$$

and

$$A_{24} = a_{22} + a_{22} + a_{44} + a_{44} \tag{3.3.48}$$

In that case, the result is a perfectly orthogonal matrix at the end of the cycle!!! However, let's see the results after the simulations, compared to the performance of the original S_1.

3.3.2 Simulation Results

As it can be seen in Figure 3.12, the result for the orthogonal matrix (S_{1C}) is surprisingly worse than for the others (only nearly orthogonal), so it doesn't make any sense to take it into account cause it can't improve the system.

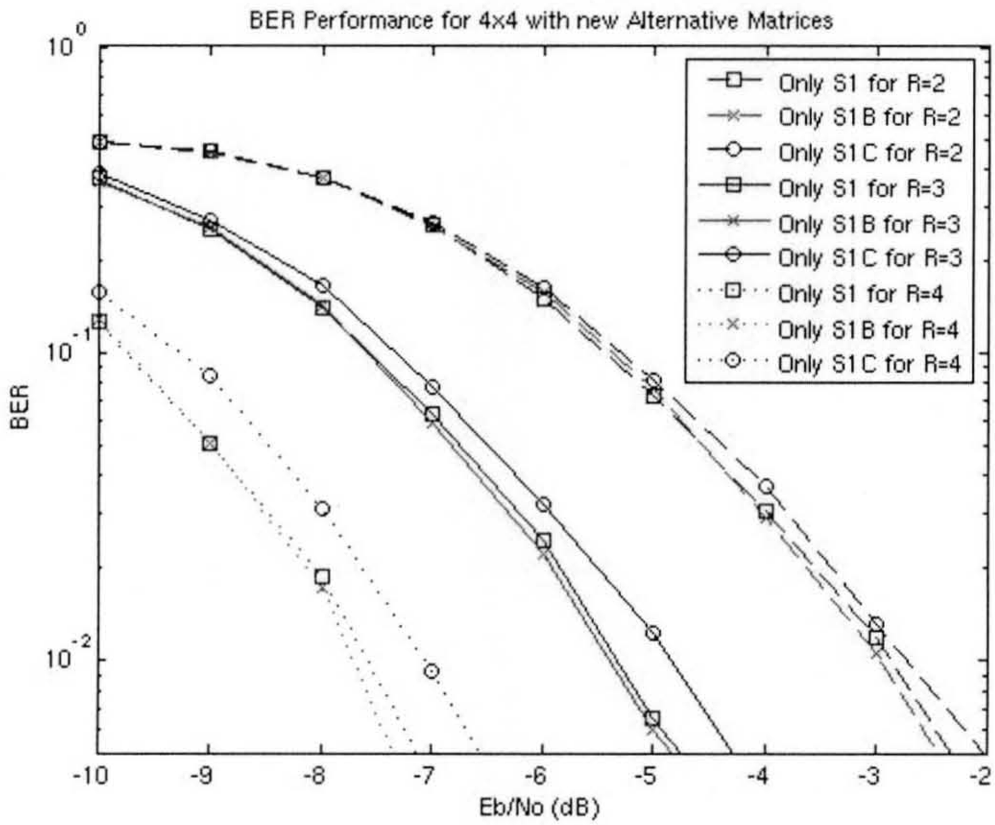

Figure 3.12 BER performance for 4x4 with new Alternative Matrices.

The reason why it has such a bad performance might be that we are sending twice the same information (with only sign changes) from the same antennas, which can be seen at the end of the cycle in (3.3.46b) through the main diagonal.

3.3.3 Extension from the LG Scheme

These two previous alternatives (S_{1B} and S_{1C}) may look pretty similar to the odd alternatives 2 and 3 in the LG Electronics proposal [9]:

$$S_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ -s_4^* \\ s_1^* \\ s_{21}^* \end{bmatrix} \quad \text{and} \quad S_{ALT3}^{(odd)} = \begin{bmatrix} -s_4^* \\ -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix} \tag{3.3.49}$$

However, it can be shown that it's not the same concept when the Extended Alamouti Block Coding is used. For instance, if the odd alternative 3 is taken as the second transmission (or first retransmission), it can be figured out that there actually is another alternative!!! Both the interference and the determinant will be different, so it can be considered as another option to send at the beginning of the retransmissions:

$$S_{1B}' = \begin{bmatrix} s_1 & -s_4^* & s_3^* & s_2 \\ s_2 & -s_3^* & s_4^* & s_1 \\ s_3 & s_2^* & -s_1^* & s_4 \\ s_4 & s_1^* & -s_2^* & s_3 \end{bmatrix} \tag{3.3.50}$$

The resulting matrix is:

$$C_X = C + J_1^T C^* J_1 + J_2^T C^* J_2 + J_3^T C J_3 \tag{3.3.51a}$$

$$= \begin{bmatrix} a_{11}+a_{22}+a_{33}+a_{44} & a_{12}+a_{12}^*+a_{34}+a_{34}^* & 0 & 0 \\ a_{12}^*+a_{12}+a_{34}+a_{34}^* & a_{11}+a_{22}+a_{33}+a_{44} & 0 & 0 \\ 0 & 0 & a_{11}+a_{22}+a_{33}+a_{44} & a_{12}^*+a_{12}+a_{34}+a_{34}^* \\ 0 & 0 & a_{12}+a_{12}^*+a_{34}+a_{34}^* & a_{11}+a_{22}+a_{33}+a_{44} \end{bmatrix}$$

That is

$$C_{X1B}' = A \begin{bmatrix} 1 & X_{1B}' & 0 & 0 \\ X_{1B}' & 1 & 0 & 0 \\ 0 & 0 & 1 & X_{1B}' \\ 0 & 0 & X_{1B}' & 1 \end{bmatrix} \tag{3.3.51b}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.52}$$

and

$$X_{1B}' = \frac{a_{12} + a_{12}^* + a_{34} + a_{34}^*}{A} \tag{3.3.53}$$

On the other hand, if the odd alternative 2 is taken, it will be seen right away that it has the same problem as our alternative S_{1C}, i.e. the same information is sent twice, so the BER will not improve that much:

$$S_{1C}' = \begin{bmatrix} s_1 & -s_3^* & s_3^* & s_1 \\ s_2 & -s_4^* & s_4^* & s_2 \\ s_3 & s_1^* & -s_1^* & s_3 \\ s_4 & s_2^* & -s_2^* & s_4 \end{bmatrix} \tag{3.3.54}$$

Let's remember how the Extended Alamouti block coding is based on the extension of the 2x2 matrices (alamoutisaton):

$$S = \begin{bmatrix} A & -B^* \\ B & A^* \end{bmatrix} \quad \text{or the equivalent} \quad S = \begin{bmatrix} A & B^* \\ B & -A^* \end{bmatrix} \quad \text{as in (2.2)}$$

If these two previous alternatives are analized, it can be shown how the concept of the Extended Alamouti Block Coding is actually being distorsioned, because the real ones should be:

$$S_{1B}'' = \begin{bmatrix} s_1 & s_4^* & s_2^* & s_3 \\ s_4 & -s_1^* & s_3^* & -s_2 \\ s_2 & s_3^* & -s_1^* & -s_4 \\ s_3 & -s_2^* & -s_4^* & s_1 \end{bmatrix} \quad \text{and} \quad S_{1C}'' = \begin{bmatrix} s_1 & s_3^* & s_2^* & s_4 \\ s_3 & -s_1^* & s_4^* & -s_2 \\ s_2 & s_4^* & -s_1^* & -s_3 \\ s_4 & -s_2^* & -s_3^* & s_1 \end{bmatrix}$$

Reorganizing them, it gives:

$$S_{1B}''' = \begin{bmatrix} s_1 & s_4^* & s_2^* & s_3 \\ s_2 & s_3^* & -s_1^* & -s_4 \\ s_3 & -s_2^* & -s_4^* & s_1 \\ s_4 & -s_1^* & s_3^* & -s_2 \end{bmatrix} \quad \text{and} \quad S_{1C}''' = \begin{bmatrix} s_1 & s_3^* & s_2^* & s_4 \\ s_2 & s_4^* & -s_1^* & -s_3 \\ s_3 & -s_1^* & s_4^* & -s_2 \\ s_4 & -s_2^* & -s_3^* & s_1 \end{bmatrix}$$

which apparently has not an Extended Alamouti structure but at the end of the cycle there is still a nearly othogonal matrix. The problem comes when analyzing their development, cause while S_{1B}''' has surprisingly different interferences than S_{1B}'' but still the same determinant (so they are actually equivalents); in fact, S_{1C}''' has not only the same C_X matrix as S_{1C}'' but also completely the same as S_1, seen in Chapter 2!!!

Visually,

$$C_{X1B}'' = A \begin{bmatrix} 1 & 0 & X_{1B}'' & 0 \\ 0 & 1 & 0 & -X_{1B}'' \\ X_{1B}'' & 0 & 1 & 0 \\ 0 & -X_{1B}'' & 0 & 1 \end{bmatrix} \tag{3.3.55}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.56}$$

and

$$X_{1B}'' = \frac{a_{14} + a_{14}^* - a_{23} - a_{23}^*}{A} \tag{3.3.57}$$

while

$$C_{X1B}''' = A \begin{bmatrix} 1 & 0 & X_{1B}''' & 0 \\ 0 & 1 & 0 & -X_{1B}''' \\ X_{1B}''' & 0 & 1 & 0 \\ 0 & -X_{1B}''' & 0 & 1 \end{bmatrix} \tag{3.3.58}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.59}$$

and

$$X_{1B}''' = \frac{a_{13} + a_{13}^* - a_{24} - a_{24}^*}{A} \tag{3.3.60}$$

On the other hand,

$$C_{X1C}'' = A\begin{bmatrix} 1 & 0 & 0 & X_{1C}'' \\ 0 & 1 & -X_{1C}'' & 0 \\ 0 & -X_{1C}'' & 1 & 0 \\ X_{1C}'' & 0 & 0 & 1 \end{bmatrix} \tag{3.3.61}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.62}$$

and

$$X_{1C}'' = \frac{a_{14} + a_{14}^{*} - a_{23} - a_{23}^{*}}{A} \tag{3.3.63}$$

while

$$C_{X1C}''' = A\begin{bmatrix} 1 & 0 & 0 & X_{1C}''' \\ 0 & 1 & -X_{1C}''' & 0 \\ 0 & -X_{1C}''' & 1 & 0 \\ X_{1C}''' & 0 & 0 & 1 \end{bmatrix} = C_X \tag{3.3.64}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.65}$$

and

$$X_{1C}''' = \frac{a_{14} + a_{14}^{*} - a_{23} - a_{23}^{*}}{A} = X_{1C}'' = X_1 \tag{3.3.66}$$

Due to the structure of the Extended Alamouti Block Coding there are a lot of equivalencies, so not all the combinations are absolutely needed to optimize the system. In the last case, it can actually be seen how the symmetry comes from the simple switch of the two middle columns of the matrices ($s^{(2)} = s^{(3)'}$ and $s^{(3)} = s^{(2)'}$), that is:

$$\begin{matrix} & s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} \\ S_1 = & s_1 & s_2^{*} & s_3^{*} & s_4 \\ & s_2 & -s_1^{*} & s_4^{*} & -s_3 \\ & s_3 & s_4^{*} & -s_1^{*} & -s_2 \\ & s_4 & -s_3^{*} & -s_2^{*} & s_1 \end{matrix} \quad \Leftrightarrow \quad \begin{matrix} & s^{(1)'} & s^{(2)'} & s^{(3)'} & s^{(4)'} \\ S_{1C}''' = & s_1 & s_3^{*} & s_2^{*} & s_4 \\ & s_2 & s_4^{*} & -s_1^{*} & -s_3 \\ & s_3 & -s_1^{*} & s_4^{*} & -s_2 \\ & s_4 & -s_2^{*} & -s_3^{*} & s_1 \end{matrix}$$

Looking at the interferences X_i, it can be shown that there are still two permutations of Altenative Matrices missing:

$$S_{1A} = \begin{bmatrix} s_1 & s_3^* & s_2^* & s_4 \\ s_2 & -s_4^* & -s_1^* & s_3 \\ s_3 & -s_1^* & s_4^* & -s_2 \\ s_4 & s_2^* & -s_3^* & -s_1 \end{bmatrix} \quad \text{and} \quad S_{1D} = \begin{bmatrix} s_1 & s_2^* & s_4^* & s_3 \\ s_2 & -s_1^* & s_3^* & -s_4 \\ s_3 & s_4^* & -s_2^* & -s_1 \\ s_4 & -s_3^* & -s_1^* & s_2 \end{bmatrix}$$

In the first case,

$$C_{X1A} = A \begin{bmatrix} 1 & 0 & 0 & -X_{1A} \\ 0 & 1 & -X_{1A} & 0 \\ 0 & X_{1A} & 1 & 0 \\ X_{1A} & 0 & 0 & 1 \end{bmatrix} \tag{3.3.67}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.68}$$

and

$$X_{1A} = \frac{a_{14} + a_{14}^* - a_{23} - a_{23}^*}{A} \tag{3.3.69}$$

In the second case,

$$C_{X1D} = A \begin{bmatrix} 1 & 0 & -X_{1D} & 0 \\ 0 & 1 & 0 & X_{1D} \\ X_{1D} & 0 & 1 & 0 \\ 0 & -X_{1D} & 0 & 1 \end{bmatrix} \tag{3.3.70}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.3.71}$$

and

$$X_{1D} = \frac{-a_{13} + a_{13}^* + a_{24} - a_{24}^*}{A} \tag{3.3.72}$$

Note that the situation of the negative interferences is not exactly the same as in C_{X1} and C_{X1B}''' (matrices 2.28b and 3.3.54, respectively).

3.4 Ext. Alamouti with Sign Changes in Permutations of Alternative Matrices

Right now, besides S_1 and S_2, there are five more alternatives: S_{1B}, S_{1B}', S_{1B}''', S_{1A} and S_{1D}. As it was done with the two original ones, the possible sign changes in their first raws should result in five more.

3.4.1 Analytical Description

From S_{1B}, after changing the sign of the first raw, we have:

$$S_{2B} = \begin{array}{c} \begin{matrix} s^{(1)'} & s^{(2)'} & s^{(3)'} & s^{(4)'} \end{matrix} \\ \begin{bmatrix} -s_1 & -s_4^* & -s_3^* & -s_2 \\ s_2 & -s_3^* & s_4^* & -s_1 \\ s_3 & s_2^* & -s_1^* & -s_4 \\ s_4 & -s_1^* & -s_2^* & s_3 \end{bmatrix} \end{array} \tag{3.4.1}$$

That is

$$r^{(1)} = H\, s^{(1)'} + n^{(1)} \tag{3.4.2a}$$

$$= H\, \mathcal{J}_0\, s^{(1)} + n^{(1)} \tag{3.4.2b}$$

where

$$\mathcal{J}_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4.3}$$

So, after matched filtering at the receiver,

$$x^{(1)} = \mathcal{J}_0^T H^\dagger H\, \mathcal{J}_0\, s^{(1)} + \mathcal{J}_0^T H^\dagger n^{(1)} \tag{3.4.4a}$$

$$= \mathcal{J}_0^T C\, \mathcal{J}_0\, s^{(1)} + \mathcal{J}_0^T H^\dagger n^{(1)} \tag{3.4.4b}$$

At the second instant, the signal $s^{(2)'} = \begin{bmatrix} -s_4^* \\ -s_3^* \\ s_2^* \\ -s_1^* \end{bmatrix}$ is transmitted, hence

$$r^{(2)} = H\, s^{(2)'} + n^{(2)} = H\, \mathcal{J}_1\, s^{(1)*} + n^{(2)} \qquad (3.4.5)$$

where

$$\mathcal{J}_1 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \qquad (3.4.6)$$

Therefore,

$$r^{(2)*} = H^*\, \mathcal{J}_1\, s^{(1)} + n^{(2)*} \qquad (3.4.7)$$

and after matched filtering,

$$x^{(2)} = \mathcal{J}_1^{\mathsf{T}}\, H^{\mathsf{T}} H^*\, \mathcal{J}_1\, s^{(1)} + \mathcal{J}_1^{\mathsf{T}}\, H^{\mathsf{T}}\, n^{(2)*} \qquad (3.4.8)$$

So adding both terms to get the output for decision,

$$\hat{s} = (\mathcal{J}_0^{\mathsf{T}}\, C\, \mathcal{J}_0 + \mathcal{J}_1^{\mathsf{T}}\, C^*\, \mathcal{J}_1)\, s^{(1)} + \mathcal{J}_0^{\mathsf{T}}\, H^{\dagger}\, n^{(1)} + \mathcal{J}_1^{\mathsf{T}}\, H^{\mathsf{T}}\, n^{(2)*} \qquad (3.4.9)$$

Now,

$$\mathcal{J}_0^{\mathsf{T}}\, C\, \mathcal{J}_0 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.4.10a)$$

$$= \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -a_{11} & -a_{12} & -a_{13} & -a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \qquad (3.4.10b)$$

$$= \begin{bmatrix} a_{11} & -a_{12} & -a_{13} & -a_{14} \\ -a_{12}^* & a_{22} & a_{23} & a_{24} \\ -a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ -a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \qquad (3.4.10c)$$

$$\mathcal{J}_1^{\mathrm{T}}\, \mathrm{C}^*\, \mathcal{J}_1 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a^*_{12} & a^*_{13} & a^*_{14} \\ a_{12} & a_{22} & a^*_{23} & a^*_{24} \\ a_{13} & a_{23} & a_{33} & a^*_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \qquad (3.4.11a)$$

$$= \begin{bmatrix} a_{44} & -a_{34} & a_{24} & a_{14} \\ -a^*_{34} & a_{33} & -a_{23} & -a^*_{13} \\ a^*_{24} & -a^*_{23} & a_{22} & a_{12} \\ a^*_{14} & -a_{13} & a^*_{12} & a_{11} \end{bmatrix} \qquad (3.4.11b)$$

$$\mathcal{J}_0^{\mathrm{T}}\, \mathrm{C}\, \mathcal{J}_0 + \mathcal{J}_1^{\mathrm{T}}\, \mathrm{C}^*\, \mathcal{J}_1 = \begin{bmatrix} a_{11} & -a_{12} & -a_{13} & -a_{14} \\ -a^*_{12} & a_{22} & a_{23} & a_{24} \\ -a^*_{13} & a^*_{23} & a_{33} & a_{34} \\ -a^*_{14} & a^*_{24} & a^*_{34} & a_{44} \end{bmatrix} + \begin{bmatrix} a_{44} & -a_{34} & a_{24} & a_{14} \\ -a^*_{34} & a_{33} & -a_{23} & -a^*_{13} \\ a^*_{24} & -a^*_{23} & a_{22} & a_{12} \\ a^*_{14} & -a_{13} & a^*_{12} & a_{11} \end{bmatrix} \qquad (3.4.12a)$$

$$= \begin{bmatrix} a_{11} + a_{44} & -a_{12} - a_{34} & a_{24} - a_{13} & 0 \\ -a^*_{12} - a^*_{34} & a_{22} + a_{33} & 0 & a_{24} - a^*_{13} \\ a^*_{24} - a^*_{13} & 0 & a_{33} + a_{22} & a_{12} + a_{23} \\ 0 & a^*_{24} - a_{13} & a^*_{12} + a^*_{34} & a_{44} + a_{11} \end{bmatrix} \qquad (3.4.12b)$$

Similarly,

$$\mathrm{C_X} = \mathcal{J}_0^{\mathrm{T}}\, \mathrm{C}\, \mathcal{J}_0 + \mathcal{J}_1^{\mathrm{T}}\, \mathrm{C}^*\, \mathcal{J}_1 + \mathcal{J}_2^{\mathrm{T}}\, \mathrm{C}^*\, \mathcal{J}_2 \qquad (3.4.13a)$$

$$= \begin{bmatrix} a_{11} + a_{33} + a_{44} & -a_{12} - a_{34} + a^*_{34} & a_{24} & -a_{23} \\ a_{34} - a^*_{12} - a^*_{34} & a_{11} + a_{22} + a_{44} & a_{14} & -a_{13} \\ a^*_{24} & a^*_{14} & a_{11} + a_{22} + a_{33} & a_{12} - a^*_{12} + a_{34} \\ -a^*_{23} & -a^*_{13} & a^*_{12} - a_{12} + a^*_{314} & a_{22} + a_{33} + a_{44} \end{bmatrix} \qquad (3.4.13b)$$

and finally,

$$\mathrm{C_{X2B}} = \mathcal{J}_0^{\mathrm{T}}\, \mathrm{C}\, \mathcal{J}_0 + \mathcal{J}_1^{\mathrm{T}}\, \mathrm{C}^*\, \mathcal{J}_1 + \mathcal{J}_2^{\mathrm{T}}\, \mathrm{C}^*\, \mathcal{J}_2 + \mathcal{J}_3^{\mathrm{T}}\, \mathrm{C}\, \mathcal{J}_3 \qquad (3.4.14)$$

$$= \begin{bmatrix} a_{11} + a_{22} + a_{33} + a_{44} & a^*_{12} - a_{12} + a^*_{34} - a_{34} & 0 & 0 \\ a_{12} - a^*_{12} + a_{34} - a^*_{34} & a_{11} + a_{22} + a_{33} + a_{44} & 0 & 0 \\ 0 & 0 & a_{11} + a_{22} + a_{33} + a_{44} & a_{12} - a^*_{12} + a_{34} - a^*_{34} \\ 0 & 0 & a^*_{12} - a_{12} + a^*_{34} - a_{34} & a_{11} + a_{22} + a_{33} + a_{44} \end{bmatrix}$$

where

$$J_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \qquad (3.4.15)$$

and

$$J_3 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (3.4.16)$$

So, at the end the C_{X2B} matrix is:

$$C_{X2B} = A\begin{bmatrix} 1 & X_{2B} & 0 & 0 \\ -X_{2B} & 1 & 0 & 0 \\ 0 & 0 & 1 & -X_{2B} \\ 0 & 0 & X_{2B} & 1 \end{bmatrix} \qquad (3.4.17)$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \qquad (3.4.18)$$

and

$$X_{2B} = \frac{-a_{12} + a_{12}^{*} - a_{34} + a_{34}^{*}}{A} \qquad (3.4.19)$$

Doing all that steps again, it can also be found out for S_{2B}', S_{2B}''', S_{2A} and S_{2D}:

$$C_{X2B}' = A\begin{bmatrix} 1 & X_{2B}' & 0 & 0 \\ X_{2B}' & 1 & 0 & 0 \\ 0 & 0 & 1 & X_{2B}' \\ 0 & 0 & X_{2B}' & 1 \end{bmatrix} \qquad (3.4.20)$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \qquad (3.4.21)$$

and

$$X_{2B}' = \frac{-a_{12} - a_{12}^{*} + a_{34} + a_{34}^{*}}{A} \qquad (3.4.22)$$

$$C_{X2B}''' = A\begin{bmatrix} 1 & 0 & X_{2B}''' & 0 \\ 0 & 1 & 0 & -X_{2B}''' \\ X_{2B}''' & 0 & 1 & 0 \\ 0 & -X_{2B}''' & 0 & 1 \end{bmatrix} \tag{3.4.23}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.4.24}$$

and

$$X_{2B}''' = \frac{-a_{13} - a_{13}^{*} - a_{24} - a_{24}^{*}}{A} \tag{3.4.25}$$

$$C_{X2A} = A\begin{bmatrix} 1 & 0 & 0 & -X_{1A} \\ 0 & 1 & -X_{1A} & 0 \\ 0 & X_{1A} & 1 & 0 \\ X_{1A} & 0 & 0 & 1 \end{bmatrix} \tag{3.4.26}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.4.27}$$

and

$$X_{2A} = \frac{a_{14} + a_{14}^{*} - a_{23} - a_{23}^{*}}{A} \tag{3.4.28}$$

Finally,

$$C_{X2D} = A\begin{bmatrix} 1 & 0 & -X_{1D} & 0 \\ 0 & 1 & 0 & X_{1D} \\ X_{1D} & 0 & 1 & 0 \\ 0 & -X_{1D} & 0 & 1 \end{bmatrix} \tag{3.4.29}$$

with

$$A = a_{11} + a_{22} + a_{33} + a_{44} \tag{3.4.30}$$

and

$$X_{2D} = \frac{-a_{13} + a_{13}^{*} + a_{24} - a_{24}^{*}}{A} \tag{3.4.31}$$

In summary, the 12 final alternative matrices are:

$$S_1 = \begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad C_{X1} = A\begin{bmatrix} 1 & 0 & 0 & X_1 \\ 0 & 1 & -X_1 & 0 \\ 0 & -X_1 & 1 & 0 \\ X_1 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{with} \quad X_1 = \frac{a_{14} + a_{14}^* - a_{23} - a_{23}^*}{A}$$

$$S_2 = \begin{bmatrix} -s_1 & -s_2^* & -s_3^* & -s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} \qquad C_{X2} = A\begin{bmatrix} 1 & 0 & 0 & X_2 \\ 0 & 1 & -X_2 & 0 \\ 0 & -X_2 & 1 & 0 \\ X_2 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{with} \quad X_2 = \frac{-a_{14} - a_{14}^* - a_{23} - a_{23}^*}{A}$$

$$S_{1A} = \begin{bmatrix} s_1 & s_3^* & s_2^* & s_4 \\ s_2 & -s_4^* & -s_1^* & s_3 \\ s_3 & -s_1^* & s_4^* & -s_2 \\ s_4 & s_2^* & -s_3^* & -s_1 \end{bmatrix} \qquad C_{X1A} = A\begin{bmatrix} 1 & 0 & 0 & -X_{1A} \\ 0 & 1 & -X_{1A} & 0 \\ 0 & X_{1A} & 1 & 0 \\ X_{1A} & 0 & 0 & 1 \end{bmatrix}$$

$$\text{with} \quad X_{1A} = \frac{-a_{14} + a_{14}^* - a_{23} + a_{23}^*}{A}$$

$$S_{2A} = \begin{bmatrix} -s_1 & -s_3^* & -s_2^* & -s_4 \\ s_2 & -s_4^* & -s_1^* & s_3 \\ s_3 & -s_1^* & s_4^* & -s_2 \\ s_4 & s_2^* & -s_3^* & -s_1 \end{bmatrix} \qquad C_{X2A} = A\begin{bmatrix} 1 & 0 & 0 & -X_{2A} \\ 0 & 1 & -X_{2A} & 0 \\ 0 & X_{2A} & 1 & 0 \\ X_{2A} & 0 & 0 & 1 \end{bmatrix}$$

$$\text{with} \quad X_{2A} = \frac{a_{14} - a_{14}^* - a_{23} + a_{23}^*}{A}$$

$$S_{1B} = \begin{bmatrix} s_1 & s_4^* & s_3^* & s_2 \\ s_2 & -s_3^* & s_4^* & -s_1 \\ s_3 & s_2^* & -s_1^* & -s_4 \\ s_4 & -s_1^* & -s_2^* & s_3 \end{bmatrix} \qquad C_{X1B} = A\begin{bmatrix} 1 & X_{1B} & 0 & 0 \\ -X_{1B} & 1 & 0 & 0 \\ 0 & 0 & 1 & -X_{1B} \\ 0 & 0 & X_{1B} & 1 \end{bmatrix}$$

with $X_{1B} = \dfrac{a_{12} - a_{12}^* - a_{34} + a_{34}^*}{A}$

$$S_{2B} = \begin{bmatrix} -s_1 & -s_4^* & -s_3^* & -s_2 \\ s_2 & -s_3^* & s_4^* & -s_1 \\ s_3 & s_2^* & -s_1^* & -s_4 \\ s_4 & -s_1^* & -s_2^* & s_3 \end{bmatrix} \qquad C_{X2B} = A\begin{bmatrix} 1 & X_{2B} & 0 & 0 \\ -X_{2B} & 1 & 0 & 0 \\ 0 & 0 & 1 & -X_{2B} \\ 0 & 0 & X_{2B} & 1 \end{bmatrix}$$

with $X_{2B} = \dfrac{-a_{12} + a_{12}^* - a_{34} + a_{34}^*}{A}$

$$S_{1B}' = \begin{bmatrix} s_1 & -s_4^* & s_3^* & s_2 \\ s_2 & -s_3^* & s_4^* & s_1 \\ s_3 & s_2^* & -s_1^* & s_4 \\ s_4 & s_1^* & -s_2^* & s_3 \end{bmatrix} \qquad C_{X1B}' = A\begin{bmatrix} 1 & X_{1B}' & 0 & 0 \\ X_{1B}' & 1 & 0 & 0 \\ 0 & 0 & 1 & X_{1B}' \\ 0 & 0 & X_{1B}' & 1 \end{bmatrix}$$

with $X_{1B}' = \dfrac{a_{12} + a_{12}^* + a_{34} + a_{34}^*}{A}$

$$S_{2B}' = \begin{bmatrix} -s_1 & s_4^* & -s_3^* & -s_2 \\ s_2 & -s_3^* & s_4^* & s_1 \\ s_3 & s_2^* & -s_1^* & s_4 \\ s_4 & s_1^* & -s_2^* & s_3 \end{bmatrix} \qquad C_{X2B}' = A\begin{bmatrix} 1 & X_{2B}' & 0 & 0 \\ X_{2B}' & 1 & 0 & 0 \\ 0 & 0 & 1 & X_{2B}' \\ 0 & 0 & X_{2B}' & 1 \end{bmatrix}$$

with $X_{2B}' = \dfrac{-a_{12} - a_{12}^* + a_{34} + a_{34}^*}{A}$

$$S_{1B}''' = \begin{bmatrix} s_1 & s_4^* & s_2^* & s_3 \\ s_2 & s_3^* & -s_1^* & -s_4 \\ s_3 & -s_2^* & -s_4^* & s_1 \\ s_4 & -s_1^* & s_3^* & -s_2 \end{bmatrix} \qquad C_{X1B}''' = A \begin{bmatrix} 1 & 0 & X_{1B}''' & 0 \\ 0 & 1 & 0 & -X_{1B}''' \\ X_{1B}''' & 0 & 1 & 0 \\ 0 & -X_{1B}''' & 0 & 1 \end{bmatrix}$$

with $X_{1B}''' = \dfrac{a_{13} + a_{13}^* - a_{24} - a_{24}^*}{A}$

$$S_{2B}''' = \begin{bmatrix} -s_1 & -s_4^* & -s_2^* & -s_3 \\ s_2 & s_3^* & -s_1^* & -s_4 \\ s_3 & -s_2^* & -s_4^* & s_1 \\ s_4 & -s_1^* & s_3^* & -s_2 \end{bmatrix} \qquad C_{X2B}''' = A \begin{bmatrix} 1 & 0 & X_{2B}''' & 0 \\ 0 & 1 & 0 & -X_{2B}''' \\ X_{2B}''' & 0 & 1 & 0 \\ 0 & -X_{2B}''' & 0 & 1 \end{bmatrix}$$

with $X_{2B}''' = \dfrac{-a_{13} - a_{13}^* - a_{24} - a_{24}^*}{A}$

$$S_{1D} = \begin{bmatrix} s_1 & s_4^* & s_2^* & s_3 \\ s_2 & s_3^* & -s_1^* & -s_4 \\ s_3 & -s_2^* & -s_4^* & s_1 \\ s_4 & -s_1^* & s_3^* & -s_2 \end{bmatrix} \qquad C_{X1D} = A \begin{bmatrix} 1 & 0 & -X_{1D} & 0 \\ 0 & 1 & 0 & X_{1D} \\ X_{1D} & 0 & 1 & 0 \\ 0 & -X_{1D} & 0 & 1 \end{bmatrix}$$

with $X_{1D} = \dfrac{-a_{13} + a_{13}^* + a_{24} - a_{24}^*}{A}$

$$S_{2D} = \begin{bmatrix} -s_1 & -s_4^* & -s_2^* & -s_3 \\ s_2 & s_3^* & -s_1^* & -s_4 \\ s_3 & -s_2^* & -s_4^* & s_1 \\ s_4 & -s_1^* & s_3^* & -s_2 \end{bmatrix} \qquad C_{X2D} = A \begin{bmatrix} 1 & 0 & -X_{2D} & 0 \\ 0 & 1 & 0 & X_{2D} \\ X_{2D} & 0 & 1 & 0 \\ 0 & -X_{2D} & 0 & 1 \end{bmatrix}$$

with $X_{2D} = \dfrac{a_{13} - a_{13}^* + a_{24} - a_{24}^*}{A}$

3.4.2 Simulation Results

After studying all the options and combinations, the final choice can be done among 12 alternative matrices, and although a few of them may have the same initial transmissions, they all end up with different interferences at the end of the cycle. Therefore, since the channel response is known and assumed constant, the best matrix for transmission will be established in the beginning of the simulation, optimizing the SNR with the same Determinant Criterion that was used in [1] to find the appropiate order of retransmissions.

In Figure 3.13, the BER performance comparison between the LG solution, the Alternative Matrix (seen in Section 3.1) and the Final Proposed scheme is depicted:

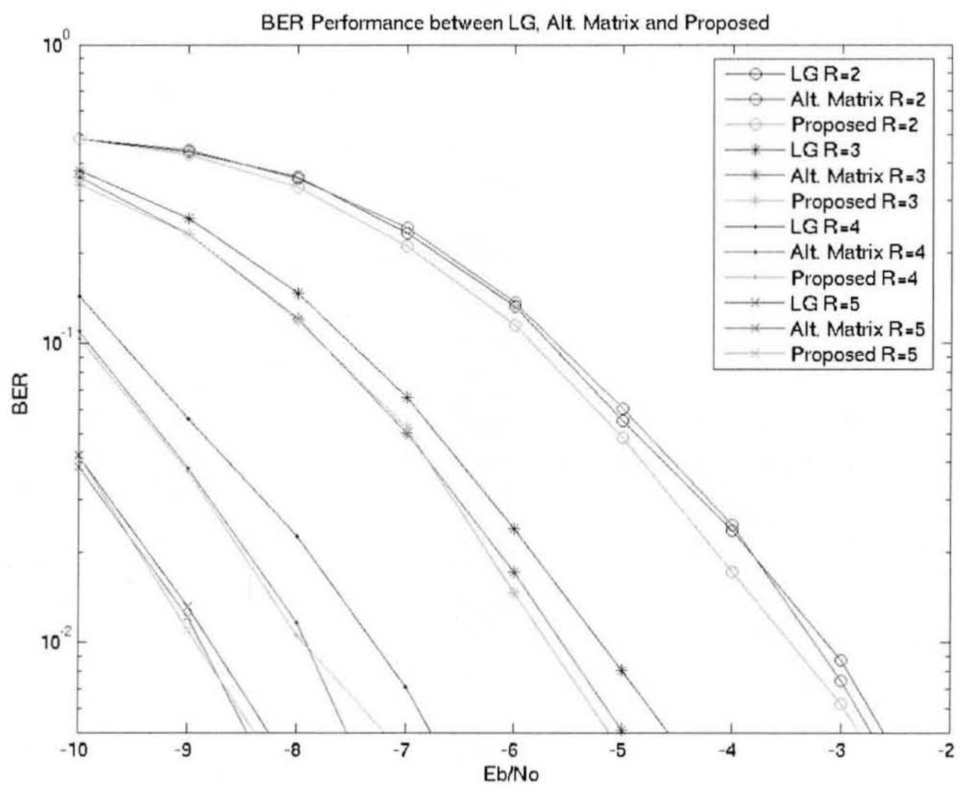

Figure 3.13 BER performance comparison between LG, Alternative Matrix and the final Proposed scheme for Invariant Channel.

In Figure 3.14, the Throughput Performance Comparison between the LG solution, the Alternative Matrix (seen in Section 3.1) and the Final Proposed scheme is depicted:

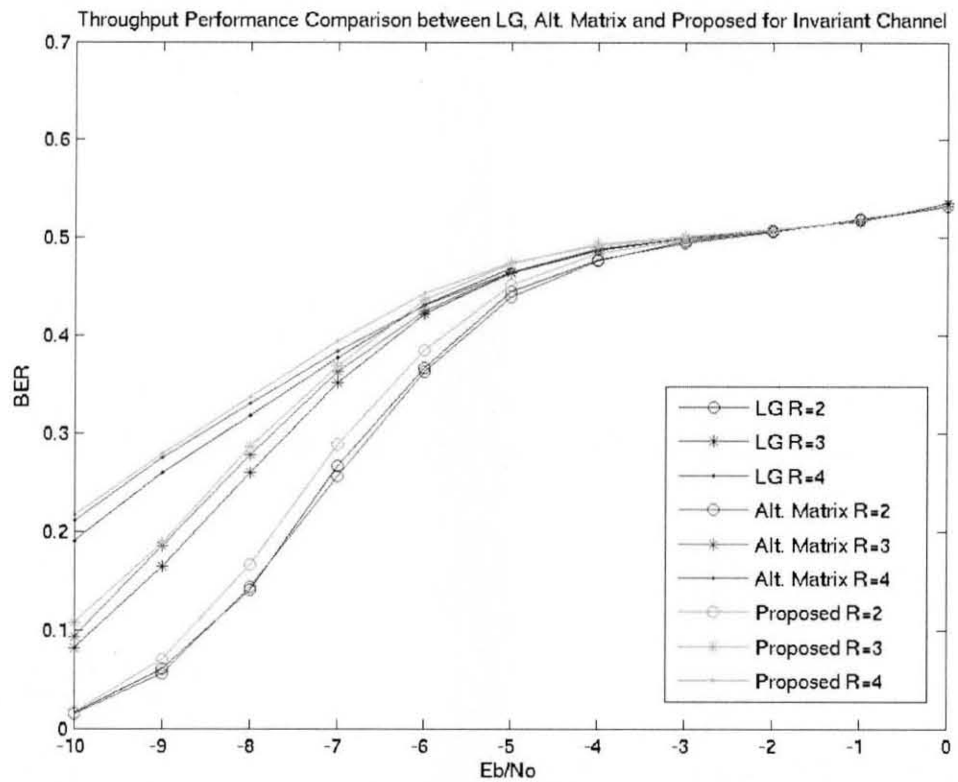

Figure 3.14 Throughput performance comparison between LG, Altenative Matrix and the final Proposed scheme for Invariant Channel.

The conclusion is that the Proposed scheme "beats" the former solutions (both in BER and Throughput Performance) in all the stages, especially in R=2 where the Alternative Matrix couldn't perform better than LG, which is also a great improvement because the Multiple Alamouti Coding was clearly worse in a 4x4 MIMO system.

CHAPTER 4

EXTENDED ALAMOUTI-BASED HARQ SCHEME FOR A 4X4 MIMO SYSTEM IN TIME VARYING CHANNEL CONDITIONS

In the previous chapters, the BER and Throughput Performance of the proposed Extended Alamouti scheme with Alternative Matrices have been analized assuming the channel was time invariant; i.e., its coefficients were constant during all retransmissionns. However, this is not realistic because a channel may change in a very short time. Let's now assume that the channel response only remains constant during a packet transmission, changing with some correlation for the next one. This chapter will show how that affects to the proposed scheme, compared once again with the LG solution.

4.1 System Model

Even though there are plenty of models to characterize a time varying channel in the literature [12-13], for simplicity, the Auto Regressive of order 1 (AR-1), as in [1], is chosen. To create the channel for simulation, a random matrix H_1 with 4 by 4 i.i.d. complex Gaussian Random Variables (Raleigh Flat Fading) with unit power is generated:

$$H_1 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \quad (4.1)$$

where

$$E\{|h_{ji}|^2\} = 1 \text{ , } E\{h_{ji}h_{ml}^*\} = 0 \text{ and } i,l = 1...4 \ \ j,m = 1...4 \text{ with } i \neq l \text{ or } j \neq m$$

(with $E\{\ \}$: the expected value)

The AR-1 model has the following discrete low pass expression:

$$h_{ji}^{k+1} = -a_1 h_{ji}^{k} + w_{ji} \tag{4.2}$$

where a_1 is a tap filter, w_{ji} is a complex Gaussian noise with power σ_W^2 and k is the transmission packet index.

To find the values of a_1 and the σ_W^2 for a given correlation the Yule-Walker equations [14] have to be solved:

$$\begin{bmatrix} R_h(0) & R_h^*(1) \\ R_h(1) & R_h(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sigma_W^2 \\ 0 \end{bmatrix} \tag{4.3}$$

where $R_h(0)$ and $R_h(1)$ are the values of the correlation between samples of successive channels.

Since, different channel gains are assumed uncorrelated and with unit variance, then from (4.1.3)

$$R_h(0) = 1 \quad \text{and} \quad R_h(1) = -a_1 \tag{4.4}$$

Note that the first value keep the power normalization to 1, and the second defines how correlated is the channel with the previous ones. Clearly $a_1 = 1$ means that the new channel is the same as the previous one, and $a_1 = 0$ means that the new channel is completely uncorrelated with the previous one.

As it can be easily shown from (4.1.3), normalizing the power of the new channel coefficients is not necessary:

$$\mathrm{E}\left\{ \left| h_{ij}^{k+1} \right|^2 \right\} = |a_1|^2 + \sigma_w^2 = 1 \tag{4.5}$$

Among the two possible approaches seen in [1] to study a Time Varying channel, this thesis will directly follow the best one, which is called Modified Retransmission Order Algorithm without Channel Modification. Since the channel changes for each retransmission, the idea is to estimate the current channel response through the previous one, thanks to the known correlation parameters.

$$\tilde{C}_2 = E\{H_2^{\Psi} H_2\} = \mathrm{E}\{(a_1 H_1 + W)^{\Psi}(a_1 H_1 + W)\} \tag{4.6a}$$

$$= a_1^2 H_1^{\Psi} H_1 + E\{W^{\Psi} W\} = a_1^2 C_1 + 4\sigma_w^2 \mathrm{I} \tag{4.6b}$$

where I is a $4x4$ identity matrix and W is the noise matrix of the AR-1 model with $4x4$ elements.

Then, instead of only using the Determinant Criterion at the beginning of the cycle, an algorithm checks the best option every time that a transmission is needed, including the initial one. The estimation of the first Channel Response might come from the known Channel State Information in some feedback bits after a training signal is sent.

4.2 Simulation Results

Table 4.1 Simulation parameters and values for Time Varying Channel conditions.

PARAMETER	VALUES
Number of samples (n)	5000 (enough for 10^{-2})
Maximum number of transmissions (R)	2, 3, 4 (complete cylcle)
Packet size (bits)	522 (splitable by 2,3,4,5 and 6)
Tranmitter(M)xReceiver(N) antennas	4x4
Eb/No (dB)	-10 to 0
a1 / σ_W^2	-0.9 / 0.19 and -0.5 / 0.75

In this previous Table 4.1 all the parameters for the simulations in time varying channel conditions are summarized.

The following figures compare the performance of the Final Proposed scheme based on Extended Alamouti using Alternative Matrices with the modified determinant algorithm for the case of a time varying channel in a $4x4$ MIMO system. The Figure 4.1 shows the BER versus E_b / N_0 when the channel at each retransmission is quite correlated with the previous one, with $a_1 = 0.9$. In Figure 4.2, a channel more varying in time ($a_1 = 0.5$) is used.

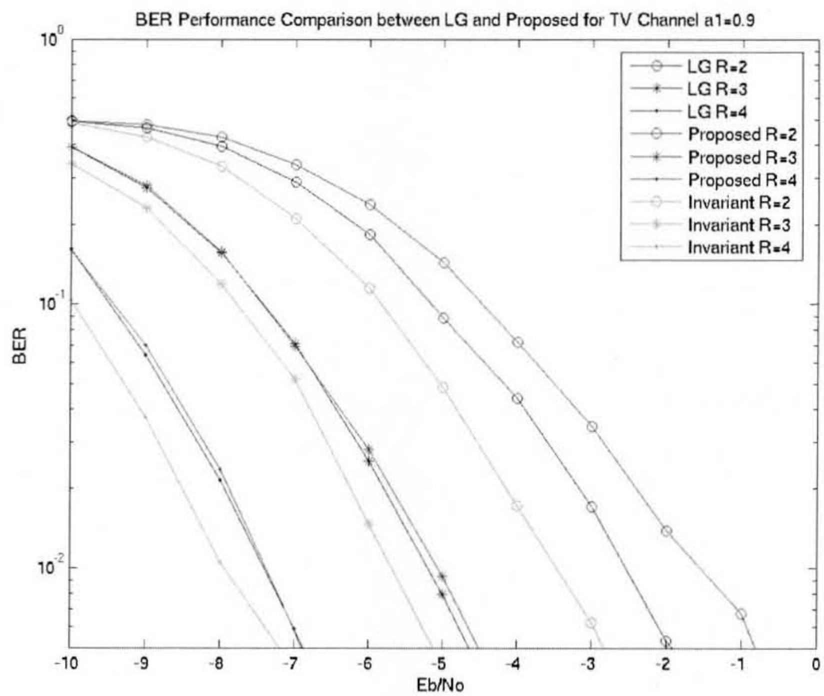

Figure 4.1 BER performance comparison between LG and Proposed schemes for Time Variant Channel and $a_1 = 0.9$.

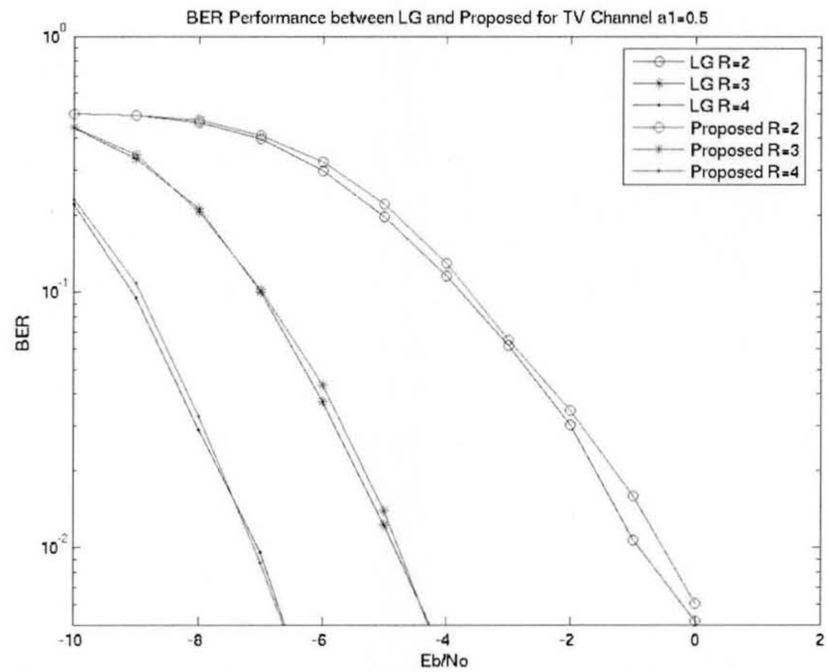

Figure 4.2 BER performance comparison between LG and Proposed schemes for Time Variant Channel and $a_1 = 0.5$.

The results are pretty disappointing compared to the Invariant Channel situation, because in this case, the LG solution still performs better than the Final Proposed scheme based on Extended Alamouti Coding using Alternative Matrices. In fact, for a quite uncorrelated channel ($a_1 = 0.5$) they are both pretty similar, but for a nearly correlated channel ($a_1 = 0.9$) LG outperforms in R=2, although still far from the ideal Invariant Channel case. As it can be seen in the following Figures 4.3 and 4.4, the Throughput analysis is parallel to the BER performance. The reason why this happens might be the extra estimation that the proposed scheme needs at the beginning to decide between the initial transmission with all positive signals or with the first one negative (Alternative Matrix).

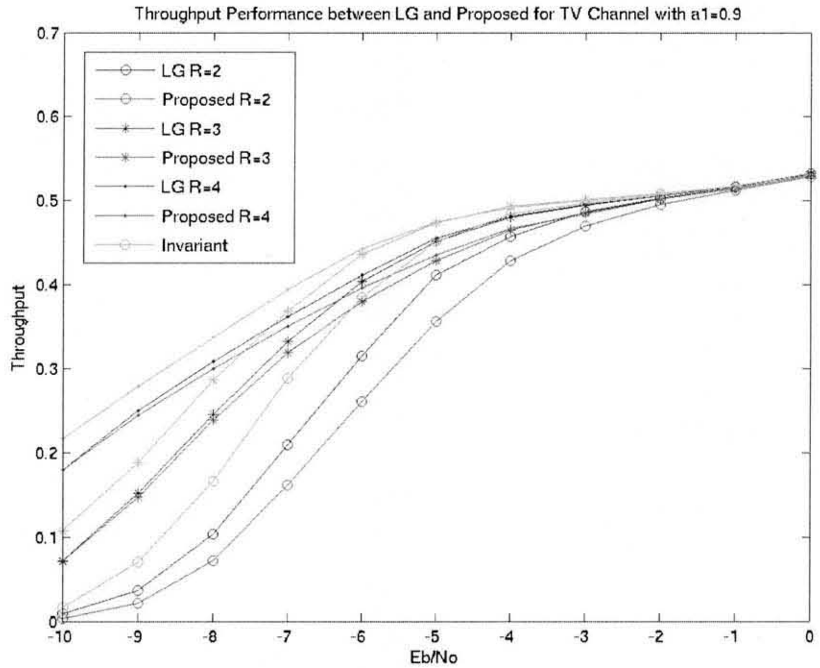

Figure 4.3 Throughput performance comparison between LG and Proposed schemes for Time Variant Channel with $a_1 = 0.9$.

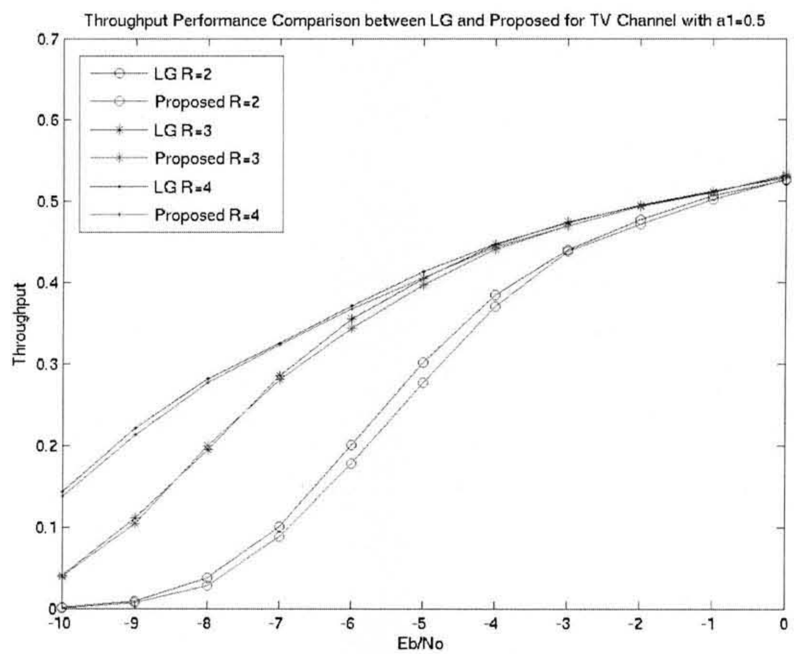

Figure 4.4 Throughput performance comparison between LG and Proposed schemes for Time Variant Channel with $a_1 = 0.5$.

CHAPTER 5

SUMMARY AND SUGGESTIONS FOR FUTURE WORK

5.1 Summary of the Thesis

In MIMO systems, HARQ is a promising and deeply investigated topic to improve the quality and increase the capacity of wireless communications exploiting their diversity.

In Chapter 2, the Quasi-orthogonal STBC called Extended Alamouti was shown as a good starting point to achieve a balanced trade-off between interference cancellation and transmission rate.

Introducing an Alternative Matrix with a sign change in Section 3.1 nearly accomplished the goal of finding a scheme with the best BER and Throughput performance (not yet for R=2).

Although some equivalents showed up during the research, a final proposal composed by 12 permutations of Alternative Matrices at the end of Chapter 3 ended up "beating" (not more than 0.5dB though) any former solution for invariant fading channel conditions.

However, in a time varying situation as described in Chapter 4, the previous LG's scheme was still better exploiting its efficiency, at least under the Determinant Criterion, the best one found so far to select the retransmission sequence.

Let's point out just a small drawback of the proposed scheme, since it definitely needs some bits of feedback from the receiver: in this case at least 4 to code all the combinations.

5.2 Suggestions for Future Work

As future topics for research where there's still a lot to investigate two ideas are basically suggested:

- Regarding the feedback issue that was just mentioned, a complicated world in the adaptive communications remains opened. For instance, with some more bits, the receiver could also specify to the transmitter what particular packet in a determined antenna was successfully decoded, so that in the next retransmision the free spot can already be used for another signal.

- Even though the thesis has beeen focused in a 4x4 MIMO system, the extension to MxM antennas (with $M=2^n$, n>2) is straight forward because we keep the block symmetries. On the other hand, when M is different the problem is not trivial at all, even less if the structure is asymmetric (MxN, with M≠N). As an example, in the case of 5x6 trying a variation of Multiple Alamouti Coding is suggested; that is, sending two basic Alamouti Codes and a zero in the fifth antenna remaining:

$$\begin{array}{cccccc} s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} & s^{(5)} & s^{(6)} \end{array}$$

$$\begin{bmatrix} s_1 & s_2^* & s_3^* & s_5^* & s_4^* & 0 \\ s_2 & -s_1^* & s_5^* & s_4^* & 0 & s_2^* \\ s_3 & s_4^* & -s_1^* & 0 & s_5^* & s_2^* \\ s_4 & -s_3^* & 0 & -s_2^* & -s_1^* & s_2^* \\ s_5 & 0 & -s_2^* & -s_1^* & -s_3^* & s_2^* \end{bmatrix}$$

APPENDIX

MATLAB SOURCE CODES

Main Program for Time Invariant (Extended Alamouti)

```
function [Result]=program();
%n is the number of packets
%R is the number of repetitions
%N is the number of transmitting antennas (Maximum 6)
%M is the number of receiveing antennas
%M must be equal or higher than N
n=5000;
R=4;
packet_size=522; %with this size we can split the packet in 2,3,4,5 and 6 parts
N=4;
M=4;
seed=69; %seed for the initiall state in the function rand
SNR=[-10:1:0]; %SNR in dB
L=length(SNR);
Result=ones(2,L);%we save the BER and the throuhput for each value of SNR

filename = ['HARQ_Extended_alamouti',num2str(N),'x',num2str(M),'_R',num2str(R)];
for(i=1:L)
    i
    Result(:,i)=HARQ(n,R,SNR(i),packet_size,floor(seed*rand),N,M)';
end;
save (filename,'Result','n');
return;
```

Main Program for Time Variant (All Alternative Matrices)

```
function [Result]=program();
%n is the number of packets
%R is the number of repetitions
%N is the number of transmitting antennas (Maximum 6)
%M is the number of receiveing antennas
%M must be equal or higher than N
n=5000;
R=5;
packet_size=522; %with this size we can split the packet in 2,3,4,5 and 6 parts
N=4;
M=4;
seed=45; %seed for the initiall state in the function rand
SNR=[-10:1:-4]; %SNR in dB
L=length(SNR);
Result=ones(2,L);%we save the BER and the throuhput for each value of SNR
a1=-0.9;
sigma=0.19;
a=9;
filename =
['HARQ_Extended_AltSAllVariant',num2str(N),'x',num2str(M),'_R',num2str(R),'_a',num2str(a)
];
for(i=1:L)
    i

Result(:,i)=HARQ_altSAllVariant(n,R,SNR(i),packet_size,floor(seed*rand),N,M,a1,sigma)';
end;
save (filename,'Result','n');
return;
```

HARQ function for Time Invariant (Extended Alamouti)

```
function [Result]=HARQ(n,R,SNR,packet_size,state,N,M);
%this function return the probability of error and the thorughput
%n is the number of packets of the simulation
%R is the maximum number of repetitions of the data
%SNR is the signal to noise ratio
%packet_size is the size of the packet.
%state is the seed for the function randn
%N is the number of transmitter antennas
%M is the number of receiver antennas
%M must be equal or higher than N
%The variable random says if we use the algorihm or not


error=0; %this variable counts the total number of bits errors
sent_packets=n; %this variable counts the total number of packets that we send.
Initially, equal to n
lost_packet=0; %this varible counts the total number of packets that we lose
%I=Alamouti_Generator(N); %we generate a matrix which contents the Alamouti Matrix for
each sequence of repetition
%P=1+factorial(N)/(factorial(N-2)*2); %this variable gives the number of vectors in the
Alamouti process
P=4;
S=packet(n,packet_size); %we create a matrix with n packets. Each packet is composed by
Info+CRC+Trellis Code modulation
randn('state',state); %we put the seed in the function randn

for(i=1:n) %the simulation starts...
    H=sqrt(0.5)*randn(M,N)+j*sqrt(0.5)*randn(M,N);%we create a matrix with iid complex
gaussian parameters for the channel
    r=0; %this variable counts the current repetition
    ack=1; %this variable tells us if the packet is correct or not
    error_packet=0; %this variable counts the number of error bits in a packet
    v=modulation(S(:,i)); %we get the QPSK signal from each packet
    L=length(v); %L must be divisible by N
    V=split(v,N,L); %we split the packet in N equal subpackets and we put in a matrix of
size Nx(L/N)
    Vest=0*ones(N,L/N); %estimated vector at the receiver Nx(L/N)
    noise=sqrt((10^((-6-SNR)/10))/2)*randn(M,L/N)+j*sqrt((10^((-6-
SNR)/10))/2)*randn(M,L/N); %noise matrix MxL/N
    C0=H'*H;
    C=zeros(4,4);
    while((r<R)&&(ack~=0))%while the packet still have errors and we have still more
repetitions
        y=mod(r,P); %this variable tells us which number of the sequence we are running
in the Alamouti
        switch (y)
            case 0
                x=H'*H*V+H'*noise;
                C=C+C0;
                [A,conjugate]=decisor(C,C0); %this function returns the matrix I with the
best order for transmission
                %break;
            case 1
                x=A(1:4,:)'*conj(H')*conj(H)*A(1:4,:)*V+A(1:4,:)'*conj(H')*conj(noise);
                C=C+A(1:4,:)'*conj(C0)*A(1:4,:);
                %break;
            case 2
                if(conjugate==0)
                    x=A(5:8,:)'*conj(H')*conj(H)*A(5:8,:)*V+A(5:8,:)'*conj(H')*noise;
                    C=C+A(5:8,:)'*conj(C0)*A(5:8,:);
                else
                    x=A(5:8,:)'*H'*H*A(5:8,:)*V+A(5:8,:)'*H'*noise;
                    C=C+A(5:8,:)'*C0*A(5:8,:);
                end;
                %break;
            case 3
```

```
                    if(conjugate==1)
                        x=A(9:12,:)'*conj(H')*conj(H)*A(9:12,:)*V+A(9:12,:)'*conj(H')*noise;
                        C=C+A(9:12,:)'*conj(C0)*A(9:12,:);
                    else
                        x=A(9:12,:)'*H'*H*A(9:12,:)*V+A(9:12,:)'*H'*noise;
                        C=C+A(9:12,:)'*C0*A(9:12,:);
                    end;
                    %break;
        end;
        Vest=Vest+x; %we combine all the vectors
        Vzf=C^-1*Vest;
        Sest=distance3(Vzf,L/N,N); %returns the estimated symbols
        %let's go to check if the packet is correct
        dem_packet=demodulation(Sest,L,N); %we recuperate the sequence of bits
        dec_packet=decoder(dem_packet); %Info+CRC
        aux=dec_packet(1:packet_size); %we take the information bits
        aux2=CRC(aux); %we have again Info+CRC
        aux3=xor(dec_packet,aux2); %we check if we have errors
        ack=ones(1,packet_size+16)*aux3'; %if ack=0 we don't have errors
        r=r+1;
        if(ack~=0) %if packet error, we count the total number of error bits
            error_packet=ones(1,packet_size)*xor(S(1:packet_size,i),aux');
            if(r<R)
                sent_packets=sent_packets+1; %we will have another repetition
            end;
        else
            error_packet=0; %free error packet
        end;
        noise=sqrt((10^((-6-SNR)/10))/2)*randn(M,L/N)+j*sqrt((10^((-6-
SNR)/10))/2)*randn(M,L/N); %noise matrix MxL/N
    end;
    if(ack~=0) %we left the loop with errors in the packet
        lost_packet=lost_packet+1;
    end;
    error=error+error_packet; %we add the total number of error bits
end;
BER=error/(n*packet_size); %Bit Error rate
throughput=(n-lost_packet)/sent_packets; %Throughput
Result(1)=BER;
Result(2)=throughput;
return;
```

HARQ function for Time Variant (All Alternative Matrices)

```
function [Result]=HARQ_altSAllVariant(n,R,SNR,packet_size,state,N,M,a1,sigma);
%this function return the probability of error and the thorughput
%n is the number of packets of the simulation
%R is the maximum number of repetitions of the data
%SNR is the signal to noise ratio
%packet_size is the size of the packet.
%state is the seed for the function randn
%N is the number of transmitter antennas
%M is the number of receiver antennas
%M must be equal or higher than N
%The variable random says if we use the algorihm or not


error=0; %this variable counts the total number of bits errors
sent_packets=n; %this variable counts the total number of packets that we send.
Initially, equal to n
lost_packet=0; %this varible counts the total number of packets that we lose
I1=Alternatives_Generator;
I2=Alternatives2_Generator;
P=N;
S=packet(n,packet_size); %we create a matrix with n packets. Each packet is composed by
Info+CRC+Trellis Code modulation
```

```
randn('state',state); %we put the seed in the function randn
%a1=-.9;
%sigma=.19;

for(i=1:n) %the simulation starts...
    H=sqrt(0.5)*randn(M,N)+j*sqrt(0.5)*randn(M,N);%we create a matrix with iid complex
gaussian parameters for the channel at the transmission -1
    r=0; %this variable counts the current repetition
    ack=1; %this variable tells us if the packet is correct or not
    error_packet=0; %this variable counts the number of error bits in a packet
    v=modulation(S(:,i)); %we get the QPSK signal from each packet
    L=length(v); %L must be divisible by N
    V=split(v,N,L); %we split the packet in N equal subpackets and we put in a matrix of
size Nx(L/N)
    Vest=0*ones(N,L/N); %estimated vector at the receiver Nx(L/N)
    noise=sqrt((10^((-6-SNR)/10))/2)*randn(M,L/N)+j*sqrt((10^((-6-
SNR)/10))/2)*randn(M,L/N); %noise matrix MxL/N
    C0=conj(a1*a1*H'*H+N*sigma*eye(N));%we estimate the channel at instant 0 through the
coefs at -1 determined by a training signal
    H=H*a1+sqrt(0.5*sigma)*randn(M,N)+j*sqrt(0.5*sigma)*randn(M,N); %the AR-1 algorithm
    J=eye(4);
    J(1,1)=-1;
    Cn=J'*C0*J;
    if(det(C0)>=det(Cn))
        matrix=0;
    else
        matrix=1;
    end
    while((r<R)&&(ack~=0))%while the packet still have errors and we have still more
repetitions
        y=mod(r,P); %this variable tells us which number of the sequence we are running
in the Alamouti
        switch (y)
            case 0
                if(matrix==0)
                    B1=I1;
                    x=H'*H*V+H'*noise;
                    C=H'*H;
                    C2=conj(a1*a1*C+N*sigma*eye(N));%we estimate C2 with C1
                    [A,max]=decisorSAllVariant(B1,C,C2); %this function returns the
matrix A with the best order for transmission
                else
                    B2=I2;
                    x=J'*H'*H*J*V+J'*H'*noise;
                    C=J'*H'*H*J;
                    C2=conj(a1*a1*C+N*sigma*eye(N));
                    [A,max]=decisorSAllVariant(B2,C,C2); %this function returns the
matrix A with the best order for transmission
                end;
            case 1
                C2=H'*H;%this is the real value of the channel
                x=A(1:4,:)'*conj(H')*conj(H)*A(1:4,:)*V+A(1:4,:)'*conj(H')*conj(noise);
                C=C+A(1:4,:)'*conj(C2)*A(1:4,:);
                C3=conj(a1*a1*C2+N*sigma*eye(N));
                [A,conjugate]=decisorSAllVariant1(A,max,C,C3);
            case 2
                C3=H'*H;%this is the real value of the channel
                if(conjugate==0)
                    x=A(1:4,:)'*conj(H')*conj(H)*A(1:4,:)*V+A(1:4,:)'*conj(H')*noise;
                    C=C+A(1:4,:)'*conj(C3)*A(1:4,:);
                else
                    x=A(1:4,:)'*H'*H*A(1:4,:)*V+A(1:4,:)'*H'*noise;
                    C=C+A(1:4,:)'*C3*A(1:4,:);
                end;
            case 3
                C4=H'*H;%this is the real value of the channel
                if(conjugate==1)
                    x=A(5:8,:)'*conj(H')*conj(H)*A(5:8,:)*V+A(5:8,:)'*conj(H')*noise;
                    C=C+A(5:8,:)'*conj(C4)*A(5:8,:);
                else
                    x=A(5:8,:)'*H'*H*A(5:8,:)*V+A(5:8,:)'*H'*noise;
```

```
                    C=C+A(5:8,:)'*C4*A(5:8,:);
                end;
        end;
        Vest=Vest+x; %we combine all the vectors
        Vzf=C^-1*Vest; %we use the zero forcing after comining all the vectors
        Sest=distance3(Vzf,L/N,N); %returns the estimated symbols
        %let's go to check if the packet is correct
        dem_packet=demodulation(Sest,L,N); %we recuperate the sequence of bits
        dec_packet=decoder(dem_packet); %Info+CRC
        aux=dec_packet(1:packet_size); %we take the information bits
        aux2=CRC(aux); %we have again Info+CRC
        aux3=xor(dec_packet,aux2); %we check if we have errors
        ack=ones(1,packet_size+16)*aux3'; %if ack=0 we don't have errors
        r=r+1;
        if(ack~=0) %if packet error, we count the total number of error bits
            error_packet=ones(1,packet_size)*xor(S(1:packet_size,i),aux');
            if(r<R)
                sent_packets=sent_packets+1; %we will have another repetition
            end;
        else
            error_packet=0; %free error packet
        end;
        H=H*a1+sqrt(0.5*sigma)*randn(M,N)+j*sqrt(0.5*sigma)*randn(M,N); %the AR-1
algorithm
        noise=sqrt((10^((-6-SNR)/10))/2)*randn(M,L/N)+j*sqrt((10^((-6-
SNR)/10))/2)*randn(M,L/N); %noise matrix MxL/N
    end;
    if(ack~=0) %we left the loop with errors in the packet
        lost_packet=lost_packet+1;
    end;
    error=error+error_packet; %we add the total number of error bits
end;
BER=error/(n*packet_size); %Bit Error rate
throughput=(n-lost_packet)/sent_packets; %Throughput
Result(1)=BER;
Result(2)=throughput;
return;
```

Extended Alamouti Generator function

```
function [I]=Extended_Alamouti_Generator(N);

columns=N;
rows=N;
I=zeros(rows,columns);
aux=zeros(N,N);
I(1:N,:)=eye(N); %the first matrix is always diagonal
counter=1;
power=1;

for i=1:N-3
    for j=i+1:N-1
        aux(i,j)=(-1)^(power);
        aux(j,i)=(-1)^(power+1);
        aux(N+1-j,N)=(-1)^(power);
        aux(N,N+1-j)=(-1)^(power+1);
        I(N*counter+1:N*counter+N,:)=aux;
        aux=zeros(N,N);
        counter=counter+1;
        %power=power+1;
    end;
    power=1;
end;
aux(1,4)=1;
aux(2,3)=-1;
aux(3,2)=-1;
aux(4,1)=1;
I(N*counter+1:N*counter+N,:)=aux;
return;
```

Packet function

```
function [s]=packet(n,packet_size);
%we create n random packets of size packet_size
s=0*ones((packet_size+16+2)*2,n); %16 are the bits of the CRC and 2 are the extra bits
for the convolutional code
I=floor(1.999999*rand(packet_size,n));
s(1:packet_size,:)=I; %we copy the information
for(i=1:n)
    s(1:packet_size+16,i)=CRC(s(1:packet_size,i))'; %we add the CRC
    s(:,i)=encoder(s(1:packet_size+16,i))'; %we do TCM
end;
return;
```

Encoder Function

```
function [X]=encoder(I);
%This function does a convolutional Trellis Code (2,1,3)
L=length(I);
I(L+1)=0;
I(L+2)=0; %we have to add 2 zeros in the packet I
X=0*ones(1,2*(L+2));
state=1;
for(i=1:L)
    switch(state)
        case 1
            if(I(i)==1)
                X(2*i-1)=1;
                X(2*i)=1;
                state=3;
            else
                X(2*i-1)=0;
                X(2*i)=0;
                state=1;
            end;
        case 2
            if(I(i)==1)
                X(2*i-1)=0;
                X(2*i)=0;
                state=3;
            else
                X(2*i-1)=1;
                X(2*i)=1;
                state=1;
            end;
        case 3
            if(I(i)==1)
                X(2*i-1)=1;
                X(2*i)=0;
                state=4;
            else
                X(2*i-1)=0;
                X(2*i)=1;
                state=2;
            end;
        case 4
            if(I(i)==1)
                X(2*i-1)=0;
                X(2*i)=1;
                state=4;
            else
                X(2*i-1)=1;
                X(2*i)=0;
                state=2;
            end;
    end;
end;
return;
```

CRC function

```
function [Y]=CRC(I);
%this function return the packet I + CRC
%we use the polynom for CRC-16

g=[1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]; %CRC polynom
G=length(g);
L=length(I);
Y=0*ones(1,L+G-1);
Y(1:L)=I;
Q=0*ones(1,G-1+L);
Q(1:L)=I; %we shift the vector I with G-1 zeros
fin=0;
j=0; %this count the number of 0 in the residu
k=0;
bits=G;
zeros=0;
C=xor(Q(1:G),g);
while(fin==0)
    while(k==0)
        if((j<G)&&(C(j+1)==0))
            j=j+1;
        else
            k=1;
        end;
    end;
    k=0;
    M=G+L-1-bits;
    if(j<=M)
        for(i=1:G) %number of bits that we have to shift
            if((i<=G-j))
                C(i)=C(i+j);
            else
                if((zeros+i+j)<(G+L)) %we put the bits of Q for the next XOR
                    C(i)=Q(G+zeros+i-(G-j));
                    bits=bits+1; %we count the bits of Q that we have put
                end;
            end;
        end;
    else %we are in the last bits of Q
        for(i=1:G) %we do the same but instead of shift j bits we shift M bits
            if((i<=G-M))
                C(i)=C(i+M);
            else
                if((zeros+i+M)<(G+L))
                    C(i)=Q(G+zeros+i-(G-M));
                    bits=bits+1;
                end;
            end;
        end;
    end;
    if(bits==(G+L-1)) %we have used all the bits of Q
        if(C(1)==1) %special case, that we need to do the last sum XOR
            C=xor(C,g);
        end;
        Y(L+1:L+G-1)=C(2:G); %the last G-1 bits are the bits of the CRC
        fin=1;
    end;
    zeros=zeros+j;
    j=0;
    C=xor(C,g);
end;
return;
```

Modulation Function

```
function [v]=modulation(S);
%alphabet:1,-1,j,-j
%1->11
%-1->00
%j->01
%-j->10
L=length(S);
v=0*ones(1,L/2);
for(i=1:L/2)
    B=S(2*i-1:2*i);
    if(B(1)==1)
        if(B(2)==1)
            v(i)=1;
        else
            v(i)=-j;
        end;
    else
        if(B(2)==1)
            v(i)=j;
        else
            v(i)=-1;
        end;
    end;
end;
return;
```

Split function

```
function [V]=split(v,N,L);
%this function splits the packet v in N equal subpackets and it puts them in a matrix of
size Nx(L/N)
V=zeros(N,L/N);
for i=1:N
    V(i,:)=v((i-1)*L/N+1:i*L/N);
end;
return;
```

Decisor Function (for Extended Alamouti)

```
function [A,conjugate]=decisor(C,C0);

%conjugate=0 means that the last vector is without conjugating, otherwise,
%the vector without conjugate is the third one.
A=zeros(12,4);
I1=zeros(4,4);
I2=zeros(4,4);
I3=zeros(4,4);
%Alternative 1
I1(1,2)=-1;
I1(2,1)=1;
I1(3,4)=-1;
I1(4,3)=1;
%Alternative 2
I2(1,3)=-1;
I2(2,4)=-1;
I2(3,1)=1;
I2(4,2)=1;
```

```
%Alternative 3 vector without conjugating
I3(1,4)=1;
I3(2,3)=-1;
I3(3,2)=-1;
I3(4,1)=1;

R(1)=det(C+I1(1:4,:)'*conj(C0)*I1(1:4,:));
R(2)=det(C+I2(1:4,:)'*conj(C0)*I2(1:4,:));

if(R(1)>R(2))
    A(1:4,:)=I1;
    C=C+I1(1:4,:)'*conj(C0)*I1(1:4,:);
    R(1)=det(C+I2(1:4,:)'*conj(C0)*I2(1:4,:));
    R(2)=det(C+I3(1:4,:)'*C0*I3(1:4,:));
    if(R(1)>R(2))
        A(5:8,:)=I2;
        A(9:12,:)=I3;
        conjugate=0;
    else
        A(5:8,:)=I3;
        A(9:12,:)=I2;
        conjugate=1;
    end;
else
    A(1:4,:)=I2;
    C=C+I2(1:4,:)'*conj(C0)*I2(1:4,:);
    R(1)=det(C+I1(1:4,:)'*conj(C0)*I1(1:4,:));
    R(2)=det(C+I3(1:4,:)'*C0*I3(1:4,:));
    if(R(1)>R(2))
        A(5:8,:)=I1;
        A(9:12,:)=I3;
        conjugate=0;
    else
        A(5:8,:)=I3;
        A(9:12,:)=I1;
        conjugate=1;
    end;
end;
return;
```

Decisor Function (for Time Variant and All Alternative Matrices)

```
function [A,max]=decisorSAllVariant(I,C1,C2);

R(1)=det(C1+I(1:4,:)'*conj(C2)*I(1:4,:));
R(2)=det(C1+I(5:8,:)'*conj(C2)*I(5:8,:));
%R(3)=det(C1+I(9:12,:)'*C2*I(9:12,:));
R(3)=det(C1+I(13:16,:)'*conj(C2)*I(13:16,:));
%R(5)=det(C1+I(17:20,:)'*C2*I(17:20,:));
R(4)=det(C1+I(21:24,:)'*conj(C2)*I(21:24,:));
%R(7)=det(C1+I(25:28,:)'*C2*I(25:28,:));
R(5)=det(C1+I(29:32,:)'*conj(C2)*I(29:32,:));
R(6)=det(C1+I(33:36,:)'*conj(C2)*I(33:36,:));
%R(10)=det(C1+I(37:40,:)'*C2*I(37:40,:));
%R(11)=det(C1+I(41:44,:)'*C2*I(41:44,:));
R(7)=det(C1+I(45:48,:)'*conj(C2)*I(45:48,:));
%R(13)=det(C1+I(49:52,:)'*C2*I(49:52,:));

max=1;
for i=2:7
    if(R(i)>R(max))
        max=i;
    end;
end;

switch (max)
    case 1
```

```
        A(1:12,:)=I(1:12,:);
        A(13:16,:)=I(29:32,:);
        A(17:28,:)=I(41:52,:);
    case 2
        A(1:4,:)=I(5:8,:);
        A(5:8,:)=I(1:4,:);
        A(9:28,:)=I(9:28,:);
    case 3
        A(1:4,:)=I(13:16,:);
        A(5:8,:)=I(5:8,:);
        A(9:12,:)=I(17:20,:);
    case 4
        A(1:4,:)=I(21:24,:);
        A(5:8,:)=I(5:8,:);
        A(9:12,:)=I(25:28,:);
    case 5
        A(1:12,:)=I(29:40,:);
        A(13:16,:)=I(1:4,:);
        A(17:20,:)=I(41:44,:);
    case 6
        A(1:4,:)=I(33:36,:);
        A(5:8,:)=I(29:32,:);
        A(9:12,:)=I(37:40,:);
    case 7
        A(1:4,:)=I(45:48,:);
        A(5:8,:)=I(1:4,:);
        A(9:12,:)=I(49:52,:);
end;
```

Determinant Function

```
function [y]=determinant(I,C0x,Ci,k,N);
%this function choose the matrix which has the highest determinant
result=ones(1,k);
for(i=0:k-1)
    result(i+1)=det(Ci+I(N*i+1:N*i+N,:)'*C0x*I(N*i+1:N*i+N,:));
end;
max=1;
for(i=2:k)
    if(result(i)>result(max))
        max=i;
    end;
end;
y=max-1;
return;
```

Send Fuction

```
function [x]=send(v,H,noise,I,y);

if(y==0)
    x=H'*H*v+H'*noise;
else
    x=I'*conj(H')*conj(H)*I*v+I'*conj(H')*conj(noise);
end;
return;
```

Zero forcing Function

```
function [Vzf]=ZF(Vest,H,I,r,y,N,P);
%this function implement the algorithm of the zero forcing
%we try to find the inverse matrix that we'll cancel the coeficients

C=H'*H;
W=C;
for(i=1:r)
    if(y==0)%we don't need to do zero forcing
        W=eye(N);
        break;
    else
        p=mod(i,P);
        if(p==0)
            W=W+C;
        else
            W=W+I(N*p+1:N*p+N,:)'*conj(C)*I(N*p+1:N*p+N,:);
        end;
    end;
end;
Vzf=W^-1*Vest;
return;
```

Distance Function

```
function [Sest]=distance3(Vest,L,N);

D=ones(4,L);
for(i=1:N)
    D(1,:)=abs(Vest(i,:)-1);
    D(2,:)=abs(Vest(i,:)-j);
    D(3,:)=abs(Vest(i,:)+1);
    D(4,:)=abs(Vest(i,:)+j);
    Sest(i,:)=mindistance(D);

end;
return;
```

Mindistance Function (in C)

```
#include "mex.h"

void mindistance(double *y, double *zr, double *zi,int m, int n)
{
  int i,j,min,count1,count2;    /*count1 for the input matrix, count2 for output matrix*/
  count1=0;
  count2=0;
  min=0;
  zr[0]=0.0;
  zi[0]=0.0;

  for (i = 0; i < n; i++) {
    for (j = 0; j < m; j++) {

        if(*(y+count1+j)<*(y+count1+min)) {
            min=j;
        }
    }

    count1=count1+m;
    if(min==0)  {
        *(zr+count2)=1;
        *(zi+count2)=0;
    }
    if(min==1)  {
        *(zr+count2)=0;
        *(zi+count2)=1;
```

```
    }
    if(min==2)  {
        *(zr+count2)=-1;
        *(zi+count2)=0;
    }
    if(min==3)  {
        *(zr+count2)=0;
        *(zi+count2)=-1;
    }
    min=0;
    count2++;
    }
}


/* The gateway routine */
void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
  double *y;
  double *zr,*zi;
  int mrows,ncols;

  /*  Check for proper number of arguments. */
  /* NOTE: You do not need an else statement when using
     mexErrMsgTxt within an if statement. It will never
     get to the else statement if mexErrMsgTxt is executed.
     (mexErrMsgTxt breaks you out of the MEX-file.)
  */
  if (nrhs != 1)
    mexErrMsgTxt("One input required.");
  if (nlhs != 1)
    mexErrMsgTxt("One output required.");

  /* Create a pointer to the input matrix y. */
  y = mxGetPr(prhs[0]);

  /* Get the dimensions of the matrix input y. */
  mrows = mxGetM(prhs[0]);
  ncols = mxGetN(prhs[0]);

  /* Set the output pointer to the output matrix. */
  plhs[0] = mxCreateDoubleMatrix(1,ncols, mxCOMPLEX);

  /* Create a C pointer to a copy of the output matrix. */
  zr = mxGetPr(plhs[0]);
  zi = mxGetPi(plhs[0]);

  /* Call the C subroutine. */
  mindistance(y,zr,zi,mrows,ncols);
}
```

Demodulation Function

```
function [Dem_packet]=demodulation(Sest,L,N)
%first we have to join the N parts of the packet
Packet=0*ones(1,L);
for i=1:N
    Packet((i-1)*L/N+1:i*L/N)=Sest(i,:);
end;
Dem_packet=0*ones(1,2*L);
for(i=1:L)
    Q=Packet(i);
    switch Q
        case 1
            Dem_packet(2*i-1:2*i)=[1,1];
        case -1
            Dem_packet(2*i-1:2*i)=[0,0];
        case j
            Dem_packet(2*i-1:2*i)=[0,1];
```

```
        case -j
            Dem_packet(2*i-1:2*i)=[1,0];
    end;
end;
```

Decoder Function

```
function [Yf]=decoder(Z);
%this function uses the viterbi algorithm for decoding code
%state 1='00'   state 2='01'    state 3='10'    state 4='11'
M=length(Z);
m=M/2; %this is the size of the trellis diagram
D=1000*ones(4,m); %this matrix mesure the distances
Y=0*ones(1,m); %the last two digits are 0's

%the first two cases are special because we don't have to compare between
%two different paths
x=Z(1:2);
d1=Hamdistance([0,0],x);
D(1,1)=d1;
d2=Hamdistance([1,1],x);
D(3,1)=d2;
x=Z(3:4);
d1=Hamdistance([0,0],x);
D(1,2)=D(1,1)+d1;
d2=Hamdistance([1,1],x);
D(3,2)=D(1,1)+d2;
d3=Hamdistance([0,1],x);
D(2,2)=D(3,1)+d3;
d4=Hamdistance([1,0],x);
D(4,2)=D(3,1)+d4;
for(i=3:m)
    x=Z(2*i-1:2*i); %we take two bits every time
    %to arrive in state 1 we can arrive from state 1 or 2
    d1=Hamdistance([0,0],x); %from state 1
    d2=Hamdistance([1,1],x); %from state 2
    if((D(1,i-1)+d1)<(D(2,i-1)+d2))
        D(1,i)=D(1,i-1)+d1;
    else
        D(1,i)=D(2,i-1)+d2;
    end;
    %to arrive in state 2 we can arrive from state 3 or 4
    d1=Hamdistance([0,1],x); %from state 3
    d2=Hamdistance([1,0],x); %from state 4
    if((D(3,i-1)+d1)<(D(4,i-1)+d2))
        D(2,i)=D(3,i-1)+d1;
    else
        D(2,i)=D(4,i-1)+d2;
    end;
    %to arrive in state 3 we can arrive from state 1 or 2
    d1=Hamdistance([1,1],x); %from state 1
    d2=Hamdistance([0,0],x); %from state 2
    if((D(1,i-1)+d1)<(D(2,i-1)+d2))
        D(3,i)=D(1,i-1)+d1;
    else
        D(3,i)=D(2,i-1)+d2;
    end;
    %to arrive in state 4 we can arrive from state 3 or 4
    d1=Hamdistance([1,0],x); %from state 3
    d2=Hamdistance([0,1],x); %from state 4
    if((D(3,i-1)+d1)<(D(4,i-1)+d2))
        D(4,i)=D(3,i-1)+d1;
    else
        D(4,i)=D(4,i-1)+d2;
    end;
end;
%now we have a matrix D with all the shortest paths
%the last two columns are special because that we know that we have to
```

```
%receive two 0's
D(2,m)=10000;
D(3,m)=10000;
D(4,m)=10000;
D(3,m-1)=10000;
D(4,m-1)=10000;

for(i=1:m)    %we move backwards like the crabs
    v(m+1-i)=minimum(D(:,m+1-i)); %this vector cointain in what state we have the
shortest path
end;
last_state=1;
for(i=1:m-1)
    switch(last_state)
        case 1  %we are in the state 1
            if(v(m-i)==1)    %we came from the state 1
                last_state=1;
                Y(m+1-i)=0;
            end;
            if(v(m-i)==2)    %we came from the state 2
                last_state=2;
                Y(m+1-i)=0;
            end;
            if(v(m-i)==0)
                x=Z(2*(m+1-i)-1:2*(m+1-i));
                d1=Hamdistance([0,0],x); %from state 1
                d2=Hamdistance([1,1],x); %from state 2
                if(d1<d2)
                    last_state=1;
                    Y(m+1-i)=0;
                else
                    last_state=2;
                    Y(m+1-i)=0;
                end;
            end;
        case 2  %we are in the state 2
            if(v(m-i)==3)    %we came from the state 3
                last_state=3;
                Y(m+1-i)=0;
            end;
            if(v(m-i)==4)    %we came from the state 4
                last_state=4;
                Y(m+1-i)=0;
            end;
            if(v(m-i)==0)
                x=Z(2*(m+1-i)-1:2*(m+1-i));
                d1=Hamdistance([0,1],x); %from state 3
                d2=Hamdistance([1,0],x); %from state 4
                if(d1<d2)
                    last_state=3;
                    Y(m+1-i)=0;
                else
                    last_state=4;
                    Y(m+1-i)=0;
                end;
            end;
        case 3  %we are in the state 3
            if(v(m-i)==1)    %we came from the state 1
                last_state=1;
                Y(m+1-i)=1;
            end;
            if(v(m-i)==2)    %we came from the state 2
                last_state=2;
                Y(m+1-i)=1;
            end;
            if(v(m-i)==0)
                x=Z(2*(m+1-i)-1:2*(m+1-i));
                d1=Hamdistance([1,1],x); %from state 1
                d2=Hamdistance([0,0],x); %from state 2
                if(d1<d2)
                    last_state=1;
```

```
                Y(m+1-i)=1;
            else
                last_state=2;
                Y(m+1-i)=1;
            end;
        end;
    case 4  %we are in the state 4
        if(v(m-i)==3)   %we came from the state 3
            last_state=3;
            Y(m+1-i)=1;
        end;
        if(v(m-i)==4)   %we came from the state 4
            last_state=4;
            Y(m+1-i)=1;
        end;
        if(v(m-i)==0)
            x=Z(2*(m+1-i)-1:2*(m+1-i));
            d1=Hamdistance([1,0],x); %from state 3
            d2=Hamdistance([0,1],x); %from state 4
            if(d1<d2)
                last_state=3;
                Y(m+1-i)=1;
            else
                last_state=4;
                Y(m+1-i)=1;
            end;
        end;
    end;
end;
%Finally we treat with the last column
if(last_state==1)
    Y(1)=0;
else %we came from the state 3
    Y(1)=1;
end;
%we know that the last 2 bits are 0
Yf=Y(1:m-2);
return;
```

Hamdistance Function (in C)

```
#include "mex.h"

void Hamdistance(double *x, double *y, double *z,int columns)
{
  int i;
  z[0]=0.0;

  for (i = 0; i < columns; i++) {
    if(*(x+i)!=*(y+i))  z[0]++;
  }
}


/* The gateway routine */
void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
  double *x, *y;
  double *z;
  int mrows,ncols;

  /*  Check for proper number of arguments. */
  /* NOTE: You do not need an else statement when using
     mexErrMsgTxt within an if statement. It will never
     get to the else statement if mexErrMsgTxt is executed.
     (mexErrMsgTxt breaks you out of the MEX-file.)
  */
  if (nrhs != 2)
```

```
    mexErrMsgTxt("One input required.");
  if (nlhs != 1)
    mexErrMsgTxt("One output required.");

  /* Create pointers to the input matrix x and y. */
  x = mxGetPr(prhs[0]);
  y = mxGetPr(prhs[1]);

  /* Get the dimensions of the matrix input y. */
  mrows = mxGetM(prhs[0]);
  ncols = mxGetN(prhs[0]);


  /* Set the output pointer to the output matrix. */
  plhs[0] = mxCreateDoubleMatrix(1,1, mxREAL);

  /* Create a C pointer to a copy of the output matrix. */
  z = mxGetPr(plhs[0]);


  /* Call the C subroutine. */
  Hamdistance(y,x,z,ncols);
}
```

Minimum Function

```
function [min]=minimum(v);
%this function returns the position of the minimum value.
%If the are two minimuns values returns 0
L=length(v);
min=1;
equal=0;
for (k=1:L)
        if(v(k)<v(min))
            min=k;
        end;
end;
for(k=1:L)
    if((k~=min)&&(v(k)==v(min)))
        equal=1;
    end;
end;
if(equal==1)
    min=0;
end;
return
```

REFERENCES

[1] G.E. Malagarriga and Y. Bar-Ness, "New Hybrid Automatic Repeat ReQuest (HARQ) Combining with Space-Time Block Coding (STBC) in invariant and variant fading channel", NJIT Master's Thesis, January 2006.

[2] M. Rupp and C.F. Mecklenbrauker, "On Extended Alamouti Schemes for Space-Time Coding", *The 5th International Symposium on Wireless Personal Multimedia Communications*, vol. 1, pp. 115-119, October 27-30, 2002.

[3] G. Tsoulos "Adaptive Antennas and MIMO Systems for Wireless Communications: Part I", *IEEE Communications Magazine*, vol. 42, no. 10, p. 26, October 2004.

[4] S.M. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications" *IEEE JSAC*, vol. 16(8), pp. 1451-1458. October 1998.

[5] V. Tarokh, H. Jafarkhani and A.R. Calderbank, "Space-Time Block Codes from Orthogonal Designs", *IEEE Transactions*, vol. 45, no. 5, pp. 1456-1467, July 1999.

[6] H. Jafarkhani, "A Quasi-Orthogonal Space-Time Block Code", *IEEE Transactions*, vol. 49, no. 1, pp. 1- 4, January 2001.

[7] E.N. Onggosanusi, A.G. Dabak, Y. Hui and G. Jeong, "Hybrid ARQ transmission and combining for MIMO systems" *IEEE International Conference on Communications*, vol. 26, no 1, pp. 3205-3209J, May 2003.

[8] Y. Bar-Ness, "HARQ for MIMO System using Multiple Alamouti Coding" Draft report for research performed within a collaborated grant with Samsung (SIAT) Korea, January 2005.

[9] "Enhanced STC subpacket combining in OFDMA", submitted for patent to IEEE 802.16 group by LG Electronics, Inc. and Nortel Networks, March 16, 2005.

[10] "Hybrid ARQ schemes for MIMO system", presented by Samsung Corp, May 25, 2005.

[11] B. Badic, M. Rupp and H. Weinrichter, "Adaptive Channel-Matched Extended Alamouti Space-Time Code Exploiting Partial Feedback" *Etri Journal*, vol. 26, no. 5, October 2004.

[12] S. Wang, A. Abdi, J. Salo, H. El-Sallabi, and P. Vainikainen "A Time-Varying MIMO Channel Model: Theory and Measurements," in *Proc. IEEE Int. Workshop Signal Processing Advances Wireless Commun.* (New York, 2005, pp. 343-347).

[13] G.J. Byers and F. Takawira, "Spatially and temporally correlated MIMO channels: modeling and capacity analysis" *IEEE Transactions*, vol. 53, no. 3, pp. 634-643, May 2004.

[14] J. G. Proakis and M. Salehi, *Communications Systems Engineering*. Upper Saddle River, NJ: Prentice-Hall, 1994, pp. 265-267.