ABSTRACT

NEW HYBRID AUTOMATIC REPEAT REQUEST (HARQ) COMBINING WITH SPACE TIME BLOCK CODING (STBC) IN INVARIANT AND VARIANT FADING CHANNEL

by
Guillem Ernest Malagarriga Vilella

Hybrid Automatic Repeat reQuest (HARQ) combining for space time block coding Multiple Input Multiple Output (MIMO) system consisting of N transmit antennas and M receive antennas (NxM) in a time invariant channel and in a time varying channel is proposed and its performance is analyzed. Based on the measured channel matrix, the receiver chooses the best retransmission order and communicated it to the transmitter.

The scheme, where this algorithm is performed, is a combination of a pre-combining HARQ scheme and the Multiple Alamouti Space-Time Block Coding (MA-STBC), which is suitable for more than 2 transmit antennas. The combination of this two techniques increase the efficiency of the HARQ packet transmission by exploiting both the spatial and time diversity of the MIMO channel.

With MA-STBC, there exist different signal packets alternatives for retransmission that can better exploit the characteristics of the channel. An algorithm that chooses the best signal sequence alternatives for retransmission and which exploit better the characteristics of the current channel is proposed. Simulations results show that using the proposed selection algorithm results in a better performance than with random selection. These simulations are performed in invariant and variant channel, and performances are compared with other HARQ schemes.

NEW HYBRID AUTOMATIC REPEAT REQUEST (HARQ) COMBINING WITH SPACE TIME BLOCK CODING (STBC) IN INVARIANT AND VARIANT FADING CHANNEL

by
Guillem Ernest Malagarriga Vilella

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

January 2006

APPROVAL PAGE

NEW HYBRID AUTOMATIC REPEAT REQUEST (HARQ) COMBINING WITH SPACE TIME BLOCK CODING (STBC) IN INVARIANT AND VARIANT FADING CHANNEL

Guillem Ernest Malagarriga Vilella

Dr. Yeheskel Bar-Ness, Dissertation 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: Guillem Ernest Malagarriga Vilella

Degree: Master of Science

Date: January 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 (ETSTB) at Universitat Politécnica de Catalaunya (UPC), Barcelona, Spain, 2004

Major: Telecommunications

To my parents Josep and Josefina

To my sister Marta

To my love Polona

ACKNOWLEDGMENT

I would like to express my appreciation to Dr. Yeheskel Bar-Ness, who not only served as my research supervisor, but also taught me how to develop a career in the academic world. Special thanks are given to Dr. Alexander Haimovich and Dr. Ali Abdi for actively participating in my committee. I would also like to thank my family and especially to my girlfriend Polona for all the support that they gave me during all this year that I have been working in my master.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

LIST OF FIGURES
(Continued)

LIST OF ACRONYMS

ACK – Acknowledgment

AR-1 – Autoregressive model of Order 1

AWGN – Additive White Gaussian Noise

BER – Bit Error Rate

BLAST – Bell Labs Layered Space-Time Architecture

CRC – Cyclic Redundancy Check

HARQ – Hybrid Automatic Repeat reQuest

IR – Incremental Redundancy

LMMSE – Linear Minimum Mean Square Error

LOS – Line of Sight

LZF – Linear Zero Forcing

MAC – Multiple Alamouti Coding

MIMO – Multiple Input Multiple Output

NACK – Negative Acknowledgment

QPSK – Quadrature Phase Shift Keying

SM – Spatial Multiplexing

SNR – Signal to Noise Ratio

STBC – Space Time Block Coding

STC – Space Time Coding

STTD – Space Time Transmit antenna Diversity

ZF – Zero Forcing

CHAPTER 1

INTRODUCTION

1.1 Objective

For NxM antenna MIMO system a Hybrid-ARQ scheme termed Multiple Alamouti HARQ was proposed. With this scheme at each retransmission 2 out of the N transmit antenna are used to transmit two sequence which forms with the corresponding sequences transmitted earlier on these antenna Alamouti space-time block code.

For this scheme of HARQ we propose in this thesis selection algorithm by which the receiver finds the better alternative of two antennas out of the $\frac{N(N-1)}{2}$ possibilities. This will be done for time invariant and time variant channel. For the later we suggest some modification needed.

We will show that using what is called pre-combining HARQ jointly with the proposed STBC, a MIMO system can potentially provide higher throughput packet data service with higher reliability. We will show by simulation that particularly using the proposed selection algorithm performance results are better than with random selection of two antennas alternatives.

1.2 Background

MIMO systems have recently emerged as one of the most important technique in digital wireless communications. This technique satisfies the increased demand for higher data rates and higher network capacity in a wireless environment using the same bandwidth that is used by conventional systems.

The basic idea in MIMO systems is that the natural dimension of digital communication data (time) is complemented with the spatial dimension provided by the multiple distributed antennas. In a wireless system, due to the effect of reflections and attenuations prompted by different objects that are in the signal path, the transmitted signal is received at each antenna with different phase and amplitude. This problem is called multipath distortion.

A key characteristic of MIMO system is the capability to turn this multipath propagation, by tradition a drawback of wireless transmission, into an advantage for the user.

So far, many transmissions schemes for MIMO systems have been proposed in the literature. From [1] these schemes may thickly be classified in two different groups: *spatial multiplexing* (SM) schemes and *space-time coding* (STC) techniques. The first one (SM), uses multiple antennas at both the transmitter and receiver, and send independent data streams over each transmit antennas. The receiver employs an interference cancellation algorithm to separate the data stream. Under this conditions, the number of received antennas must be higher than the number of transmit antennas. An example of an SM scheme is the well-known Bell Labs Layered Space-Time Architecture (BLAST) [2].

Contrasting SM schemes, space-time coding (STC) techniques exploit spatial diversity. To guarantee spatial diversity, it is needed that fade of the individual paths between transmit and received antennas are more or less independently. This assumption of independent paths is not valid if there is not a rich scattering environment or there is a significant line-of-sight (LOS) between the transmitter and the receiver. In this case, the individual transmission paths are correlated among them. In other words, the channel has to follow a Rayleigh fading model [3].

One of the most representative STC techniques is the Alamouti scheme [4] or also called Space Time Block Coding (STBC). In [4] this technique was defined for the case of 2 element transmitter and 1 element receiver. The data stream is split in two mapped symbol blocks denoted s_1 for antenna 1 and s_2 for antenna 2 and are sent at time t. In the next symbol period signal ($t+T$), antenna 1 sends the symbol $\left(-s_2^*\right)$ and antenna 2 sends s_1^*, where * is the complex conjugate operation. The key feature of this technique remains in the fact that the channel is quasi-static between two symbol periods. The channel may be modelled by a complex multiplicative distortion composed of a magnitude response and a phase response. The path between transmit antenna 1 and the received antenna is denoted by h_1 and between the transmit antenna 2 and the received antenna is denoted by h_2 where

$$h_1 = \alpha_1 e^{j\vartheta_1}$$

$$h_2 = \alpha_2 e^{j\vartheta_2} \qquad (1.2.1)$$

The received signals at each symbol period can be modelled as

$$r_1 = r(t) = h_1 s \quad + h_2 s_1 + n_1$$

$$r_2 = r(t+T) = -h_1 s_2^* + h_2 s_1^* + n_2 \qquad (1.2.2)$$

where r_1 and r_2 are the received signals at time t and $t+T$ and n_1 and n_2 are complex random variables representing receiver noise and interference.

Then, the combining scheme builds the following two combined signals that are sent to a maximum likelihood detector:

$$\tilde{s}_1 = h_1^* r_1 + h_2 r_2^*$$

$$\tilde{s}_2 = h_2^* r_1 - h_1 r_2^* \qquad (1.2.3)$$

Note that the receiver previously has to estimate the channel coefficient in order to combine the received signals. Hence, it is important to have a good estimation of the channel to achieve the best performance in the Alamouti scheme. Finally, substituting (1.2.1) and (1.2.2) into (1.2.3) we get

$$\tilde{s}_1 = (\alpha_1^2 + \alpha_2^2)s_1 + h_1^* n_1 + h_2 n_2^*$$

$$\tilde{s}_2 = (\alpha_1^2 + \alpha_2^2)s_2 - h_1 n_2^* + h_2^* n_1 \qquad (1.2.4)$$

Note that the desired symbols are combined in a constructive way because they are multiplied by a sum of absolute terms. The noise, however, is combined incoherently. Due to the spatial diversity, the probability that the factor $(\alpha_1^2 + \alpha_2^2)$ is close to zero is comparable small.

After this brief description of the different MIMO schemes, we will introduce a technique that is bound with MIMO systems: the packet retransmission.

In packet communications systems, a packet retransmission is requested when the original packet is not received free of errors. The simplest scheme is termed Automatic Retransmission reQuest (ARQ), which discards the received erroneous packet and requests a re-retransmission of this one. However, one can perform better if instead of discarding the erroneous packets, he combines them in an intelligent way with the subsequent retransmitted packets. This scheme is called Hybrid ARQ (HARQ).

There are principally two types of techniques used in HARQ: packet combining and Incremental redundancy (IR) [5]. With the first one, also called Chase combining [6], the receiver combine the noisy packets either in the symbol or bit level to obtain a packet with a code rate which is low enough that reliable communication is possible even for low quality channels. In IR, if error is detected, the transmitter only sends the parity bits of the information packet. These parity bits are the error correction code, and at each retransmission the transmitter increase the length of this code, hence, a higher number of errors can be corrected. Combined with HARQ, MIMO can potentially supply higher throughput packet data services with higher reliability.

In this thesis we will focus on *symbol-level* combining. There are two different HARQ combining schemes: Pre-Combining and Post-Combining. In Pre-Combining, the retransmitted packets are combined at the symbol-level and then the cumulative interference is removed using a linear zero forcing (LZF) or minimum mean-square error (LMMSE) equalizer. On the other hand, in Post-Combining, the interference of each retransmitted packet is removed with a LZF or LMMSE equalizer and then the packets free of interference are combined at the symbol-level or at the bit level. Note that, IR technique is only applicable for the case of Post-Combining, since in Pre-Combining it is required that all the retransmitted packets have the same length. In [7] it is shown that Pre-Combining outperforms better than Post-Combining for a known receiver structure. So far, in this thesis we will concentrate in the Pre-Combining scheme.

Since in a MIMO system, one needs that each retransmission introduces diversity to exploit the channel's characteristics, so, when we combine the received packets, the communication reliability increases dramatically. In [7] a Pre-Combining HARQ scheme called Basis Hopping was presented. This technique artificially introduces diversity in a slowly varying channel. The key idea is to multiply the transmitted signal by a unitary complex matrix in order to provide time diversity gain. However, in this thesis, instead of using a unitary matrix to pre-process the transmitted vector so as to introduce diversity as in [7] we use the Alamouti space-time coding to introduce diversity, and pre-combining with ZF to cancel interferences [8]. In the next section we will introduce what is termed multiple Alamouti HARQ scheme, which is an extension of the Alamouti scheme for more than two transmit antennas.

1.3 Problem Statement

The scheme in [8] uses 2x2 antennas. For multiple elements MIMO a scheme termed multiple Alamouti HARQ was proposed in [9] which uses multiple space-time Alamouti with 2 transmit antennas at each retransmission. With *NxM* MIMO system, there would be $\frac{N(N-1)}{2}$ different 2 transmit Alamouti combining possibility. It was shown that after each retransmission and combining at the receiver, the cross data interference corresponding to the chosen antennas couple, are zeroed and direct data terms enhanced, therefore depending on the channel matrix the order of choosing any couple of antennas for retransmission will affect the performance of the HARQ scheme. We will propose an algorithm by which, based on the measured channel matrix, the receiver choose best alternative of 2 antennas for retransmission and communicated this information to the transmitter. We will consider for our combining and selection the case when the channel is invariant; that is remaining the same during all retransmissions. We will also consider the case of time varying channel, in which the channel varies from retransmission to the next. For the later we will suggest modification of selection algorithm.

CHAPTER 2

HARQ FOR MIMO SYSTEMS USING MULTIPLE ALAMOUTI CODING

2.1 System Model Description

In this section we describe the structures that we use to implement the Hybrid-ARQ scheme for MIMO systems. We consider a MIMO system with N transmits antennas and M receives antennas. In later sections we describe this model in more detail for a known number of antennas. First we present the transmitter structure, where we describe all the steps that we apply on the data string of bits till we send these through the air. We then describe the receiver structure, which shows all the necessary process to recover the data that was sent.

2.1.1 Transmitter Structure

The transmitter structure is divided into several blocks. The first is a source of binary information data packets, which is encoded with a high rate code for error detection and then with a half rate convolutional code for error correction. The coded packet is then digitally mapped into a for example QPSK symbol constellation. Finally, these symbols are demultiplexed into N separate data streams to be transmitted from the N individual transmits antennas according to Multiple Alamouti scheme described in the sequel. This transmitter structure of the Multiple Alamouti scheme is depicted in figure 2.1. The signal stream transmitted from antenna i is denoted by s_i.

TRANSMITTER STRUCTURE

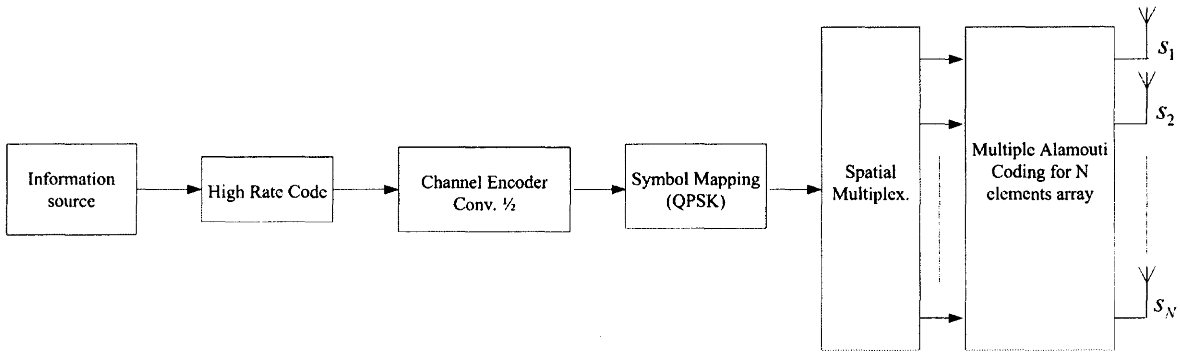

Figure 2.1 Transmitter structure of the Multiple Alamouti H-ARQ scheme

After this brief introduction of the different blocks of the transmitter, we now describe each one of them in more detail:

- Information source: It generate a random string of binary bits where the probability to have a 0 or 1 is the same. We will take as the size of the information bit packet 522 bits. The reason for such choice of packet size is to enable splitting the packet in three, four, five or six parts, for our simulation experiment.

- High rate coder: 16 bits are added at each packet for Cyclic Redundancy Check (CRC) [10]. The CRC polynomial used is of grade 16, given by:

$$G(D) = D^{16} + D^{15} + D^2 + 1$$
$$R(D) = D^{16} X(D) \bmod G(D)$$
$$Y(D) = R(D) + D^{16} X(D)$$

Where, $G(D)$ is the CRC polynomial, $X(D)$ is the input packet (522 bits), $R(D)$ is the checksum (16 bits) and $Y(D)$ is the output packet (522+16).

- Channel encoder: A half rate convolutional code showed in figure 2.2 was employed to encode the data packet for error correction [11]. With this encoder the constraint length of the code is equal to three ($L = 3$). The number of input bits is one ($k = 1$) and the number of output bits is two ($n = 2$). Therefore, the rate of the code is $\frac{1}{2}$ and the number of states is $2^{(L-1)k} = 4$. The state diagram is depicted in figure 2.3.

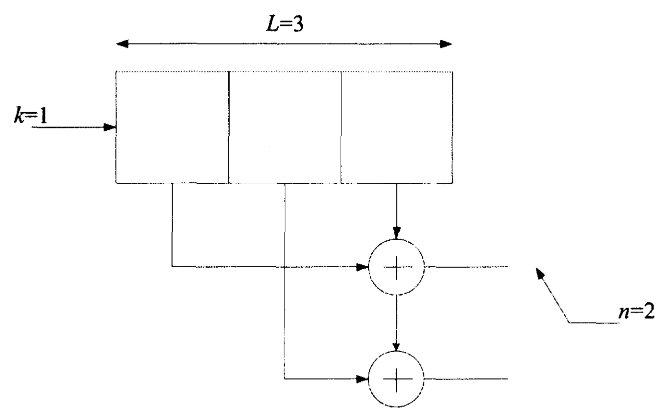

Figure 2.2 A rate $\frac{1}{2}$ convolutional encoder

- It is assumed that the encoder, before the first information bit enters it, is loaded with zeros. Therefore, once the whole packet is loaded into the encoder we have to add $k(L-1)$ zeros (in this case, 2). This makes the encoder ready to be used for the next transmission. Note that if we have two consecutive zeros we always finish in the state 0. Hence, we are always in a known state for decoding.

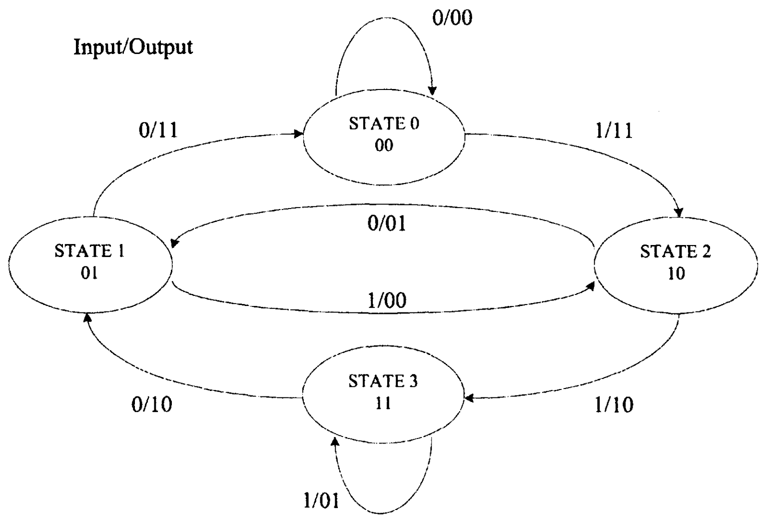

Figure 2.3 State diagram for the encoder of figure 2.2

- <u>Symbol mapping:</u> In this block we map digitally into symbols the binary data from the encoder. We use a Quadrature Phase Shift Keying (QPSK) to map the bits into their equivalent complex baseband notation. Using Gray maping the constellation of the QPSK symbols mapp is depicted in figure 2.4.

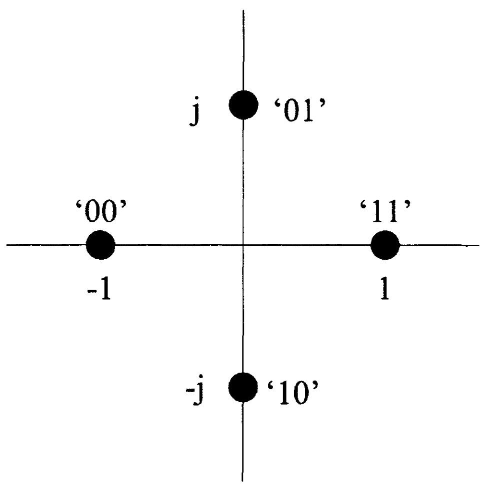

Figure 2.4 QPSK constellation and symbol mapping

- <u>Spatial Multiplexing:</u> The symbol packet is split in N equal parts to be transmitted from the N individual transmit antennas. Note that after the whole process the number of symbols in the packet has to be divisible by the number of antennas (N). The number of symbols per packet is the following,

((522 information bits+16 CRC bits+2 zeros bits)*2)/ (2 bits/symbol) =540 symbols

Note that 540 are divisible by two, three, four, five and six parts.

- Multiple Alamouti coding for N elements array: This scheme of multiple space-time Alamouti uses only 2 transmit antennas at each retransmission. With NxM MIMO system, there would exist $\binom{N}{2}$ different possible 2 transmit Alamouti alternatives. As an example, we write the sets of Alamouti transmitting alternatives for both 3 and 4 transmitter antennas [9].

For three element transmitter, the alternatives are the following columns

$$\begin{matrix} s_1 & -s_2^* & s_3^* & 0 \\ s_2 & s_1^* & 0 & -s_3^* \\ s_3 & 0 & -s_1^* & s_2^* \end{matrix} \qquad (2.1.1)$$

and for four element transmitter, the alternatives following

$$\begin{matrix} s_1 & -s_2^* & 0 & 0 & s_3^* & 0 & -s_4^* \\ s_2 & s_1^* & -s_3^* & 0 & 0 & s_4^* & 0 \\ s_3 & 0 & s_2^* & -s_4^* & -s_1^* & 0 & 0 \\ s_4 & 0 & 0 & s_3^* & 0 & -s_2^* & s_1^* \end{matrix} \qquad (2.1.2)$$

where each signal s_i represents a string of symbols with length 540/N.

2.1.2 Receiver Structure

It was shown by Alamouti though for 2 element that, data detection may be performed by means of a simple matrix vector multiplication, provided that the channel model is a frequency flat. At first we assume that we have a slow fading channel, where the channel matrix H remains constant upon L transmissions. The channel gain between the transmit antenna i and the receive antenna j is denoted by h_{ji} where $i = 1,2,\ldots,N$ and $j = 1,2,\ldots M$. The NxM channel gains are assumed i.i.d. uncorrelated complex Gaussian random variables (Raleigh Flat Fading) with unit power [12], that is,

$$\mathrm{E}\left\{\left|h_{ji}\right|^2\right\} = 1 \text{ and } \mathrm{E}\left\{h_{ji}h_{ml}^*\right\} = 0 \quad i,l = 1\ldots N \qquad j,m = 1\ldots M \quad i \neq l \text{ or } j \neq m \tag{2.1.3}$$

where $\mathrm{E}\{\cdot\}$ is the expectation.

The composite MIMO channel gain can be represented by the following matrix:

$$H_{MxN} = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ h_{M1} & \cdots & \cdots & h_{MN} \end{pmatrix} \tag{2.1.4}$$

We also assume an Additive White Gaussian Noise (AWGN) vector. The noise vectors $\vec{n}$ observed for every transmission are independents.

$$\vec{n} = \begin{bmatrix} n_1 & n_2 & \ldots & n_M \end{bmatrix}^T \tag{2.1.5}$$

The M noise terms are assumed i.i.d. uncorrelated complex Gaussian random variables with power σ^2. We now write the relation between σ^2 and the bit energy (E_b), for a given $\left({}^{E_b}\!/\!_{N_0}\right)$ in order to have the value of σ^2 for the simulation program. We assume, as shown in figure 2.5, that the power spectral density is a white process.

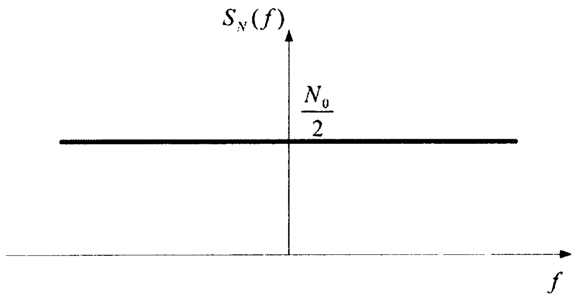

Figure 2.5 Power spectral density of a white process

Using a matched filter at the receiver we have the Signal to Noise Ratio (SNR) defined by

$$\left(\frac{S}{N}\right)=\frac{P_s}{P_n}=\frac{1}{\sigma^2}=\frac{2E_s}{N_0}=\frac{4E_b}{N_0} \qquad \text{where } \sigma^2=\frac{N_0}{2} \qquad (2.1.6)$$

where P_s is the total average signal power at all antennas. Note that we took as the signal power to equal 1. E_s is the symbol energy, and due to the fact we use QPSK modulation, the symbol energy is twice the bit energy. Hence, we get the relation

$$\sigma^2=\frac{N_0}{4E_b} \qquad \text{or in } dB, \qquad \sigma^2_{dB}=-6-\left(\frac{E_b}{N_0}\right)_{dB} \qquad (2.1.7)$$

Once we defined the channel model and the noise, the baseband received signal vector is given by:

$$\vec{r}=H\vec{s}+\vec{n} \qquad (2.1.8)$$

where $\vec{r}=[r_1 \quad r_2 \quad \cdots \quad r_M]^T$ and $\vec{s}=[s_1 \quad s_2 \quad \cdots \quad s_N]^T$

The next figure depicts the structure of the receiver, where in we will explain in detail the whole process to recover the bit information from the baseband received signal.

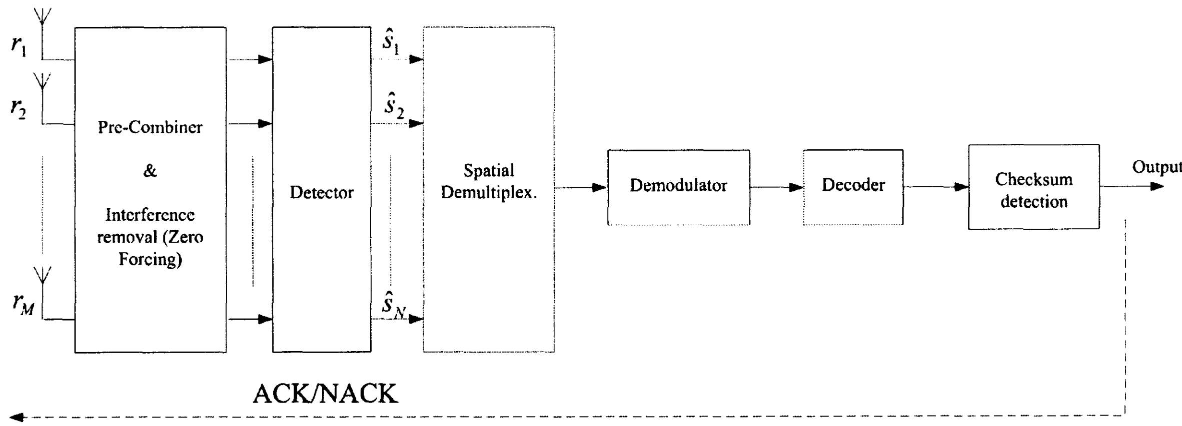

Figure 2.6 Receiver structure of the Multiple Alamouti H-ARQ scheme

- Pre-Combiner and Interference removal: After matched filtering to the received signal, if the received packet is a retransmission of a previous erroneous packet, then the retransmitted packet and the previous one are combined together at the symbol level. Then, a linear equalizer (Zero Forcing) is used to remove the interference and separate the N transmitted data sub-packets.
- Detector: A hard decision decoding is made at each symbol. For each received complex symbol, we calculate the distance among the four possible symbols of the QPSK. The symbol which has the closest distance is taken as the decision.
- Spatial Demultiplexation: This block links the N received sub-packets $(\hat{s}_1 \quad \hat{s}_2 \quad \cdots \quad \hat{s}_N)$ to recover the original symbol packet before being split.
- Demodulator: The aim of this block is to translate the complex symbols of the QPSK in binary bits. From figure 2.4, we have the relation between each complex symbol and his equivalence in bits.

- Decoder: The Viterbi Algorithm [13] is used to make a soft decision decoding of the string of bits. After this block we have a packet which contains the information (522 bits) and the CRC (16 bits).
- Checksum detection: The first 522 bits are taken and the CRC polynomial is used to calculate the new checksum of the received bits ($\hat{R}(D)$). If the new checksum is equal to the received checksum ($\hat{R}(D) = R(D)$), the received packet contain no error, the packet is accepted and a positive Acknowledgment (ACK) is sent to the transmitter otherwise the receiver sends a Negative Acknowledgement (NACK) and the transmitter resends the packet using a retransmission order algorithm for the multiple Alamouti Space Time Coding scheme, as communicate to it by the receiver will be described in future chapters.

2.2 HARQ Combining Scheme for Two Element Transmitter

In this section we describe the new HARQ combining scheme proposed in [12] for two transmitter antennas and two receiver antennas. This new scheme is a combination of the pre-combining scheme proposed in [7] and the Alamouti Space-Time Coding STC [4]. To implement this scheme, we use the model described in section 2.1. We will show how the receiver's block called the "pre-combiner and interference removal" process the data. Then, we present simulations results that compare the performance obtained with this new HARQ combining scheme to the performance of the scheme proposed in [7].

2.2.1 Pre-combiner and Interference Removal

For a two-element transmitter two-element receiver system, the received signal can be model with

$$\begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \tag{2.2.1}$$

where r_l and n_l with $l = 1,2$ is the received signal and noise on the lth receiver antenna; s_l with $l = 1,2$ is the transmitted signal on the lth transmitter antenna; and h_{lk} is the channel gain of the wireless link from the kth transmitter antenna to the lth receiver antenna.

At the first transmission, the signal $s^{(1)} = \vec{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$ is sent out. The corresponding received signal can be expressed as

$$r^{(1)} = Hs^{(1)} + n^{(1)} \tag{2.2.2}$$

with

$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \tag{2.2.3}$$

After matched filtering at the receiver,

$$x^{(1)} = H^{\psi} Hs^{(1)} + H^{\psi} n^{(1)} = Cs^{(1)} + H^{\psi} n^{(1)} \tag{2.2.4}$$

where $(\cdot)^{\psi}$ is the operation of conjugate transpose; $C = H^{\psi} H$. And following zero forcing, we get

$$\hat{s}_1 = C^{-1} x^{(1)} = s^{(1)} + C^{-1} H^{\psi} n^{(1)} \tag{2.2.5}$$

If detection error occurred, the transmitter is requested to resend a new packet using Alamouti Space Time Coded, i.e. new packets composed of $s^{(2)} = \begin{bmatrix} -s_2^* \\ s_1^* \end{bmatrix}$ are sent from the two transmit antennas. The received signal vector at the second transmission is given by:

$$r^{(2)} = Hs^{(2)} + n^{(2)} = H\gamma s^{(1)*} + n^{(2)} \qquad (2.2.6)$$

where,

$$\gamma = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \qquad (2.2.7)$$

by taking the conjugate of the new received vector packets we obtain:

$$r^{(2)*} = H^*\gamma s^{(1)} + n^{(2)*} \qquad (2.2.8)$$

From (2.2.8), it is clear that taking the conjugate of the received vector $r^{(2)*}$ is equivalent to re-sending the previous vector signal trough the new channel $H^*\gamma$ that adds time diversity.

The received vector $r^{(2)}$ is first processed by the linear receiver front end i.e. multiply symbol wise by $(H^*\gamma)^\psi = \gamma^T H^T$ (where $(\cdot)^T$ is the operation of transpose). Therefore,

$$x^{(2)} = \gamma^T H^T H^* \gamma s^{(1)} + \gamma^T H^T n^{(2)*} = \gamma^T C^* \gamma s^{(1)} + \gamma^T H^T n^{(2)*} \qquad (2.2.9)$$

Then a symbol level combining is employed to provide for soft symbol decision for $s^{(1)}$. That is adding both term $x^{(1)}$ and $x^{(2)}$ we get the output for decision

$$\hat{s}_{1,2} = \left(C + \gamma^T C^* \gamma\right)s + H^\psi n^{(1)} + \gamma^T H^T n^{(2)*} \qquad (2.2.10)$$

We now show that $C+\gamma^T C^* \gamma$ is a diagonal matrix. Hence, we do not need to do zero forcing. From definition, we have

$$C=\begin{bmatrix} h_{11}^* & h_{21}^* \\ h_{12}^* & h_{22}^* \end{bmatrix}\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \tag{2.2.11a}$$

$$=\begin{bmatrix} |h_{11}|^2+|h_{21}|^2 & h_{12}h_{11}^*+h_{22}h_{21}^* \\ h_{12}^*h_{11}+h_{22}^*h_{21} & |h_{12}|^2+|h_{22}|^2 \end{bmatrix} \tag{2.2.11b}$$

$$=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}=\begin{bmatrix} a_{11} & a_{12} \\ a_{12}^* & a_{22} \end{bmatrix} \tag{2.2.11c}$$

meanwhile

$$\gamma^T C^* \gamma=\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} a_{11}^* & a_{12}^* \\ a_{12} & a_{22}^* \end{bmatrix}\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.2.12a}$$

$$=\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} a_{12}^* & -a_{11}^* \\ a_{22}^* & -a_{12} \end{bmatrix}=\begin{bmatrix} a_{22}^* & -a_{12} \\ -a_{12}^* & a_{11}^* \end{bmatrix} \tag{2.2.12b}$$

Then, since both a_{11} and a_{22} are real number, we have

$$C+\gamma^T C^* \gamma=\begin{bmatrix} a_{11}+a_{22} & 0 \\ 0 & a_{11}+a_{22} \end{bmatrix}=\begin{bmatrix} \chi & 0 \\ 0 & \chi \end{bmatrix} \tag{2.2.13}$$

where, by substituting from terms in (2.2.11a),

$$\chi=|h_{11}|^2+|h_{21}|^2+|h_{12}|^2+|h_{22}|^2 \tag{2.2.14}$$

The effective noise is given as

$$n=H^{\Psi} n^{(1)}+\gamma^T H^T n^{(2)*} \tag{2.2.15}$$

whose autocorrelation matrix is

$$R_{NN}=\mathrm{E}\left\{\left[\left(H^{\Psi} n^{(1)}+\gamma^T H^T n^{(2)*}\right)\right]\left[\left(H^{\Psi} n^{(1)}+\gamma^T H^T n^{(2)*}\right)\right]^{\Psi}\right\} \tag{2.2.16a}$$

$$= \sigma^2\left(C + \gamma^T C^* \gamma\right) = \begin{bmatrix} \sigma^2\chi & 0 \\ 0 & \sigma^2\chi \end{bmatrix} \tag{2.2.16b}$$

where σ^2 is the noise power that we defined in the previous chapter.

If after decoding the combined packets, no error occurs, the packets are accepted and an ACK is sent otherwise the transmitter resends the packets as in the first transmission i.e. $s^{(3)} = [s_1 \quad s_2]^T$. The newly received signal vector $r^{(3)}$ is combined with the two previous received vectors $r^{(2)*}$ and $r^{(1)}$. The procedure continues ($[s_1 \quad s_2]^T$ sent during the odd transmission and $[-s_2^* \quad s_1^*]^T$ sent during even transmission and all the received vector signal are combined as in (2.2.10)) until the packets are correctly decoded or until a preset maximum allowed number of retransmission attempts is reached. Note that after combining the received packets, we only have to do interference cancellation (ZF) at the odd transmission; since we always have a diagonal matrix after receiving an even transmission.

2.2.2 Numerical Results

We use the model and the data structure proposed in the section 2.1 for the case of two elements transmitters and two elements receivers. The total numbers of packets in the MonteCarlo simulation were 5000. The resulted bit error rates (BER) are depicted in the figure 1 versus E_b / N_0 for different number of transmissions ($R_i, i = 1,2,3,4$). For comparison we also added in the figure what is termed *Basis* which is the Basis Hopping with Pre-Combining Scheme presented in [7]. As we explained earlier, this technique uses an artificially diversity in the slowly varying channel, where in the transmit signal vector is first multiplied by a unitary matrix V. The received vector signal is the given by:

$$r^{(i)} = HV^{(i)}s^{(i)} + n^{(i)},\ i = 1,......,L \tag{2.2.17}$$

where L is the maximum number of transmissions. The matrix $V^{(i)}$ is taken to be different for every retransmission and it changes the MIMO channel from H to $HV^{(i)}$. By doing so, the unitary matrix $V^{(i)}$ introduces time diversity upon retransmission. The received signal vectors $r^{(i)}$ are combined before the interference cancellation by ZF receiver. From [7], we define the following matrix set $S = \{V^{(1)},\ V^{(2)},\ V^{(3)},\ V^{(4)}\}$, where

$$V^{(1)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad V^{(2)} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.2.18a}$$

$$V^{(3)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad V^{(4)} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \tag{2.2.18b}$$

Note that V must be unitary to avoid any increase in the transmitter power.

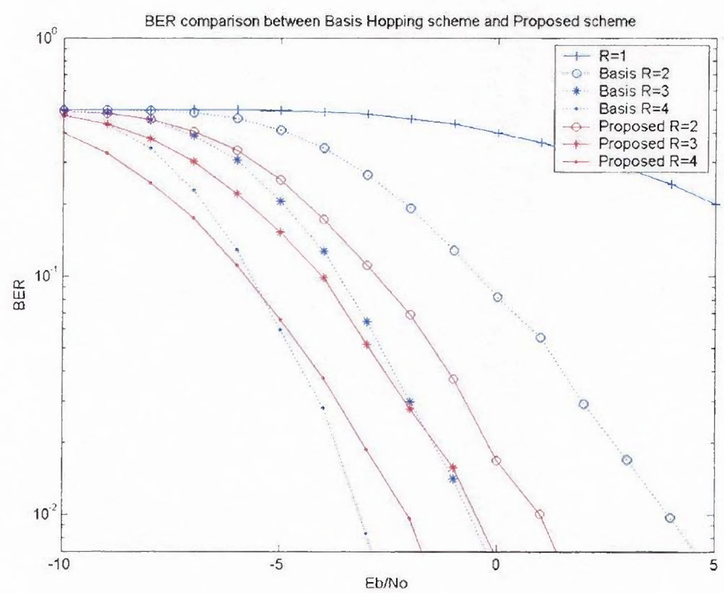

Figure 2.7 BER performance of the new HARQ combining scheme on comparison to Basis Hopping for two transmitter antennas

The next figure depicts the throughput performance of both schemes versus the E_b / N_0. To calculate the throughput we allowed up to 4 transmissions to receive correctly the packet. If after the fourth transmission the packet is still erroneous, it is counted as a lost packet. The formula that we use to calculate the throughput is as follows,

$$Throughput = \frac{K_{TOT} - K_{LOST}}{K_{TOT} + K_R} \tag{2.2.19}$$

Where, K_{TOT} is the total number of different packets that we send (in this case is 5000), K_{LOST} is the number of lost packets and K_R is the total number of retransmissions that we send. Note that if we correctly transmit all the packets at the first attempt, the throughput is 1.

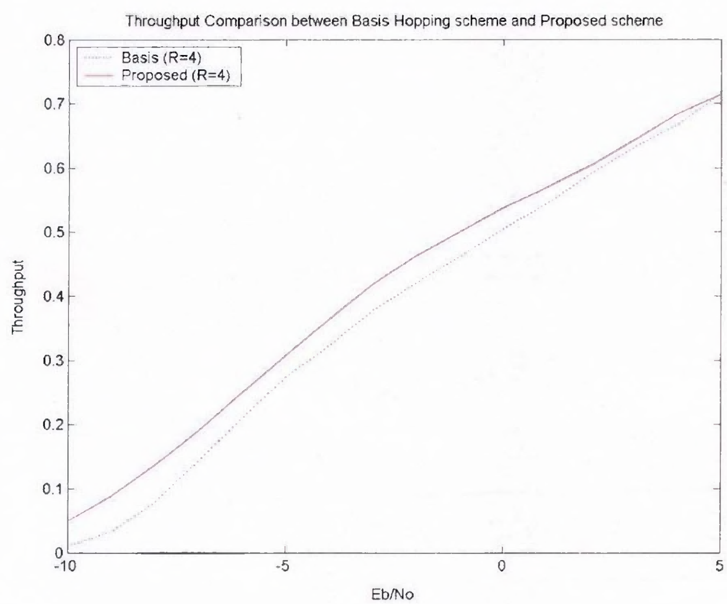

Figure 2.8 Throughput performance of the new HARQ combining scheme in comparison to Basis Hopping for two transmitter antennas

The results show that the new HARQ combining scheme has better performance in terms of BER and throughput. Only after 4th transmission the Basis Hopping scheme have better error probability than the proposed scheme, but this improvement does not affect the result in the throughput.

In the next sections we will discuss the proposed scheme in a MIMO channel with more than two transmit antennas.

2.3 HARQ Combining Scheme for Three Element Transmitter

A new Hybrid Automatic Repeat request (HARQ) transmission scheme for Multiple Input Multiple Output MIMO systems consisting of two transmit antennas in a slowly varying channel was discussed in section 2.2.

This scheme uses multiple space-time Alamouti with 2 transmit antennas at each retransmission. With 3x3 MIMO system, there would exist $\binom{3}{2}$ different alternatives of 2 transmit Alamouti combining. First, we will show that when using data at any chosen antennas couple, then after each retransmission and combining at the receiver, two corresponding off-diagonal terms of the channel matrix are zeroed and two related diagonal terms are increased. Therefore the order of choosing any of these antennas couples in retransmission will affect differently the performance of this HARQ scheme depending on the channel matrix.

We will present an algorithm by which, based on the measured channel matrix, the receiver will decide on the retransmission order and communicated it to the transmitter. A SNR criterion will be used for such order selection.

Simulations results will compare the performance obtained with this selection algorithm to the performance of retransmitting from antenna couples in a random order.

2.3.1 Combining Alternatives

In the previous section, we showed that with two transmitting antennas and using Alamouti Coding the received signal was effectively presented by the transmitted signal through a diagonal channel matrix. We now intend to show how one can get a diagonal channel matrix for the case of three element transmitter.

If detection error occurred, following the first transmission $s^{(1)}$, a request for a sequence out of three retransmissions alternative represented by the columns below

$$\begin{array}{cccc} s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} \\ \\ s_1 & -s_2^* & s_3^* & 0 \\ s_2 & s_1^* & 0 & -s_3^* \\ s_3 & 0 & -s_1^* & s_2^* \end{array} \tag{2.3.1}$$

We can relate the columns of the retransmission to the column vector $s^{(1)}$, as follows

Alternative (1)

$$s^{(2)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ 0 \end{bmatrix} = \gamma_1 s^{(1)*} \tag{2.3.2a}$$

Alternative (2)

$$s^{(3)} = \begin{bmatrix} 0 \\ -s_3^* \\ s_2^* \end{bmatrix} = \gamma_2 s^{(1)*} \tag{2.3.2b}$$

Alternative (3)

$$s^{(4)} = \begin{bmatrix} s_3^* \\ 0 \\ -s_1^* \end{bmatrix} = \gamma_3 s^{(1)*} \tag{2.3.2c}$$

where,

$$\gamma_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.3.3a) \qquad \gamma_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.3.3b)$$

$$\gamma_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (2.3.3c)$$

If we combine all these signals in the same way we did for the case of two antennas, we will end up with channel matrix $C + \gamma_1^T C^* \gamma_1 + \gamma_2^T C^* \gamma_2 + \gamma_3^T C^* \gamma_3$, which we show in the following steps, it is a diagonal.

Following similar approach as before, we define $C = H^{\psi} H$

$$C = \begin{bmatrix} h_{11}^* & h_{21}^* & h_{31}^* \\ h_{12}^* & h_{22}^* & h_{32}^* \\ h_{13}^* & h_{23}^* & h_{33}^* \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (2.34a)$$

$$= \begin{bmatrix} |h_{11}|^2 + |h_{21}|^2 + |h_{31}|^2 & h_{11}^* h_{12} + h_{21}^* h_{22} + h_{31}^* h_{32} & h_{11}^* h_{13} + h_{21}^* h_{23} + h_{31}^* h_{33} \\ h_{11} h_{12}^* + h_{21} h_{22}^* + h_{31} h_{32}^* & |h_{12}|^2 + |h_{22}|^2 + |h_{32}|^2 & h_{12}^* h_{13} + h_{22}^* h_{23} + h_{32}^* h_{33} \\ h_{11} h_{13}^* + h_{21} h_{23}^* + h_{31} h_{33}^* & h_{12} h_{13}^* + h_{22} h_{23}^* + h_{32} h_{33}^* & |h_{13}|^2 + |h_{23}|^2 + |h_{33}|^2 \end{bmatrix} \quad (2.3.4b)$$

$$= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12}^* & a_{22} & a_{23} \\ a_{13}^* & a_{23}^* & a_{33} \end{bmatrix} \quad (2.3.4c)$$

where h_{ji} is the channel gain between the transmit antenna i and the receive antenna j ($i = 1,2,3$ and $j = 1,2,3$). Then,

$$C^* = \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* \\ a_{12} & a_{22} & a_{23}^* \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \quad (2.3.5)$$

and for the first retransmission, if we use alternative (1), we have

$$\gamma_1^T C^* \gamma_1 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* \\ a_{12} & a_{22} & a_{23}^* \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.3.6a)$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{12}^* & -a_{11} & 0 \\ a_{22} & -a_{12} & 0 \\ a_{23} & -a_{13} & 0 \end{bmatrix} \quad (2.3.6b)$$

$$= \begin{bmatrix} a_{22} & -a_{12} & 0 \\ -a_{12}^* & a_{11} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (2.3.6c)$$

As the result, we have after combining with the first transmission of $s^{(1)}$,

$$C + \gamma_1^T C^* \gamma_1 = \begin{bmatrix} a_{11} + a_{22} & 0 & a_{13} \\ 0 & a_{11} + a_{22} & a_{23} \\ a_{13}^* & a_{23}^* & a_{33} \end{bmatrix} \quad (2.3.7)$$

For the second retransmission, if we use alternative (2), we have

$$\gamma_2^T C^* \gamma_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* \\ a_{12} & a_{22} & a_{23}^* \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.3.8a)$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & a_{13}^* & -a_{12}^* \\ 0 & a_{23}^* & -a_{22} \\ 0 & a_{33} & -a_{23} \end{bmatrix} \quad (2.3.8b)$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_{33} & -a_{23} \\ 0 & -a_{23}^* & a_{22} \end{bmatrix} \quad (2.3.8c)$$

and we have after combining with previous combined signal

$$C + \gamma_1^T C^* \gamma_1 + \gamma_2^T C^* \gamma_2 = \begin{bmatrix} a_{11} + a_{22} & 0 & a_{13} \\ 0 & a_{11} + a_{22} + a_{33} & 0 \\ a_{13}^* & 0 & a_{22} + a_{33} \end{bmatrix} \quad (2.3.9)$$

For the third retransmission, if we use alternative (3), we have

$$\gamma_3^T C^* \gamma_3 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12}^* & a_{13}^* \\ a_{12} & a_{22} & a_{23}^* \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (2.3.10a)$$

$$= \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -a_{13}^* & 0 & a_{11} \\ -a_{23}^* & 0 & a_{12} \\ -a_{33} & 0 & a_{13} \end{bmatrix} \quad (2.3.10b)$$

$$= \begin{bmatrix} a_{33} & 0 & -a_{13} \\ 0 & 0 & 0 \\ -a_{13}^* & 0 & a_{11} \end{bmatrix} \quad (2.3.10c)$$

And we have after combining with the previous combined signal

$$C + \gamma_1^T C^* \gamma_1 + \gamma_2^T C^* \gamma_2 + \gamma_3^T C^* \gamma_3 \quad (2.3.11a)$$

$$= \begin{bmatrix} a_{11} + a_{22} + a_{33} & 0 & 0 \\ 0 & a_{11} + a_{22} + a_{33} & 0 \\ 0 & 0 & a_{11} + a_{22} + a_{33} \end{bmatrix} \quad (2.3.11b)$$

$$= \begin{bmatrix} \chi & 0 & 0 \\ 0 & \chi & 0 \\ 0 & 0 & \chi \end{bmatrix} \quad (2.3.11c)$$

where

$$\chi = \sum_{l=1}^{3} \sum_{k=1}^{3} |h_{l,k}|^2 \quad (2.3.12)$$

Note that if after the third retransmission, the result out channel matrix is always the same, regardless of the order of alternative chosen before. However, after the first and the second retransmission, two different corresponding off-diagonal terms are zeroed and two diagonal terms are enhanced and hence the performance of this HARQ will depend on the order of choosing any of these retransmissions. We will discuss this issue in the next section.

2.3.2 The SNR Criterion

For a three-element transmitter and three-element receiver system, the received signal from the first transmission can be presented with

$$r^{(1)} = Hs^{(1)} + n^{(1)} \tag{2.3.13}$$

where

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{2.3.14}$$

and $s^{(1)} = [s_1, s_2, s_3]^T$ is the transmitted sequence on the three elements, respectively.

After matched filtering at the receiver,

$$x^{(1)} = H^{\psi} H s^{(1)} + H^{\psi} n^{(1)} = C_1 s^{(1)} + H^{\psi} n^{(1)} \tag{2.3.15}$$

where $(\cdot)^{\psi}$ is the operation of conjugate transpose,

and following zero forcing, we get the vector,

$$\hat{s}_1 = C_1^{-1} x^{(1)} = s^{(1)} + C_1^{-1} H^{\psi} n^{(1)} \tag{2.3.16}$$

where

$$C_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12}^* & a_{22} & a_{23} \\ a_{13}^* & a_{23}^* & a_{33} \end{bmatrix} \tag{2.3.17}$$

If error is detected a request for one sequence alternative $s^{(i+1)}$, i.e. $i \in (1,2,3)$ according to the columns in (2.3.1) is sent to the transmitter. That is second transmission, R_2 is given by,

$$r^{(2)} = Hs^{(i+1)} + n^{(2)} = H\gamma_i s^{(1)*} + n^{(2)} \tag{2.3.18}$$

where the choice of i will depend on the SNR criterion. After conjugating $r^{(2)}$ and match filtering by multiplying by $\gamma_i^T H^T$ we obtain

$$x^{(2)} = \gamma_i^T C_1^* \gamma_i s^{(1)} + \gamma_i^T H^T n^{(2)*} \tag{2.3.19}$$

Combining $x^{(1)}$ and $x^{(2)}$ and after zero forcing, we obtain

$$\hat{s}_{1,2} = s + C_2^{-1}(H^{\Psi} n^{(1)} + \gamma_i^T H^T n^{(2)*}) \tag{2.3.20}$$

where,

$$C_2 = (C_1 + \gamma_i^T C_1^* \gamma_i) \tag{2.3.21}$$

To decide which is the best i to use, we use the maximum Signal to noise ratio (SNR) at the receiver.

The autocorrelation matrix of the noise

$$R_{NN} = \mathrm{E}\left\{ C_2^{-1}\left(H^{\Psi} n^{(1)} + \gamma_i^T H^T n^{(2)*}\right)\left[\left(H^{\Psi} n^{(1)} + \gamma_i^T H^T n^{(2)*}\right) C_2^{-1}\right]^{\Psi} \right\} \tag{2.3.22}$$

can be shown to equal:

$$R_{NN} = \sigma^2 C_2^{-\Psi} \tag{2.3.23}$$

where $n^{(1)}$ and $n^{(2)}$ assumed uncorrelated, and where σ^2 is the noise variance of the components of $n^{(1)}$ and $n^{(2)}$ assumed equal. Normalizing the signal power to 1, we calculate the SNR at each receiving antenna and for each of the alternative i.

1. For $i = 1$ (see (2.3.7))

$$C_2 = \begin{bmatrix} b_1 & 0 & a_{13} \\ 0 & b_1 & a_{23} \\ a_{13}^* & a_{23}^* & a_{33} \end{bmatrix} \qquad \text{where,} \qquad b_1 = a_{11} + a_{22} \tag{2.3.24}$$

and we would have at each branches of the receiver an SNR given by:

$$SNR_1^{(1)} = \frac{b_1^2 a_{33} - b_1\left(|a_{13}|^2 + |a_{23}|^2\right)}{\sigma^2\left(b_1\, a_{33} - |a_{23}|^2\right)} \tag{2.3.25}$$

$$SNR_1^{(2)} = \frac{b_1^2 a_{33} - b_1\left(|a_{13}|^2 + |a_{23}|^2\right)}{\sigma^2\left(b_1\, a_{33} - |a_{13}|^2\right)} \tag{2.3.26}$$

$$SNR_1^{(3)} = \frac{b_1^2 a_{33} - b_1\left(|a_{13}|^2 + |a_{23}|^2\right)}{\sigma^2 b_1^2} \tag{2.3.27}$$

2. For $i = 2$

$$C_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12}^* & b_2 & 0 \\ a_{13}^* & 0 & b_2 \end{bmatrix} \qquad \text{where,} \qquad b_2 = a_{22} + a_{33} \tag{2.3.28}$$

and we would have at each of the receiver branches a SNR given by:

$$SNR_2^{(1)} = \frac{b_2^2 a_{11} - b_2\left(|a_{13}|^2 + |a_{12}|^2\right)}{\sigma^2 b_2^2} \tag{2.3.29}$$

$$SNR_2^{(2)} = \frac{b_2^2 a_{11} - b_2\left(|a_{13}|^2 + |a_{12}|^2\right)}{\sigma^2\left(b_2 a_{11} - |a_{13}|^2\right)} \tag{2.3.30}$$

$$SNR_2^{(3)} = \frac{b_2^2 a_{11} - b_2\left(|a_{13}|^2 + |a_{12}|^2\right)}{\sigma^2\left(b_2 a_{11} - |a_{12}|^2\right)} \tag{2.3.31}$$

3. Similarly for $i = 3$

$$C_2 = \begin{bmatrix} b_3 & a_{12} & 0 \\ a_{12}^* & a_{22} & a_{23} \\ 0 & a_{23}^* & b_3 \end{bmatrix} \quad \text{where,} \quad b_3 = a_{11} + a_{33} \tag{2.3.32}$$

and we would have at each branch a SNR given by:

$$SNR_3^{(1)} = \frac{b_3^2 a_{22} - b_3\left(|a_{12}|^2 + |a_{23}|^2\right)}{\sigma^2\left(b_3 a_{22} - |a_{23}|^2\right)} \tag{2.3.33}$$

$$SNR_3^{(2)} = \frac{b_3^2 a_{22} - b_3\left(|a_{12}|^2 + |a_{23}|^2\right)}{\sigma^2 b_3^2} \tag{2.3.34}$$

$$SNR_3^{(3)} = \frac{b_3^2 a_{22} - b_3\left(|a_{12}|^2 + |a_{23}|^2\right)}{\sigma^2\left(b_3 a_{22} - |a_{12}|^2\right)} \tag{2.3.35}$$

Note that for each i the branches' SNR expressions have the same numerator which is also the determinant of C_2. We use this propriety to decide on the vector that will be sent next. That is we choose the vector $s^{(i)}$ which results in the highest numerator in its SNR, will be the one requested to be sent at next transmission R_2.

If after the second transmission R_2, there would be still detection error in the CRC code, then, third transmission R_3 would be needed. For its alternative selection we look at the received signal

$$\hat{s}_{1,2,3} = s + C_3^{-1}(H^{\Psi} n^{(1)} + \gamma_i^T H^T n^{(2)*} + \gamma_{i+1}^T H^T n^{(3)*}) \tag{2.3.36}$$

where

$$C_3 = (C_1 + \gamma_i^T C_1^* \gamma_i + \gamma_{i+1}^T C_1^* \gamma_{i+1}) \tag{2.3.37}$$

where i the specific alternative used for R_2 and $i+1$ is the one examined for R_3. Again it can be shown that the noise correlation matrix is given by

$$R_{NN} = \sigma^2 C_3^{-\Psi} \tag{2.3.38}$$

In this case we have also three different alternatives, but we can only decide between two of them, since we have already decided on one of the vectors in the previous step.

1. If the vector of sequence are $s^{(2)}$ and $s^{(3)}$ shown in (2.3.2c), then

$$C_3 = \begin{bmatrix} b_1 & 0 & a_{13} \\ 0 & b_1 + a_{33} & 0 \\ a_{13}^* & 0 & b_2 \end{bmatrix} \tag{2.3.39}$$

Then, the SNR in each of the receiver branches are given by:

$$SNR_1^{(1)} = \frac{b_1 b_2 - |a_{13}|^2}{\sigma^2 b_2} \tag{2.3.40}$$

$$SNR_1^{(2)} = \frac{a_{11} + a_{22} + a_{33}}{\sigma^2} \tag{2.3.41}$$

$$SNR_1^{(3)} = \frac{b_1 b_2 - |a_{13}|^2}{\sigma^2 b_1} \tag{2.3.42}$$

2. If the vector of sequence are $s^{(2)}$ and $s^{(4)}$

$$C_3 = \begin{bmatrix} b_1 + a_{33} & 0 & 0 \\ 0 & b_1 & a_{23} \\ 0 & a_{23}^* & b_3 \end{bmatrix} \tag{2.3.43}$$

Then,

$$SNR_2^{(1)} = \frac{a_{11} + a_{22} + a_{33}}{\sigma^2} \tag{2.3.44}$$

$$SNR_2^{(2)} = \frac{b_1 b_3 - |a_{23}|^2}{\sigma^2 b_3} \tag{2.3.45}$$

$$SNR_2^{(3)} = \frac{b_1 b_3 - |a_{23}|^2}{\sigma^2 b_1} \tag{2.3.46}$$

3. If the vector of sequence are $s^{(3)}$ and $s^{(4)}$

$$C_3 = \begin{bmatrix} b_3 & a_{12} & 0 \\ a_{12}^* & b_2 & 0 \\ 0 & 0 & b_1 + a_{33} \end{bmatrix} \tag{2.3.47}$$

Hence,

$$SNR_3^{(1)} = \frac{b_2 b_3 - |a_{12}|^2}{\sigma^2 b_2} \tag{2.3.48}$$

$$SNR_3^{(2)} = \frac{b_2 b_3 - |a_{12}|^2}{\sigma^2 b_3} \tag{2.3.49}$$

$$SNR_3^{(3)} = \frac{a_{11} + a_{22} + a_{33}}{\sigma^2} \tag{2.3.50}$$

Following the same criterion as in the previous step we will choose the column which gives us the highest numerator in the SNR. Note that to choose the highest numerator is also to choose the highest determinant of the matrix C_3.

It is important to note that we are not actually choosing the maximum SNR, since we do not consider the denominator as well. It would have better to find a mathematic expression for the best SNR at all branches for each retransmission. This would have been too complex to handle. The fact to decide the best SNR in one branch for the first repetition, then implies that we do not have the freedom to decide the best option for the next retransmission i.e. we check the SNR's for first retransmission and we see that the best vector of sequence is $s^{(2)}$, but if we check for the second retransmission the best vector of sequence are $s^{(3)}$ and $s^{(4)}$. Hence, it could be that we get the best performance for the first retransmission, but with the second retransmission we do not get better results than if we send a random combination i.e. the transmission of $s^{(i)}$, $i = 2,3,4$ is used at R_2 or R_3 randomly. With the determinant criterion, although that we do not get the best performance in each retransmission, simulations results show that we always have better performance in all the retransmissions in case that we compare with the random solution

2.3.3 Numerical Results

The performance of the algorithm presented in this section is simulated with a MIMO channel consisting of 3 transmit antennas and 3 receiver antennas (3x3 MIMO system). The channel is assumed invariant for a maximum of L transmissions (=4 in this case). The 9 channel gains are assumed i.i.d. complex Gaussian random variables (Raleigh Flat Fading) and with unit power. The size of the information bit packet is taken to be 522 bits. The total numbers of packets in the MonteCarlo simulation were 5000. Each packet is split in three equal parts (s_1, s_2, s_3). The decision that a packet is received correctly is taken after combining again the three parts.

In figure 2.9 the bit error rates are depicted versus E_b / N_0 for R_2 and R_3 using the proposed algorithm. R_1 presents the first transmission and R_4 shows the result after completing the transmissions. If error is detected after R_4, then the process is repeated or combined with data received from previous trials. For comparison we also added in the figure what is termed Random in which the transmission of $s^{(i)}$, $i = 2,3,4$ is used at R_2 or R_3 randomly.

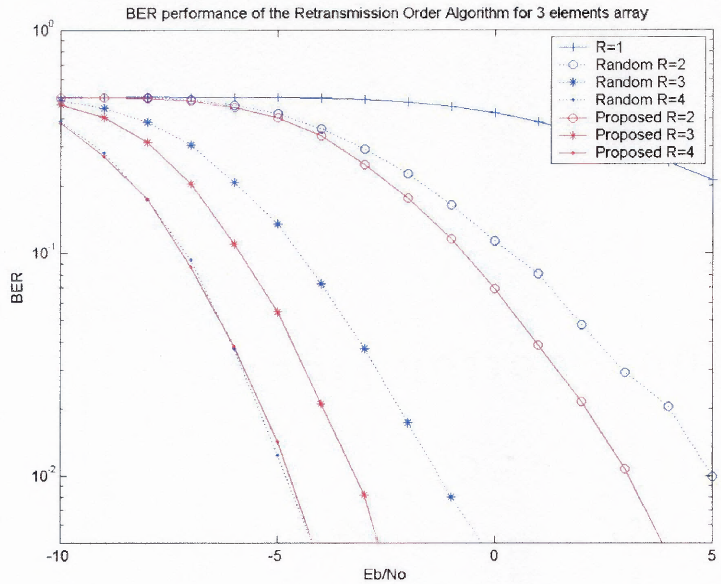

Figure 2.9 BER comparison when using the determinant criterion and random selection decision for 3 elements array

Note that for R_1 and R_4, the curves for the random solution and for the proposed algorithm is the same, since there is no degree of freedom for choosing different alternative at the first or the fourth transmission.

In the next figure we plot the throughput performance of the proposed algorithm. The maximum number of allowed transmission is 4, hence, if a fifth transmission is required, the packet is considered lost.

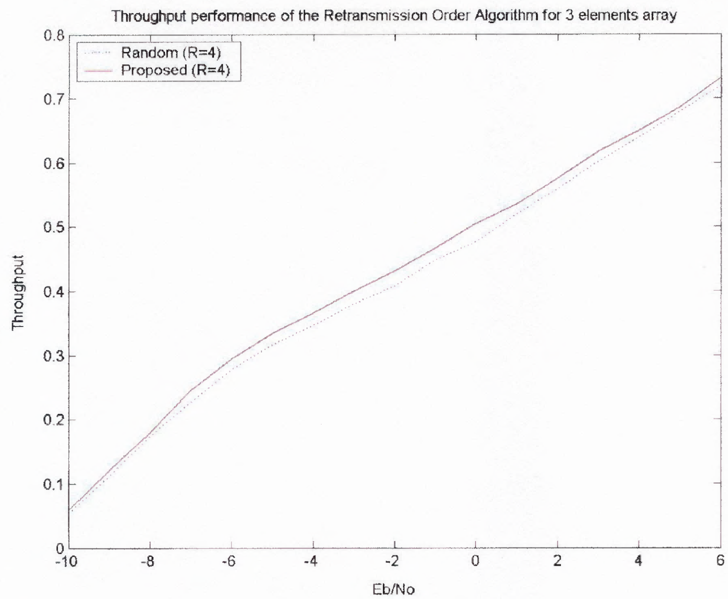

Figure 2.10 Throughput comparison when using the determinant criterion and random selection decision for 3 elements array

In this section we proposed an algorithm for packet order retransmission supporting HARQ for a MIMO system. This algorithm exploits the proprieties of the channel matrix to choose the best sequence of retransmission with the Multiple Alamouti Coding. Simulation shows that the BER performance of our algorithm is almost 2 *dBs* better than random retransmissions both for two and three retransmission in $3x3$ MIMO system. Also, the throughput performance of the proposed algorithm is better than random retransmissions. Note that the technique is valid only in a slow varying channel. Extension to the case of *NxM* MIMO channel will be shown in the next section.

2.4 Retransmission Order Algorithm for NxM Elements Array

In section 2.2 we proposed a new method for HARQ and in section 2.3 we suggest an algorithm by which, based on the measured channel matrix, the receiver can choose the order for transmitting the Alamouti coding on the different transmitting antennas. The goal of the algorithm is to decide which transmission would give the maximum Signal to Noise Ratio (SNR) in all receivers' branches. The expressions of each SNR after each retransmission were calculated and showed that such a maximum is related to the maximum of the determinant of the resulting channel matrix. This was done for $3x3$ MIMO system. In this section the selection algorithm is generalized to the case of NxM elements array. Clearly, the value of the determinant depends on the chosen order of sequences transmitted and the number of retransmissions used.

2.4.1 The Determinant Criterion

To depict the method of generalization of the algorithm, we consider the case of $4xM$ MIMO channel (extension to more transmitter antennas is straightforward).

For a four-element transmitter, if detection error occurred a request for ARQ will be for any of six alternatives for retransmissions following the first, $s^{(1)}$ according to the columns below

$$\begin{array}{ccccccc}
s^{(1)} & s^{(2)} & s^{(3)} & s^{(4)} & s^{(5)} & s^{(6)} & s^{(7)} \\
 & & & & & & \\
s_1 & -s_2^* & 0 & 0 & s_3^* & 0 & -s_4^* \\
s_2 & s_1^* & -s_3^* & 0 & 0 & s_4^* & 0 \\
s_3 & 0 & s_2^* & -s_4^* & -s_1^* & 0 & 0 \\
s_4 & 0 & 0 & s_3^* & 0 & -s_2^* & s_1^*
\end{array} \qquad (2.4.1)$$

As was done in the previous sections we can write each column of the retransmission alternative, as follows

$$s^{(i+1)} = \gamma_i s^{(1)*} \qquad \text{where } i = 1 \ldots 6 \qquad (2.4.2)$$

with

$$\gamma_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.4.3a) \qquad \gamma_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.4.3b)$$

$$\gamma_3 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.4.3c) \qquad \gamma_4 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.4.3d)$$

$$\gamma_5 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \quad (2.4.3e) \qquad \gamma_6 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.4.3f)$$

Following similar expression in the case of 3x3 elements, we define

$$C = H^{\Psi} H \qquad (2.4.4a)$$

$$= \begin{bmatrix} h_{11}^* & h_{21}^* & \cdots & h_{M1}^* \\ h_{12}^* & h_{22}^* & \cdots & h_{M2}^* \\ h_{13}^* & h_{23}^* & \cdots & h_{M3}^* \\ h_{14}^* & h_{24}^* & \cdots & h_{M4}^* \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ \vdots & \vdots & \vdots & \vdots \\ h_{M1} & h_{M2} & h_{M3} & h_{M4} \end{bmatrix} \qquad (2.4.4b)$$

$$= \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{pmatrix} \qquad (2.4.4c)$$

where h_{ji} is the channel gain between the transmit antenna i and the receive antenna j ($i = 1,2,3,4$ and $j = 1,2,...,M$). Also,

$$a_{kl} = \sum_{m=1}^{M} h_{mk}^{*} h_{ml} \qquad k = 1,....,4 \qquad \text{and} \qquad l = 1,...,4 \tag{2.4.5}$$

Note that if $k = l$, then

$$a_{kk} = \sum_{m=1}^{M} \left| h_{mk} \right|^2 \tag{2.4.6}$$

As a result of using any of the column in (2.4.1) in the first retransmission, we calculate the determinant of C_1 for each matrix γ_i, where

$$C_1 = C + \gamma_i^T C^* \gamma_i \tag{2.4.7}$$

The column $s^{(i)}$ that results in the highest determinant for C_1, will be the first one to be sent at next transmission R_2. If after the second transmission R_2, there would be still detection error in the CRC code, then, in the third transmission R_3, we will calculate which of the columns will gives us the highest determinant for the matrix C_2, defined as

$$C_2 = C + \gamma_i^T C^* \gamma_i + \gamma_j^T C^* \gamma_j \tag{2.4.8}$$

where i is the first vector selected and j is the second one. Note that j belongs a set of integers in which i is not included.

If after the third transmission R_3, there would be still detection error, we check the determinant of the next C_3.

$$C_3 = C + \gamma_i^T C^* \gamma_i + \gamma_j^T C^* \gamma_j + \gamma_k^T C^* \gamma_k \tag{2.4.9}$$

where $k \in \{1,2,3,4,5,6\} - \{i, j\}$

If a fourth transmission R_4 is needed, we decided according to the determinant C_4

$$C_4 = C + \gamma_i^T C^* \gamma_i + \gamma_j^T C^* \gamma_j + \gamma_k^T C^* \gamma_k + \gamma_l^T C^* \gamma_l \tag{2.4.10}$$

where $l \in \{1,2,3,4,5,6\} - \{i, j, k\}$

and so on for the fifth transmission R_5,

$$C_5 = C + \gamma_i^T C^* \gamma_i + \gamma_j^T C^* \gamma_j + \gamma_k^T C^* \gamma_k + \gamma_l^T C^* \gamma_l + \gamma_m^T C^* \gamma_m \tag{2.4.11}$$

where $m \in \{1,2,3,4,5,6\} - \{i, j, k, l\}$

In case that we have to send the sixth transmission R_6,

$$C_6 = C + \sum_{\substack{n=1 \\ n \neq p}}^{6} \gamma_n^T C^* \gamma_n \tag{2.4.12}$$

where p is the last column that we will send, if a seventh transmission R_7 is required. Note that for the last transmission there is no need for any criterion checking as just one vector left. If error is detected after R_7, then the process is repeated or combined with offered data from previous trials.

2.4.2 Numerical results

To show the performance of the proposed HARQ scheme using the corresponding selection algorithm, we will use it in different MIMO scenarios with different N and M. The first example we examine a $4x4$ MIMO system. The data used in the simulations for this scenario was same as in the previous case of 3x3. The bit error rates are depicted in figure 2.11 versus E_b / N_0 for R_2 to R_6 using the proposed scheme and selection algorithm. R_1 presents the first transmission and R_7 shows the result after completing the transmissions. For comparison also added in the figure what is termed Random in which the selection of $s^{(i+1)}, i = 1...6$ to send R_2 to R_6 were done randomly.

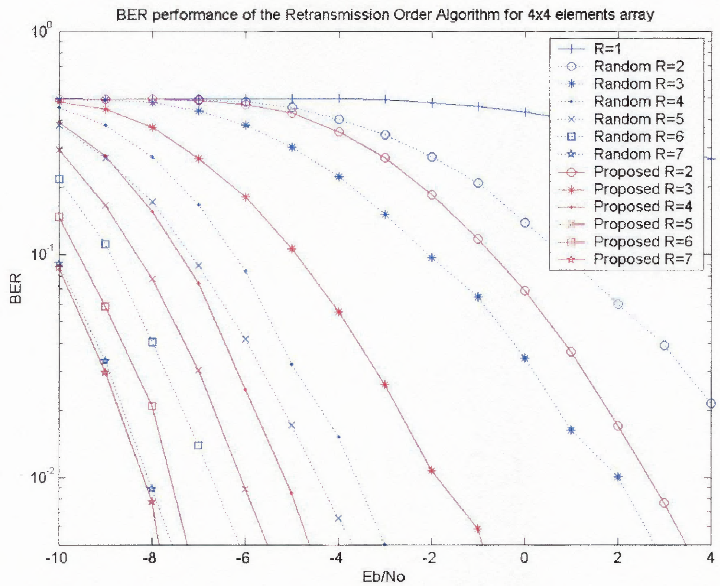

Figure 2.11 BER comparison when using the Determinant criterion and random selection decisions in a 4x4 MIMO system

Note that as expected for R_1 and R_7 the performance using random selection or the proposed determinant criterion is the same, since there is no degree of freedom for choosing different sequences.

We also depict in the next figure the throughput performance of the proposed scheme using the determinant criterion for the $4x4$ MIMO system. For comparison we also added in the figure throughput with random selection. For finding throughput we took as maximum number of allowed transmission 7, that is, if an eight transmission is required, the packet is considered lost.

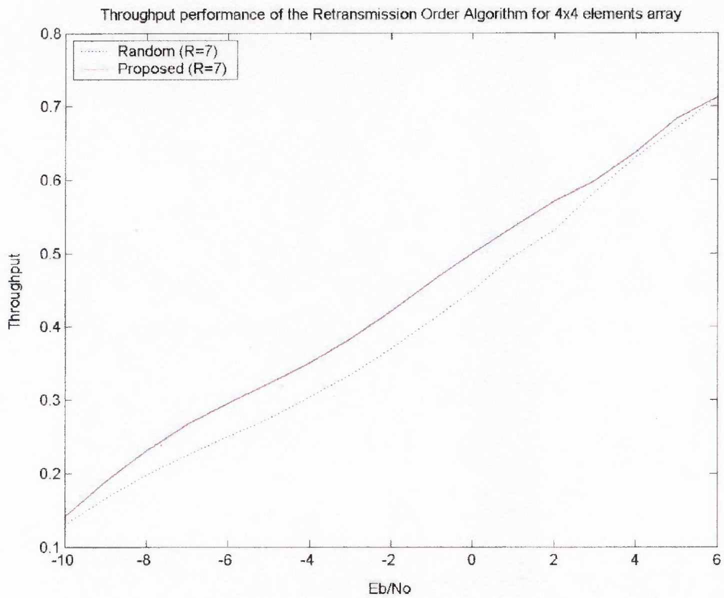

Figure 2.12 Throughput comparison when using the Determinant criterion and random selection decisions in a 4x4 MIMO system

As in the case of $3x3$ MIMO's system it is shown that the proposed selection criterion worked quite well. In fact comparing to the result in section 2.3.3, with the current case of 4 elements the results are even better. For example, in comparison to the random selection there is an improvement of 4 *dBs* for R_3, at $BER = 5.10^{-3}$, and approximately 2 *dBs* with the fourth retransmissions R_4. This improvement is also reflected in the throughput, where the proposed scheme depicts 2 *dBs* better.

In the next set of figures, the bit error rates versus E_b / N_0 are shown for different NxM arrangement. The first two figures depict the performance for R_2 and R_3 using the proposed scheme in two different NxM MIMO system, $3x4$ and $3x5$. Added to the figures R_1, the first transmission and R_4 which shows the result after completing the transmissions. Again for comparison we added to these figures performance of the multiple Alamouti coding scheme when alternative selection in R_2 and R_3 were done randomly. The third figure compares the performance for the MIMO $3x3$, $3x4$ and $3x5$ when we use the determinant criterion for alternative selection.

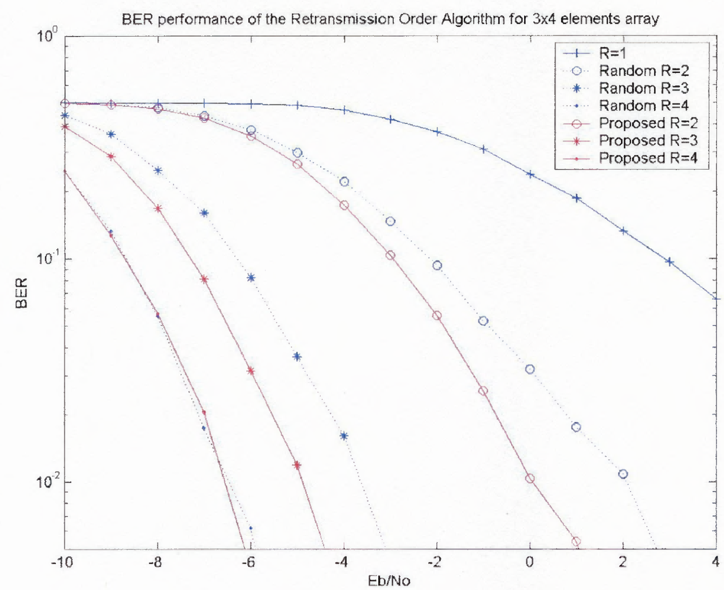

Figure 2.13 BER comparison when using the Determinant criterion and random selection decisions in a 3x4 MIMO system

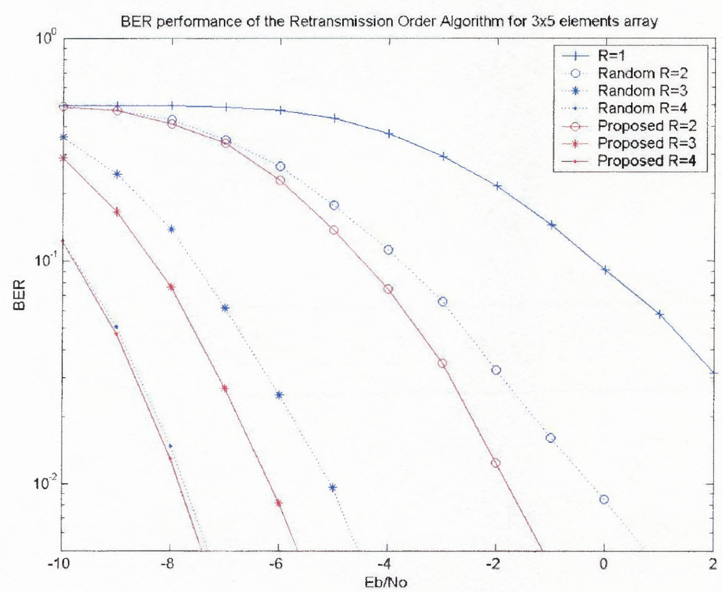

Figure 2.14 BER comparison when using the Determinant criterion and random selection decisions in a 3x5 MIMO system

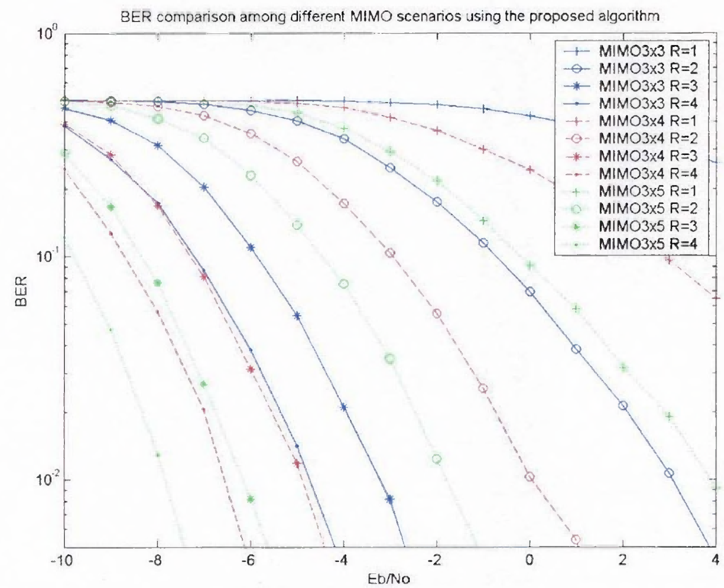

Figure 2.15 BER comparison among the three different MIMO scenarios 3x3, 3x4 and 3x5 using the Determinant criterion for alternative selection

Correspondingly, in the next three figures we plot the throughput performance for the three scenarios described above. Note that in all the cases we use as maximum number of allowed transmissions 4.

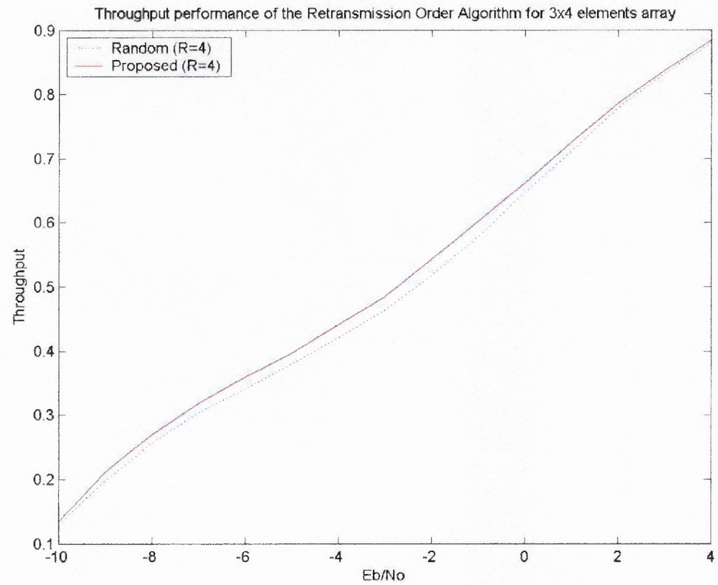

Figure 2.16 Throughput comparison when using the Determinant criterion and random selection decisions in a 3x4 MIMO system

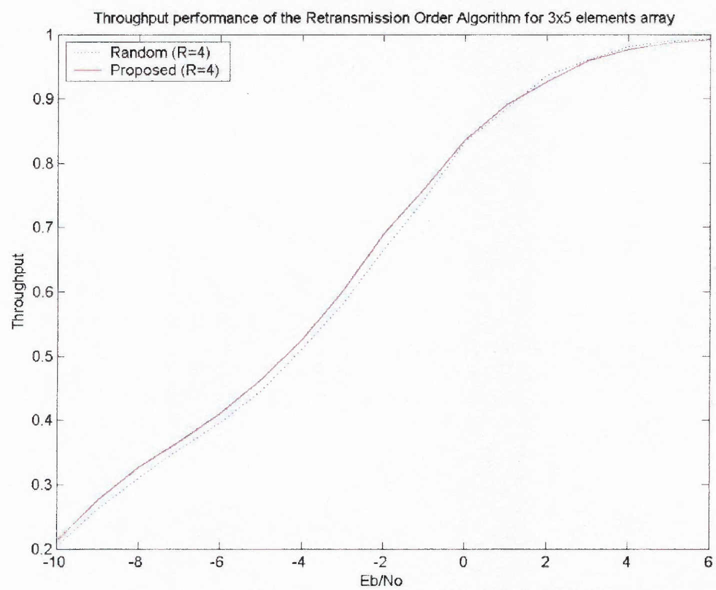

Figure 2.17 Throughput comparison when using the Determinant criterion and random selection decisions in a 3x5 MIMO system

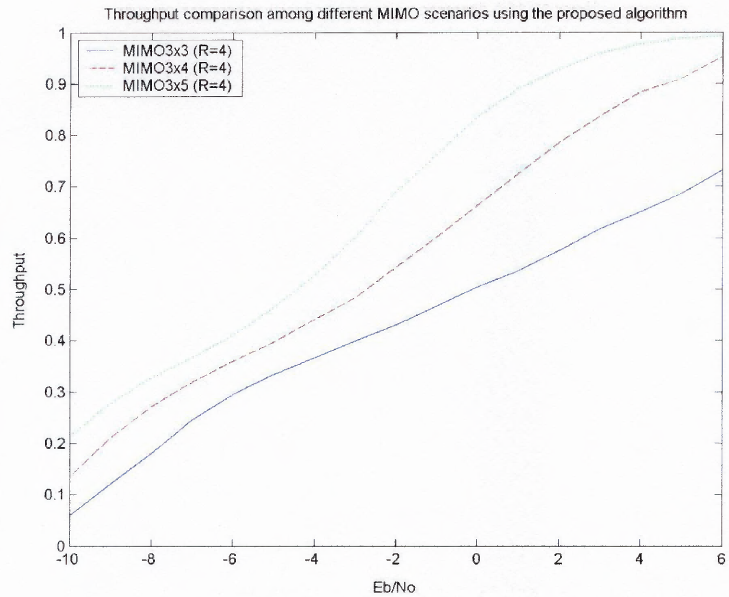

Figure 2.18 Throughput comparison among the three different MIMO scenarios, 3x3, 3x4, and 3x5 using the Determinant criterion for alternative selection

The next figure (Figure 2.19) depicts the bit error rates versus E_b / N_0 for the case of 4 elements transmitters and 5 elements receivers. Following the same presentation as before for R_2 to R_6 we used the proposed determinant criterion for selection. R_1 presents the first transmission and R_7 shows the result after completing the transmissions. For comparison also added in the figure the performance results when the alternative for R_2 to R_6 were selected randomly. Figure 2.20 compares the performance between the cases for $4x4$ and $4x5$ when using the determinant criterion.

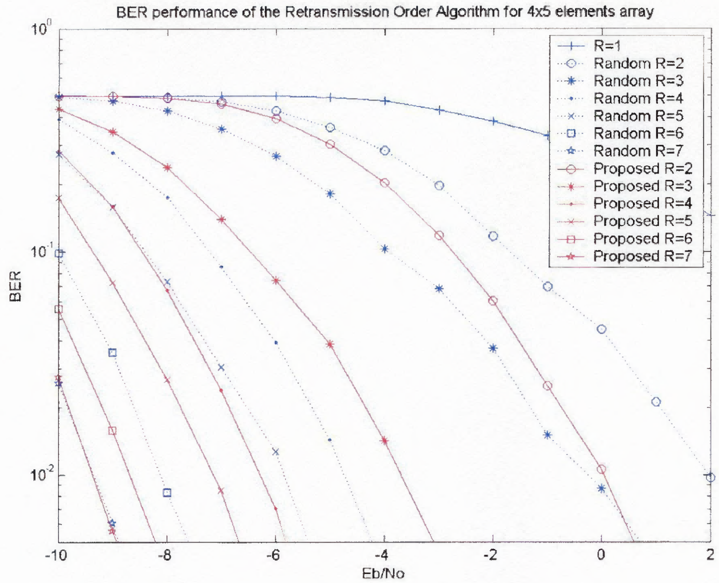

Figure 2.19 BER comparison when using the Determinant criterion and random selection decisions in a 4x5 MIMO system

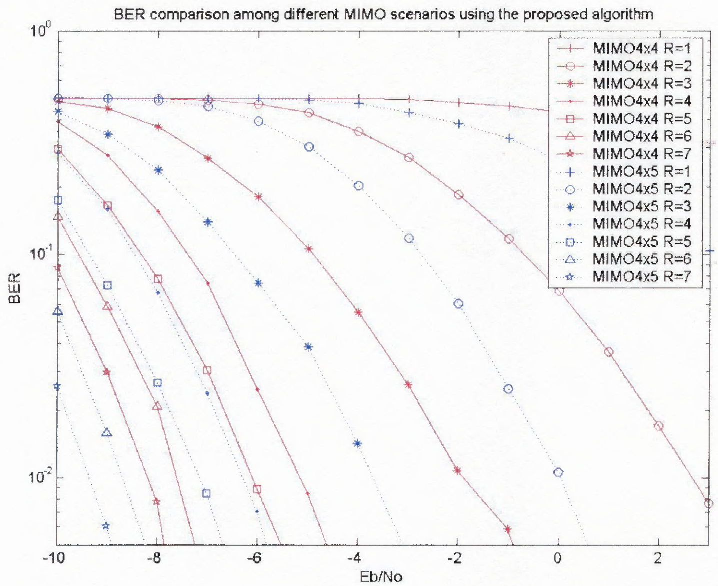

Figure 2.20 BER comparison between the 4x4 and 4x5 MIMO using the Determinant criterion

The last two figures depict the throughput performance for the corresponding scenarios described in Figures 2.19 and 2.20. For these cases, we used as maximum number of allowed transmissions 7.

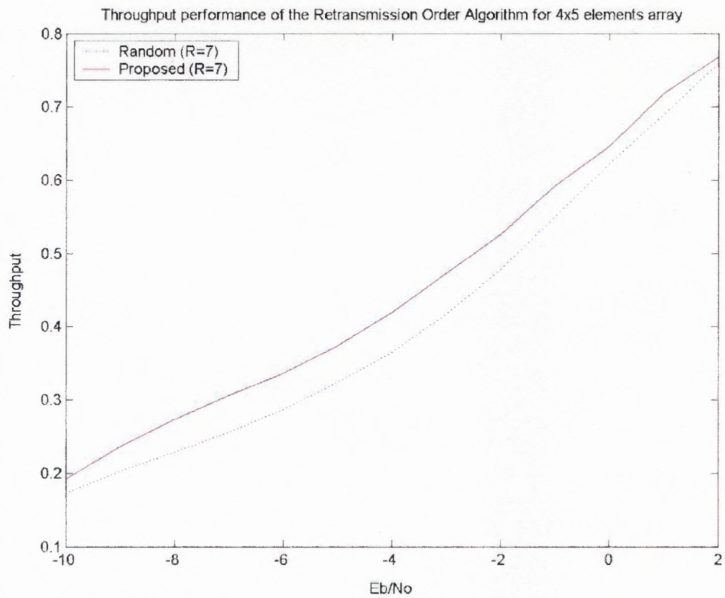

Figure 2.21 Throughput comparison when using the Determinant criterion and random selection decisions in a 4x5 MIMO system

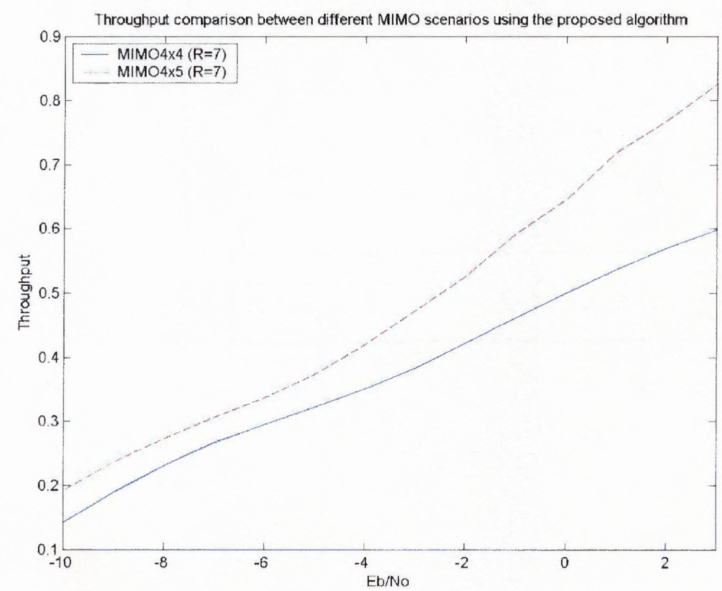

Figure 2.22 Throughput comparison between the 4x4 and 4x5 MIMO using the Determinant criterion

In previous sections, it was shown mathematically and with simulation that the proposed H-ARQ algorithm worked quite well for a 3x3 MIMO system. In this section, the H-ARQ algorithm Determinant criterion was extended to the general case NxM . In particular, the performance of the proposed scheme for the cases of $3x4$, $3x5$ and $4x5$ MIMO are shown in figures 2.13, 2.14 and 2.19 respectively. It was shown that if we increase the number of receiving antennas (M), the performance in terms of BER and throughput gains a considerable improvement. The reason is due to the fact that the number of channel links are increased (there are NM links). Particularly from (2.4.6) we note that the diagonal terms of the resulting channel matrix are increased as we increase the value of M , meaning that the Multiple Almouti coding exploits these increases of links making the received signal stronger. For example, in comparing the case $3x3$ and $3x5$ (see figure 2.15) we found an improvement of almost $5\ dBs$ for R_2, at $BER = 5.10^{-3}$. Note that increasing M do not affect the proposed alternative selection for packet retransmission, since the proposed algorithm depends only on the resulting matched filtered and combined cross-correlation matrix, which remains of dimension NxN. Note that M must be equal or higher than N, since the resulting channel matrix combining of dimension NxN after each retransmission would have rank $\min(N,M)$, which is a singular matrix if N is higher than M, and is not invertible as it is required for zero forcing.

CHAPTER 3

THE DETERMINANT CRITERION FOR ANOTHER HARQ SCHEME AND PERFORMANCE COMPARISON WITH MULTIPLE ALAMOUTI HARQ APPROACH

In previous chapter we proposed the Multiple Alamouti for HARQ and suggested an algorithm by which, based on the measured channel matrix, the receiver can choose the order for transmitting subsequent sequences on the different antennas transmitter. In this chapter we present a different HARQ retransmission and combining scheme. This other scheme was suggested by LG [14] for the cases of 3 and 4 transmitter antennas. The main difference between the two schemes is that in the LG solution, the retransmitted sequences are applied to all transmitting antennas instead of using only two antennas at a time and zeros on all others.

Truly sending zero sequence on some antennas will make the Multiple Alamouti less effective that the LG approach. Therefore we will propose for the former what we term power normalization before we make performance comparison with LG scheme. With such normalization we will have unit average power over all antennas in both schemes. The performance the two schemes will be compared both for the case of 3 elements array and 4 elements array. Such normalization will also affect order decision's criterion which is based in the determinant, which has to be modified.

3.1 The LG Scheme

In [14] a different retransmission scheme is proposed for supporting HARQ for MIMO systems. As it is done in [14] two cases will be handled: 3 transmitter antennas and 4 transmitter antennas. We will make a brief description of these two cases for LG scheme, then we will explain how we can use the proposed selection algorithm in this scheme and finally we will present the performance obtained with the proposed algorithm.

3.1.1 The Case of 3 Transmitting Antennas

In the HARQ model described in section 2.1, if detection error occurred a request in a sequence of two retransmissions following the first transmission, $s^{(1)}$ according to three different alternatives described in the following

Table 3.1 Retransmissions alternatives for the case of 3 elements in LG proposal

Initial transmission	Odd retransmission	Even retransmission
$s^{(1)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$	$s_{ALT1}^{(odd)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ s_3^* \end{bmatrix}$	$s_{ALT1}^{(even)} = \begin{bmatrix} s_3 \\ s_1 \\ s_2 \end{bmatrix}$
	$s_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix}$	$s_{ALT2}^{(even)} = \begin{bmatrix} s_2 \\ s_3 \\ s_1 \end{bmatrix}$
	$s_{ALT3}^{(odd)} = \begin{bmatrix} s_1^* \\ -s_3^* \\ s_2^* \end{bmatrix}$	$s_{ALT3}^{(even)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$

In alternative 1 retransmission sub-packet and initial transmission sub-packet forms a Space Time Transmit antenna Diversity (STTD) structure with antenna 1 and antenna 2. With alternative 2 and initial sub-packet form STTD structure with antenna 1 and antenna 3, while with alternative 3 and initial sub-packet forms of STTD structure with antenna 2 and antenna 3. Note that instead of sending a zero in the antenna that does not form STTD structure with the initial transmission, the remaining sub-packet is sent conjugated. Then, the 2nd and 3rd re-transmissions follows the same structure as the previous two, and it can be further combined in energy with the result of the first pair of STTD. However, it can further exploited the diversity gain from the 2nd and 3rd retransmissions. In this case it achieves the full diversity for 3 transmit antennas. Note that once certain alternative is selected at first retransmission, the second retransmission follows the first.

We will use the same transformation presented in the previous chapter to combine the retransmitted packets. We can write the columns of the retransmission sequences, as follows,

$$s_{ALT1}^{(odd)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ s_3^* \end{bmatrix} = \gamma_1 s^{(1)*} \qquad (3.1.1a) \qquad s_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix} = \gamma_2 s^{(1)*} \qquad (3.1.1b)$$

$$s_{ALT3}^{(odd)} = \begin{bmatrix} s_1^* \\ -s_3^* \\ s_2^* \end{bmatrix} = \gamma_3 s^{(1)*} \qquad (3.1.1c)$$

where,

$$\gamma_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.1.2a) \qquad \gamma_2 = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad (3.1.2b)$$

$$\gamma_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (3.1.2c)$$

also,

$$s_{ALT1}^{(even)} = \begin{bmatrix} s_3 \\ s_1 \\ s_2 \end{bmatrix} = \lambda_1 s^{(1)} \qquad (3.1.3a) \qquad s_{ALT2}^{(even)} = \begin{bmatrix} s_2 \\ s_3 \\ s_1 \end{bmatrix} = \lambda_2 s^{(1)} \qquad (3.1.3b)$$

$$s_{ALT3}^{(even)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \lambda_3 s^{(1)} \qquad (3.1.3c)$$

where,

$$\lambda_1 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad (3.1.4a) \qquad \lambda_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad (3.1.4b)$$

$$\lambda_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.1.4c)$$

If we assume for example we select alternative 1, when after first re-transmission, matched filtering and zero forcing, the combined signal can be written as follows,

$$\hat{s}_{1,2} = s + C_2^{-1}(H^{\psi} n^{(1)} + \gamma_1^T H^T n^{(2)*}) \qquad (3.1.5)$$

where $C_2 = (C_1 + \gamma_1^T C_1^* \gamma_1)$ with

$$C_1 = H^{\psi} H = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12}^* & a_{22} & a_{23} \\ a_{13}^* & a_{23}^* & a_{33} \end{bmatrix} \qquad (3.1.6)$$

when it can be shown that

$$C_2 = \begin{bmatrix} a_{11} + a_{22} & 0 & a_{13} + a_{23}^* \\ 0 & a_{22} + a_{11} & a_{23}^* - a_{13}^* \\ a_{13}^* + a_{23} & a_{23} - a_{13} & 2a_{33} \end{bmatrix} \qquad (3.1.7)$$

If a second re-transmission is required, then

$$\hat{s}_{1,2,3} = s + C_3^{-1}(H^{\psi} n^{(1)} + \gamma_1^T H^T n^{(2)*} + \lambda_1^T H^{\psi} n^{(3)*}) \qquad (3.1.8)$$

where $C_3 = (C_1 + \gamma_1^T C_1^* \gamma_1 + \lambda_1^T C_1 \lambda_1)$. Note that in the second re-transmission the signal is not conjugate, hence for matched filtering we do not conjugate. It can be shown that,

$$C_3 = \begin{bmatrix} a_{11} + 2a_{22} & a_{23} & a_{13} + a_{23}^* + a_{12}^* \\ a_{23}^* & a_{22} + a_{11} + a_{33} & a_{23}^* \\ a_{13}^* + a_{23} + a_{12} & a_{23} & 2a_{33} + a_{11} \end{bmatrix} \quad (3.1.9)$$

Note that we lose the diagonal matrix that we had in the previous scheme. Nevertheless we show an increase data throughput using all the antennas at each transmission instead on only two antennas. That is we are facing compromise between cancellation of interfered signal (off-diagonal terms) and an increase in the data throughput due to increase in diagonal terms. Later, simulations results will show which scheme performs better.

3.1.2 The Case of 4 Transmitter Antennas

With the proposed LG HARQ scheme for 4 transmitter antenna system, the first re-transmission (odd) sub-packet combined with the initial transmission constitutes a double STTD. The 2nd re-transmissions (even) follow the structure of the previous sequences which will further be combined in energy with the result of the first combined pair of double STTD. However, in order to further exploit the diversity gain from the 2nd and 3rd re-transmissions and obtain a full diversity for 4 transmit antennas, there is a swap of the s_1, s_2 and s_3, s_4.

Accordingly the different alternatives of LG for the case of $4x4$ antennas are given in following set of sequences:

Table 3.2 Retransmissions alternatives for the case of 4 elements in LG proposal

Initial transmission	Odd retransmission	Even retransmission
$s^{(1)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$	$s_{ALT1}^{(odd)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ -s_4^* \\ s_3^* \end{bmatrix}$	$s_{ALT1}^{(even)} = \begin{bmatrix} s_3 \\ -s_4 \\ -s_1 \\ s_2 \end{bmatrix}$
	$s_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ -s_4^* \\ s_1^* \\ s_2^* \end{bmatrix}$	$s_{ALT2}^{(even)} = \begin{bmatrix} s_2 \\ s_1 \\ s_4 \\ s_3 \end{bmatrix}$
	$s_{ALT3}^{(odd)} = \begin{bmatrix} -s_4^* \\ -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix}$	$s_{ALT3}^{(even)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$

As we use for the case of 3 antennas, we can relate all these alternatives in a matrix form to the first transmission.

$$s_{ALT1}^{(odd)} = \begin{bmatrix} -s_2^* \\ s_1^* \\ -s_4^* \\ s_3^* \end{bmatrix} = \gamma_1 s^{(1)*} \quad (3.1.10a) \qquad s_{ALT2}^{(odd)} = \begin{bmatrix} -s_3^* \\ -s_4^* \\ s_1^* \\ s_2^* \end{bmatrix} = \gamma_2 s^{(1)*} \quad (3.1.10b)$$

$$s_{ALT3}^{(odd)} = \begin{bmatrix} -s_4^* \\ -s_3^* \\ s_2^* \\ s_1^* \end{bmatrix} = \gamma_3 s^{(1)*} \quad (3.1.10c)$$

where,

$$\gamma_1 = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.1.11a) \qquad \gamma_2 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.1.11b)$$

$$\gamma_3 = \begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (3.1.11c)$$

Also for even retransmissions

$$s_{ALT1}^{(even)} = \begin{bmatrix} s_3 \\ -s_4 \\ -s_1 \\ s_2 \end{bmatrix} = \lambda_1 s^{(1)} \quad (3.1.12a) \qquad s_{ALT2}^{(even)} = \begin{bmatrix} s_2 \\ s_1 \\ s_4 \\ s_3 \end{bmatrix} = \lambda_2 s^{(1)} \quad (3.1.12b)$$

$$s_{ALT3}^{(even)} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \lambda_3 s^{(1)} \quad (3.1.12c)$$

where,

$$\lambda_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.1.13a) \qquad \lambda_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.1.13b)$$

$$\lambda_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1.13c)$$

If we combine the received signals we get the same equations as in (3.1.5) and (3.1.8) for the first and the second retransmission respectively. If we choose alternative 1, for the first retransmission we get the following matched and combined cross-correlation matrix C_2 ,

$$C_2 = \begin{pmatrix} a_{11}+a_{22} & 0 & a_{13}+a_{24}^* & a_{14}-a_{23}^* \\ 0 & a_{11}+a_{22} & a_{23}-a_{14}^* & a_{24}+a_{13}^* \\ a_{13}^*+a_{24} & a_{23}^*-a_{14} & a_{33}+a_{44} & 0 \\ a_{14}^*-a_{23} & a_{24}^*+a_{13} & 0 & a_{33}+a_{44} \end{pmatrix} \qquad (3.1.14)$$

and for the second retransmission, we can write C_3 as

$$C_3 = \begin{pmatrix} a_{11}+a_{22}+a_{33} & -a_{34} & a_{13}-a_{13}^*+a_{24}^* & a_{14} \\ -a_{34}^* & a_{11}+a_{22}+a_{44} & a_{23} & a_{24}+a_{13}^* \\ a_{13}^*-a_{13}+a_{24} & a_{23}^* & a_{11}+a_{33}+a_{44} & -a_{12} \\ a_{14}^* & a_{24}^*+a_{13} & -a_{12}^* & a_{22}+a_{33}+a_{44} \end{pmatrix} \qquad (3.1.15)$$

It can also be shown that, like in the case of three elements, the channel matrix obtained after combining all the received packets is not diagonal. However, the data throughput has a considerable increment due to the fact that we are using the 4 antennas diversity. Simulation result will show later the performance of this scheme and it compares to that of multiple Alamouti.

3.1.3 The Determinant Criterion for the LG Scheme

In [14] it is not mentioned what criterion was used to decide on the different alternatives. We suggest the determinant criterion to select the alternatives at the odd retransmission. At the even retransmissions we use the same alternatives as the previous selection.

After the first transmission (R_1), there is a need for a retransmission (R_2), we check the following determinants to decide on one of the alternatives

$$\det\left|C+\gamma_i^T C^* \gamma_i\right| \qquad i=1,2,3 \qquad (3.1.16)$$

where C is the channel matrix that we get after the match filter. Note that the determinant depends only on the channel matrix and the transformation matrix γ_i of the three alternatives that were defined earlier. The alternative which give us the highest value in its determinant, it will be selected for the retransmission.

Then, if a third transmission is required (R_3), we do not make any selection, as we use the alternative i of the previous transmission. In the case that we need a fourth transmission (R_4), the new determinants that we have to check to decide on one of the alternatives is given by

$$\det\left|C + \gamma_k^T C^* \gamma_k + \lambda_k^T C \lambda_k + \gamma_i^T C^* \gamma_i\right| \qquad (3.1.17)$$

where k is the alternative that we chose before, and $i = 1,2,3$. Extension to more retransmissions is straightforward.

Using the same data as in simulations performed in previous sections, the performance of this HARQ scheme is shown in the next figures for both cases 3 and 4 transmitter antennas. The bit error rates are depicted versus E_b / N_0. R_i presents the i^{th} transmission, where $i = 1..4$ in figure 3.1 and $i = 1..7$ in figure 3.2. For comparison also added in the figure what is termed Random in which the selection of one of the alternatives is done randomly.

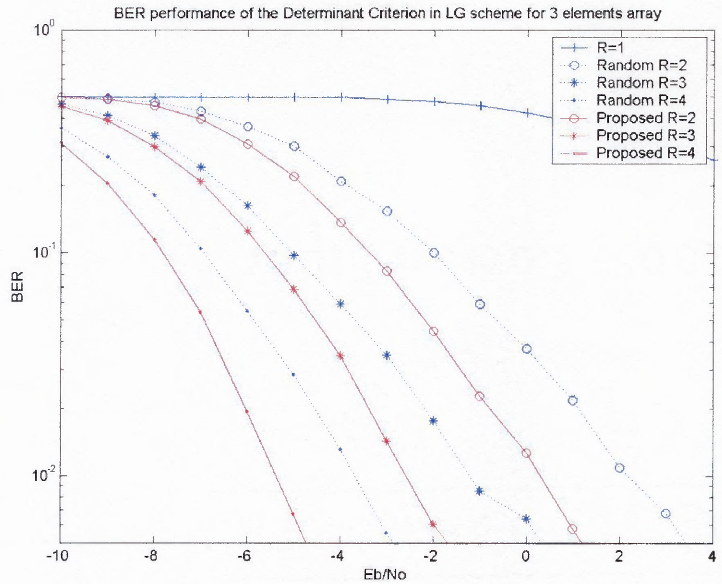

Figure 3.1 BER performance of LG HARQ scheme for which the Determinant Criterion was used for a 3x3 MIMO system

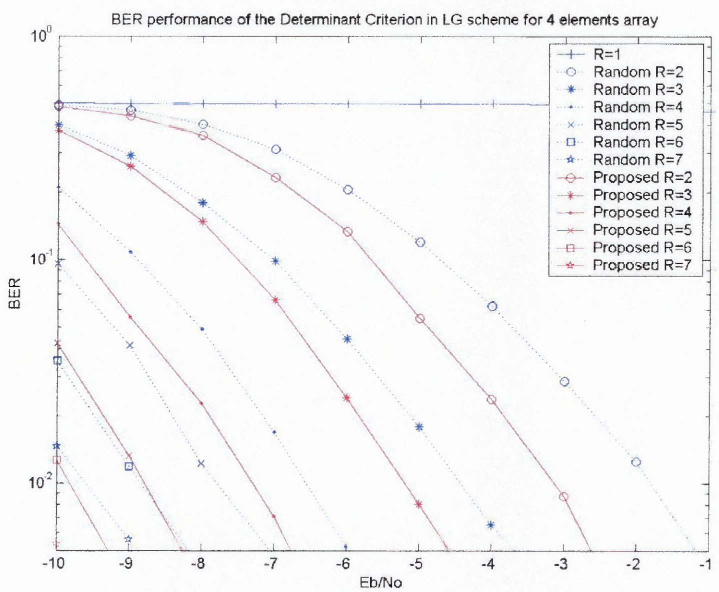

Figure 3.2 BER performance of LG HARQ scheme for which the Determinant Criterion was used for a 4x4 MIMO system

In figures 3.3 and 3.4 we plot respectively for 3 and 4 elements antennas the throughput performance of the proposed LG scheme using the determinant criterion. For comparison also added in the figure the throughput performance using random selecting instead. The maximum number of transmission allowed is 4 (for the case of 3 elements array) and 7 (for the case of 4 elements array).

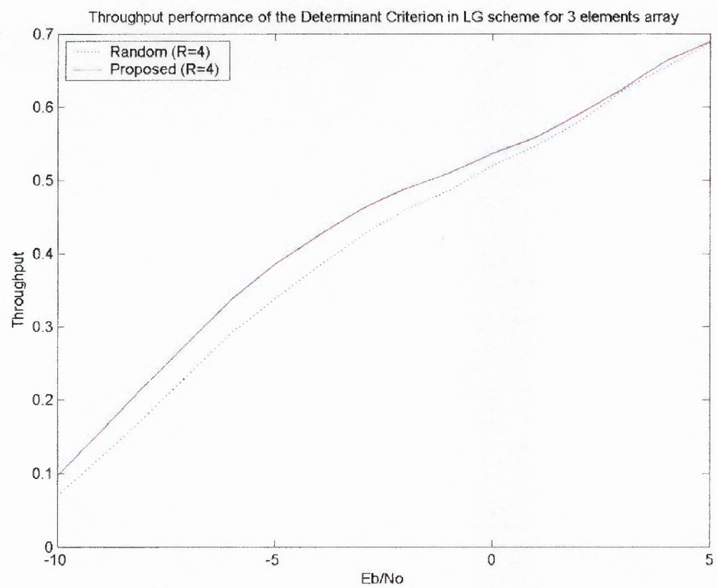

Figure 3.3 Throughput performance of LG HARQ for which the Determinant Criterion was used in a 3x3 MIMO system

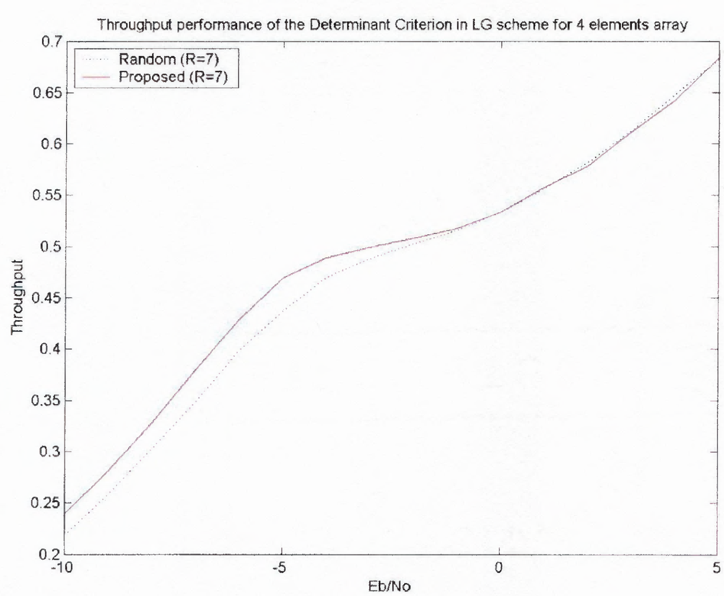

Figure 3.4 Throughput performance of LG HARQ for which the Determinant Criterion was used in a 4x4 MIMO system

As was expected the determinant criterion also work quite well in LG scheme. We can note that for both cases of 3 and 4 elements antenna we obtain an improvement in terms of BER and throughput in comparison if we use the determinant criterion. The next step will be to compare using the determinant criterion the performance of LG scheme with the multiple Alamouti HARQ scheme presented earlier. However, since in the later we are using in each retransmission only two antennas we must first do power normalization. Next section shows how to normalize the power and how this normalization affects the determinant criterion.

3.2 Multi-Alamouti Hybrid ARQ with Power Normalization

As stated earlier in the Multi-alamouti HARQ scheme, the transmitted power in retransmissions is lower than the power used in the initial transmission. That results in unfair comparison between Multi-alamouti HARQ and other schemes such as LG. In order to solve this problem, one can simply scale the power in the retransmissions so that the same total power is always used.

Consider the $3xM$ case with one retransmission (extension to more retransmissions and more element antennas is straightforward). For the first transmission, the received signal is given by

$$r^{(1)} = Hs^{(1)} + n^{(1)} \tag{3.2.1}$$

The power normalized retransmitted signal

$$r^{(2)*} = \sqrt{\frac{3}{2}} H^{*} \gamma_1 s^{(1)} + n^{(2)*} \tag{3.2.2}$$

For the first transmission the received signal after matched filter

$$x^{(1)} = H^{\psi} H s^{(1)} + H^{\psi} n^{(1)} = C_1 s^{(1)} + H^{\psi} n^{(1)} \tag{3.2.3}$$

For the received signal from the first retransmitted and after matched filter and normalization, we have

$$x^{(2)} = \sqrt{\frac{2}{3}}\gamma_1^T H^T \left(\sqrt{\frac{3}{2}} H^* \gamma_1 s^{(1)} + n^{(2)*} \right) \tag{3.2.4a}$$

$$= \gamma_1^T C_1^* \gamma_1 s^{(1)} + \sqrt{\frac{2}{3}}\gamma_1^T H^T n^{(2)*} \tag{3.2.4b}$$

Then, after soft combining

$$\hat{s}_{1,2} = \left(C_1 + \gamma_1^T C_1^* \gamma_1 \right) s^{(1)} + H^{\psi} n^{(1)} + \sqrt{\frac{2}{3}}\gamma_1^T H^T n^{(2)*} \tag{3.2.5}$$

After zero forcing:

$$\hat{s}_{1,2} = s^{(1)} + C_2^{-1}\left(H^{\psi} n^{(1)} + \sqrt{\frac{2}{3}}\gamma_1^T H^T n^{(2)*} \right) \tag{3.2.6}$$

where $C_2 = C_1 + \gamma_1^T C_1^* \gamma_1$.

The scheme maintains its cancellation properties (which is what makes it different from all other schemes). In addition, normalization, equivalently causes the noise reduction by a factor of $\sqrt{2/3}$ in the retransmissions. However, the autocorrelation matrix of the noise is not $\sigma^2 C_2^{-\psi}$ as before but rather more complicated expression.

The impact of such a change in noise correlation on the selection criterion is as follows: The determinant criterion is based on the SNR, which a function of the noise autocorrelation matrix. Using the proposed power normalization this matrix is given by,

$$R_{NN} = \mathrm{E}\left\{ \left[C_2^{-1}\left(H^{\Psi} n^{(1)} + \sqrt{\frac{2}{3}}\gamma_1^T H^T n^{(2)*} \right) \right] \left[C_2^{-1}\left(H^{\Psi} n^{(1)} + \sqrt{\frac{2}{3}}\gamma_1^T H^T n^{(2)*} \right) \right]^{\Psi} \right\} \tag{3.2.7}$$

It can be shown that

$$R_{NN} = \sigma^2 C_2^{-1}\left(C_1 + \frac{2}{3}\gamma_1^T C^* \gamma_1 \right)\left(C_2^{-1}\right)^{\psi} \tag{3.2.8}$$

which differs from the previous $R_{NN} = \sigma^2 C_2^{-\psi}$. The determinant criterion is still useful because $Det|C_2|$ is a common factor in all diagonal elements of R_{NN}. In order to apply the same criterion as before, we can re-write (3.2.8) as:

$$R_{NN} = \sigma^2 \left(C_x^{-1}\right)^{\psi} \tag{3.2.9}$$

where now

$$C_x = C_2^{\psi}\left(C_1^{\psi} + \frac{2}{3}\gamma_1 C_1^T \gamma_1^T \right)^{-1} C_2 \tag{3.2.10}$$

The determinant of this matrix is:

$$Det|C_x| = \frac{\left|Det|C_2|\right|^2}{Det\left|C_1^{\psi} + \frac{2}{3}\gamma_1 C_1^T \gamma_1^T\right|} \tag{3.2.11}$$

We can generalize this expression for the case of N transmitter antennas and L transmissions.

$$Det|C_x| = \frac{\left|Det|C_L|\right|^2}{Det\left|C_1^{\psi} + \sum_{i=1}^{L-1}\frac{2}{N}\gamma_i C_1^T \gamma_i^T\right|} \tag{3.2.12}$$

where,

$$C_L = C_1 + \sum_{i=1}^{L-1}\gamma_i^T C_1^* \gamma_i \qquad \text{and} \qquad L \le \binom{N}{2} \tag{3.2.13}$$

Note that this expression is only valid in a time invariant channel, since we assume that the channel remain constant in all the retransmissions.

Note also that, in the normalized case, the selection criterion is based on this determinant (3.2.12). The following two figures 3.5 and 3.6 for 3 and 4 antenna element respectively compare the bit error rates versus E_b / N_0 for the multiple Alamouti HARQ scheme when we use this normalization and when we do not use it. The value of R is the number of times the signal has been transmitted.

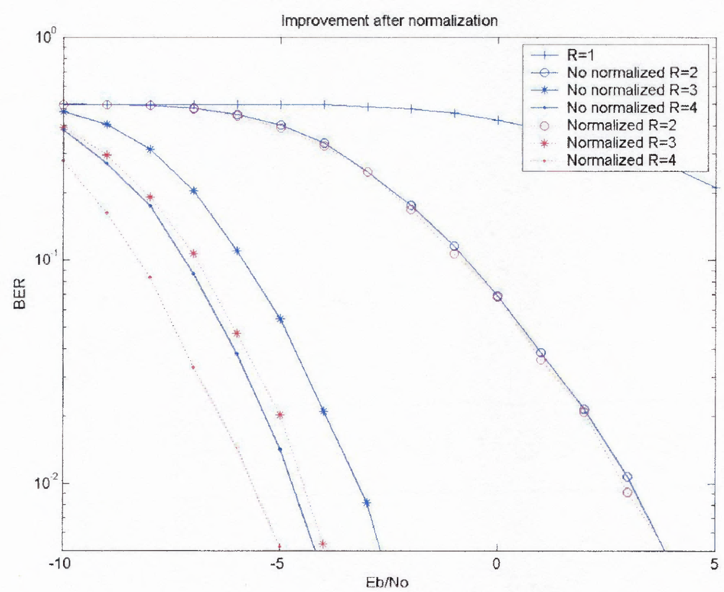

Figure 3.5 Effect of power normalization on the performance of Multiple Alamouti Coding HARQ for the case of 3x3 antennas

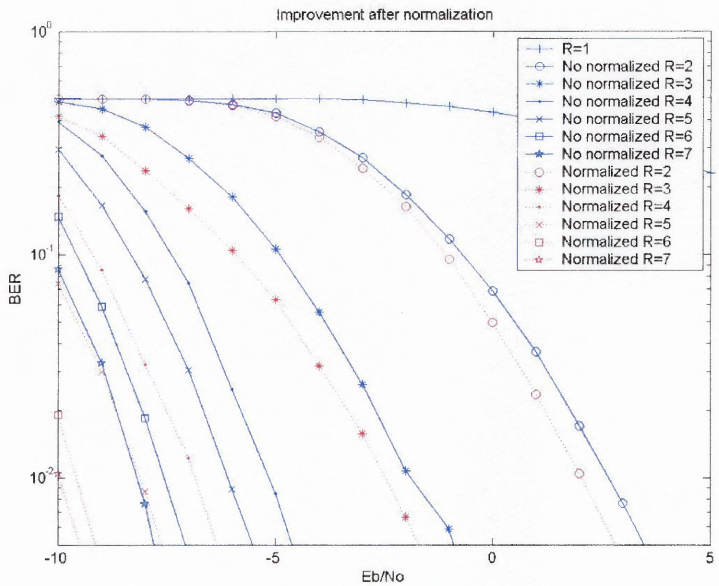

Figure 3.6 Effect of power normalization on the performance of Multiple Alamouti Coding HARQ for the case of 4x4 antennas

Clearly with normalization, the performance is improved, since it is keeping the cancellation properties while at the same time reducing the equivalent noise power.

3.3 Performance Comparison between Multiple Alamouti Coding and LG HARQ Schemes

Once we normalized the transmitted average power of the Multiple Alamouti Coding, we can make a fair comparison of its performance with that of the LG HARQ scheme.

The bit error rates performance comparisons are depicted in figure 3.7 versus E_b / N_0 for R_2, R_3 and R_4 using the proposed determinant selection algorithm for the LG scheme and the modified algorithm of the multiple Alamouti (termed normalized). R_1 presents the first transmission.

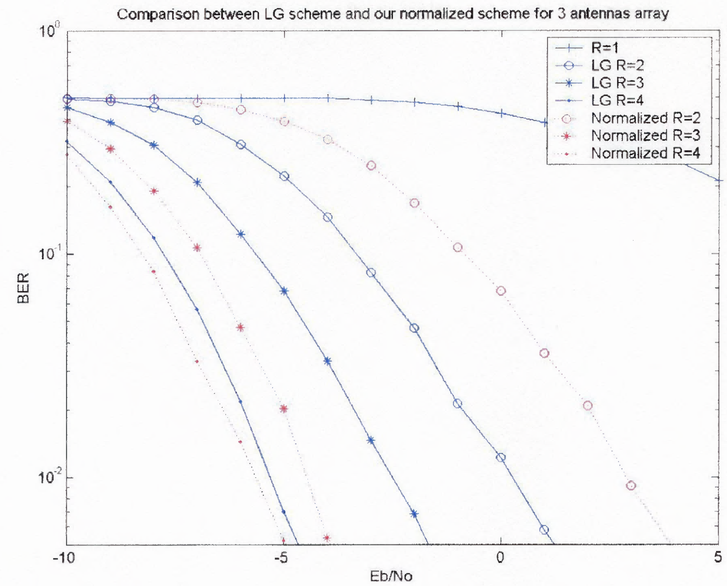

Figure 3.7 BER comparison between LG and normalized Multiple Alamouti Coding HARQ schemes

In figure 3.8 we compare the throughput between the two schemes when maximum four transmissions are allowed.

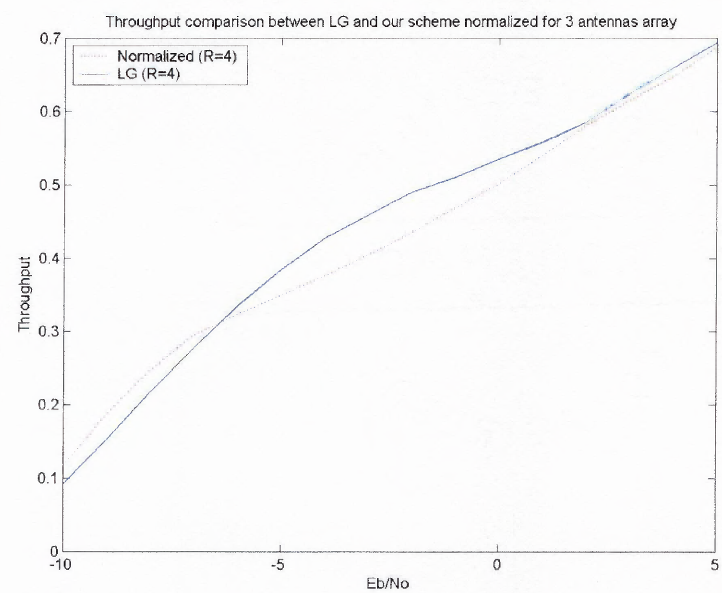

Figure 3.8 Throughput comparisons between LG and normalized Multiple Alamouti Coding HARQ schemes

Note that, for low values of E_b / N_0, the normalized multi-Alamouti scheme is better in terms of throughput, this because as shown in figure 3.7 the BER performance for the R_3 and R_4 transmissions is better than that with LG scheme. However, when the E_b / N_0 is higher, which means we need less transmissions to obtain the correct packet, LG has better throughput. This is because the BER performance for R_2 with LG is almost 3 *dBs* better than the multiple Alamouti.

For the case of 4 element transmitter, the bit error rates performance comparison are depicted in the figure 3.9 versus E_b / N_0 for R_1 to R_7 using the corresponding determinant. R_1 presents the first transmission.

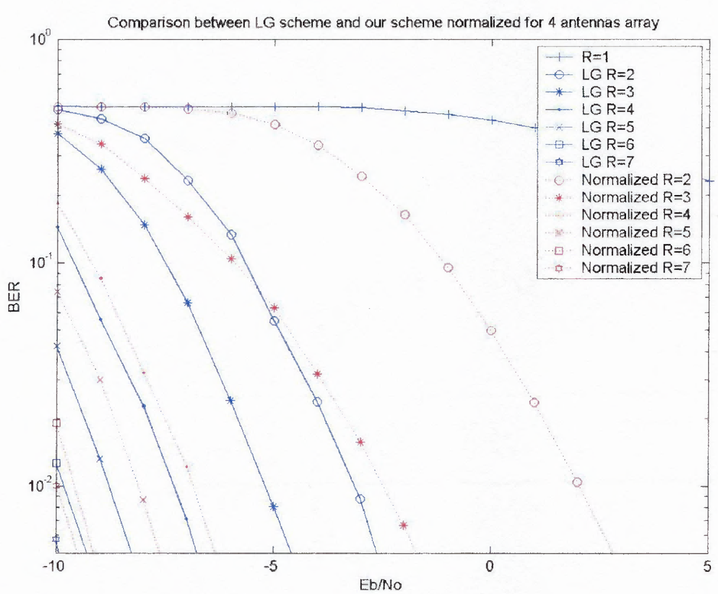

Figure 3.9 BER comparisons between LG and normalized Multiple Alamouti Coding

Again, in figure 3.10 we compare the throughput between the two schemes when seven transmissions are allowed.

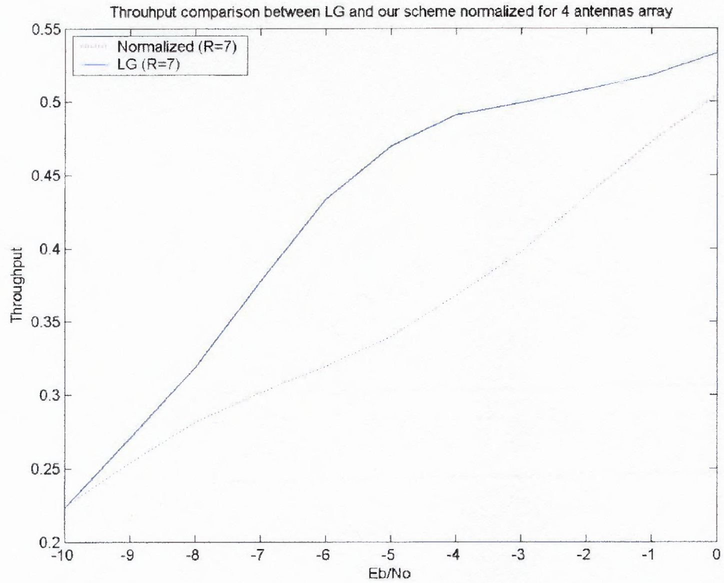

Figure 3.10 Throughput comparisons between LG and normalized Multiple Alamouti Coding

In this case of 4 antennas, the LG scheme is much better in both throughput and BER. This difference is due to the strong effect of the first retransmission; in LG the chances to receive the correct packet after the first retransmission are almost 5 *dBs* higher than the multiple Alamouti scheme, resulted from the fact that LG is using the four antennas to send information instead of only two. This effect is reflected in the resulting matched filtered and combined cross-correlation matrix, where 4 off-diagonal terms are zeroed and all the values of the diagonal are increased, rather than two like in the multiple Alamouti scheme. Although the difference between the performance of multiple Alamouti and LG is becoming closer after each retransmission, the fact that we get an orthogonal channel matrix at R_7, it is not enough to have a better performance than the LG scheme.

Note that we used our determinant selection criterion in the LG's scheme, so if LG uses another criterion the results could be quite different.

CHAPTER 4

HARQ SCHEMES FOR MIMO SYSTEMS IN A TIME VARYING CHANNEL CONDITIONS

In the previous sections we assumed that the channel response remained constant during all retransmissions. Clearly, this assumption is too optimistic, since in a real environment, the channel changes. Therefore this chapter deals with the case when the channel response remains constant only during the length of one packet. If another transmission is required we assume a different channel response which is correlated with the previous one. Such assumption will require different retransmission ordering algorithm than the one we used for the invariant channel. In the first section we will present the time varying channel model that we use. In the next section we will show a solution which cancels the effects produced by a variant channel. Realizing is does not work well, we proposed a modified algorithm which is present and examine in the last section.

4.1 Time Varying Channel Model

In the previous chapters, the channel was assumed to remain invariant for a maximum of L transmissions. Thus the channel model for all retransmissions is a matrix of MxN elements, where N is the number of transmitter antennas and M the number of receiver antennas. The MxN channel gains are then assumed i.i.d. uncorrelated complex Gaussian random variables (Raleigh Flat Fading) with unit power.

$$H_{MxN} = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ h_{M1} & \cdots & \cdots & h_{MN} \end{pmatrix} \qquad (4.1.1)$$

where,

$\mathrm{E}\{|h_{ji}|^2\}=1$, $\mathrm{E}\{h_{ji}h^*_{ml}\}=0$ and $i,l=1\ldots N$ $j,m=1\ldots M$ with $i\neq l$ or $j\neq m$

where $\mathrm{E}\{\cdot\}$ is the expectation.

With the new variant channel response model, it is assumed that the channel matrix remains constant during a packet transmission; however, it will be different at the following retransmission. However to model a different channel matrix at each transmission we use an Autoregressive model of order 1 (AR-1) for each channel gain, which leads to correlation between the current channel and the previous one. To create these channels, we perform the following steps:

- We generate a random matrix H_1 with M by N i.i.d. complex Gaussian Random Variables with unit power.
- The AR-1 model has the following discrete low pass expression

$$h_{ji}^{k+1}=-a_1h_{ji}^k+w_{ji} \tag{4.1.2}$$

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 we need to solve the Yule-Walker equations [15]

$$\begin{bmatrix} R_h(0) & R_h^*(1) \\ R_h(1) & R_h(0) \end{bmatrix}\begin{bmatrix} 1 \\ a_1 \end{bmatrix}=\begin{bmatrix} \sigma_W^2 \\ 0 \end{bmatrix} \tag{4.1.3}$$

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)

$$R_h(0) = 1 \qquad \text{and} \qquad R_h(1) = -a_1 \tag{4.1.4}$$

- Note that the first value keep the power normalization to 1, and the second define 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.
- Note also that we do not need to normalize the power of the new channel coefficients, as can easily be shown from (4.1.3) that

$$\mathrm{E}\left\{\left|h_{ij}^{k+1}\right|^2\right\} = \left|a_1\right|^2 + \sigma_w^2 = 1 \tag{4.1.5}$$

4.2 Modified Channel Combining Algorithm

In the previous chapters we showed that with the Multiple Alamouti combining HARQ scheme, the resulting combined channel correlation matrix after matched filtering becomes diagonal following $\frac{N(N-1)}{2}$ retransmissions (where N is the number of transmit antennas). With, the proposed determinant criterion, we exploited that property; zeroing after each retransmission the most effective off diagonal terms. However, in a time varying channel we lose these cancellations properties of the off-diagonal terms, because at consecutive retransmission we will not have a common matched filtered matrix like we used to have when the channel was invariant. Therefore for time variant case, we proposed to modify each received packet from a retransmission so that to have the same matched filtered matrix as the first transmission. In such a way, when we combine all the retransmitted packets, we will be able to reach a diagonal matrix as in the invariant case. To present such modification the resulting combining after each retransmission we will use in the following an example of $3x3$ MIMO system.

4.2.1 Example of Channel Combining for 3x3 MIMO System

- *First Transmission* (R_1)

  - First channel matrix: H_1

$$r_1^{(1)} = H_1 s^{(1)} + n^{(1)} \quad \text{(First received packet)} \tag{4.2.1}$$

$$x_1^{(1)} = H_1^{\psi} H_1 s^{(1)} + H_1^{\psi} n^{(1)} = C_1 s^{(1)} + H^{\psi} n^{(1)} \quad \text{(After the match filter)} \tag{4.2.2}$$

where $C_1 = H_1^{\psi} H_1$

$$\hat{s}_1 = x_1^{(1)} = C_1 s^{(1)} + H^{\psi} n^{(1)} \quad \text{(Initial packet)} \tag{4.2.3}$$

- *Second Transmission* (R_2)

  - Second channel matrix: $H_2 = H_1 + \Delta H_1$

$$r_1^{(2)} = H_2 s^{(2)} + n^{(2)} \quad \text{(Second received packet)} \tag{4.2.4}$$

$$r_1^{(2)*} = H_2^* \gamma_1 s^{(1)} + n^{(2)*} \tag{4.2.5}$$

where γ_1 is the matrix that relates second transmission packet to the first.

  - We can easily calculate $\Delta H_1 = H_2 - H_1$

  - Before the match filter we modify $r_1^{(2)*}$

$$r_2^{(2)*} = (\mathrm{I} - (\Delta H_1 H_2^{-1})^*) r_1^{(2)*} \tag{4.2.6a}$$

$$= (H_2^* - \Delta H_1^*) \gamma_1 s^{(1)} + (\mathrm{I} - (\Delta H_1 H_2^{-1})^*) n^{(2)*} \quad \text{(after using (4.2.5))} \tag{4.2.6b}$$

$$= H_1^* \gamma_1 s^{(1)} + (\mathrm{I} - (\Delta H_1 H_2^{-1})^*) n^{(2)*} \tag{4.2.6c}$$

where I is the $3x3$ identity matrix.

- After matched filtering to the modified packet at the receiver,

$$x_2^{(2)} = \gamma_1^T H_1^T r_2^{(2)*} = \gamma_1^T C_1^* \gamma_1 s^{(1)} + \gamma_1^T H_1^T (I - (\Delta H_1 H_2^{-1})^*) n^{(2)*} \tag{4.2.7}$$

- Now, we can combine these packets

$$\hat{s}_{1,2} = x_1^{(1)} + x_2^{(2)} \tag{4.2.8a}$$

$$= (C_1 + \gamma_1^T C_1^* \gamma_1) s^{(1)} + H_1^{\psi} n^{(1)} + \gamma_1^T H_1^T (I - (\Delta H_1 H_2^{-1})^*) n^{(2)*} \tag{4.2.8b}$$

- *Third Transmission* (R_3)

- Third channel matrix: $H_3 = H_2 + \Delta H_2 = H_1 + \Delta H_1 + \Delta H_2$

$$r_1^{(3)} = H_3 s^{(3)} + n^{(3)} \quad \text{(Third received packet)} \tag{4.2.9}$$

$$r_1^{(3)*} = H_3^* \gamma_2 s^{(1)} + n^{(3)*} \tag{4.2.10}$$

where γ_2 is the matrix that relates third transmission packet to the first.

- We calculate $\Delta H_2 = H_3 - H_2$

- Before the match filtering we modify $r_1^{(3)*}$

$$r_2^{(3)*} = (I - ((\Delta H_1 + \Delta H_2) H_3^{-1})^*) r_1^{(3)*} \tag{4.2.11a}$$

$$= H_1^* \gamma_2 s^{(1)} + (I - ((\Delta H_1 + \Delta H_2) H_3^{-1})^*) n^{(3)*} \tag{4.2.11b}$$

- After matched filtering to the modified packet at the receiver,

$$x_3^{(3)} = \gamma_2^T H_1^T r_2^{(3)*} \tag{4.2.12a}$$

$$= \gamma_2^T C_1^* \gamma_2 s^{(1)} + \gamma_2^T H_1^T (I - ((\Delta H_1 + \Delta H_2) H_3^{-1})^*) n^{(3)*} \tag{4.2.12b}$$

- Now, we can combine this packet with the previous ones

$$\hat{s}_{1,2,3} = x_1^{(1)} + x_2^{(2)} + x_3^{(3)} = (C_1 + \gamma_1^T C_1^* \gamma_1 + \gamma_2^T C_1^* \gamma_2) s^{(1)} + noise \tag{4.2.13}$$

where,

$$noise = H_1^{\psi} n^{(1)} + \gamma_1^T H_1^T (I - (\Delta H_1 H_2^{-1})^*) n^{(2)*}$$

$$+ \gamma_2^T H_1^T (I - ((\Delta H_1 + \Delta H_2) H_3^{-1})^*) n^{(3)*} \qquad (4.2.14)$$

- *Fourth Transmission* (R_4), which is the last case of three elements array

- Fourth channel matrix: $H_4 = H_3 + \Delta H_3 = H_1 + \Delta H_1 + \Delta H_2 + \Delta H_3$

$$r_1^{(4)} = H_4 s^{(4)} + n^{(4)} \quad \text{(Fourth received packet)} \qquad (4.2.15)$$

$$r_1^{(4)*} = H_4^* \gamma_3 s^{(1)} + n^{(4)*} \qquad (4.2.16)$$

where γ_3 is the matrix that relates fourth transmission packet to the first.

- We calculate $\Delta H_3 = H_4 - H_3$

- Before the match filtering we modify $r_1^{(4)*}$

$$r_2^{(4)*} = (I - ((\Delta H_1 + \Delta H_2 + \Delta H_3) H_4^{-1})^*) r_1^{(4)*} \qquad (4.2.17a)$$

$$= H_1^* \gamma_3 s^{(1)} + (I - ((\Delta H_1 + \Delta H_2 + \Delta H_3) H_4^{-1})^*) n^{(4)*} \qquad (4.2.17b)$$

- After matched filtering to the modified packet at the receiver,

$$x_4^{(4)} = \gamma_3^T H_1^T r_2^{(4)*} \qquad (4.2.18a)$$

$$= \gamma_3^T C_1^* \gamma_3 s^{(1)} + \gamma_3^T H_1^T (I - ((\Delta H_1 + \Delta H_2 + \Delta H_3) H_4^{-1})^*) n^{(4)*} \qquad (4.2.18b)$$

- Now, we can combine with the previous packets

$$\hat{s}_{1,2,3,4} = x_1^{(1)} + x_2^{(2)} + x_3^{(3)} + x_4^{(4)} \qquad (4.2.19a)$$

$$= (C_1 + \gamma_1^T C_1^* \gamma_1 + \gamma_2^T C_1^* \gamma_2 + \gamma_3^T C_1^* \gamma_3) s^{(1)} + noise \qquad (4.2.19b)$$

where,

$$noise = H_1^{\psi} n^{(1)} + \gamma_1^T H_1^T (I - (\Delta H_1 H_2^{-1})^*) n^{(2)*} + \gamma_2^T H_1^T (I - ((\Delta H_1 + \Delta H_2) H_3^{-1})^*) n^{(3)*}$$

$$+ \gamma_3^T H_1^T (I - ((\Delta H_1 + \Delta H_2 + \Delta H_3) H_4^{-1})^*) n^{(4)*} \qquad (4.2.20)$$

Note that,

$$(C_1 + \gamma_1^T C_1^* \gamma_1 + \gamma_2^T C_1^* \gamma_2 + \gamma_3^T C_1^* \gamma_3) = \begin{pmatrix} \chi & 0 & 0 \\ 0 & \chi & 0 \\ 0 & 0 & \chi \end{pmatrix} \tag{4.2.21}$$

If the channel is constant in all the repetitions ($\Delta H_1 = \Delta H_2 = \Delta H_3 = 0$), then

$$\hat{s}_{1,2,3,4} = x_1^{(1)} + x_2^{(2)} + x_3^{(3)} + x_4^{(4)} \tag{4.2.22a}$$

$$= (C_1 + \gamma_1^T C_1^* \gamma_1 + \gamma_2^T C_1^* \gamma_2 + \gamma_3^T C_1^* \gamma_3) s^{(1)} + noise \tag{4.2.22b}$$

where,

$$noise = H_1^{\psi} n^{(1)} + \gamma_1^T H_1^T n^{(2)*} + \gamma_2^T H_1^T n^{(3)*} + \gamma_3^T H_1^T n^{(4)*} \tag{4.2.23}$$

This is the same equation that we get in (2.3.11)

4.2.2 Numerical Results

In the following figures, the performance of the proposed combining algorithm (probability of error versus E_b / N_0), called “Modified Channel Combining”, is compared to the case when “non modified channel combining” is implemented on a time varying channel. That is in the later although the channel is changing at each retransmission, the receiver combines the packets as if it was an invariant channel, i.e. used the channel matrix of the first retransmission. The other case in the figure called “Invariant channel” is depicted as a reference, where in the channel remains constant in all the repetitions.

The time varying channel is modelled with an AR-1 with the following parameters:

$$a_1 = -0.5 \qquad \text{and} \qquad \sigma_w^2 = 0.75 \tag{4.2.24}$$

For the three cases we depict from the performance after second retransmission (R_2) to fourth retransmission (R_4). Note that for R_1 all the cases are the same. In these results is not used any criterion of selection, since for each case should be a different criterion, and the aim of this graphic is to see if this algorithm work out.

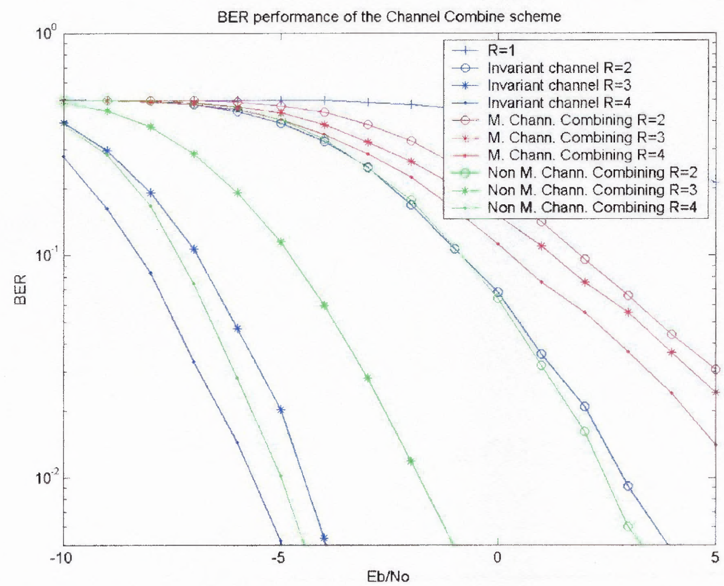

Figure 4.1 BER performance of the proposed Channel Combining scheme in a time variant channel

From figure 4.1 we can notice that if we just combine the received retransmission packet without modification, hence not taking aiming to finally reach a diagonal matrix, the performance is much better than to use the channel modification proposed above. This is due to the increase in level of noise every time we do modification when multiplying each packet by, for example, $(I-((\Delta H_1+\Delta H_2)H_3^{-1})^*)$. It seems that this factor has more effect than the fact that the channel is varying at each retransmission. Another conclusion that we can draw from these result is that the difference between the variant and invariant case is not so high (at least for R_2 and R_4), hence, we thought that if we adapt the retransmission order algorithm for the case of the variant channel, we can get better results, since from figure 4.1 we noticed that the fact of reaching a diagonal matrix is not a priority.

4.3 Modified Retransmission Order Algorithm for a Time Varying Channel (without Channel Modification)

As was described in previous section, channel modification approach which was motivated by the goal of obtaining after the final retransmission (the fourth for 3x3 MIMO) a diagonal channel matrix did not perform well enough. Therefore we intend in this section not to modify the channel matrix. Recalling in section 2.4 since the channel was assumed the same in all the retransmissions it was possible to decide on the best order of retransmission once the channel matrix was known after the first transmission. This is because the alternative selection algorithm always used the channel matrixes of previous transmission and the one for the next retransmission, which all are the same for the invariant channel. For the new scenario, where the channel matrix changes at each retransmission, instead of using the determinant criterion at the beginning of the transmission, we use an algorithm which checks every time that a retransmission is needed. However when we use the algorithm to decide on the next retransmission alternative, we know the previous channel matrixes, but we do not know the channel matrix of the next transmission, which we have to estimate in some way.

4.3.1 The Determinant Criterion in a Time Varying Channel

The following example describes how to use the algorithm for the case of $4xM$ antennas (extension to different number of element antennas is straightforward). As described in chapter 3 we apply the determinant criterion with power normalization.

After the first transmission (R_1), if a retransmission (R_2) is required, we check the following determinants to decide on one of the alternatives

$$Det\left|C_{x_1}\right| = \frac{\left|Det\left|C_1 + \gamma_i^T \widetilde{C}_2^{\,*} \gamma_i\right|\right|^2}{Det\left|C_1^{\psi} + \frac{1}{2}\gamma_i \widetilde{C}_2^{\,T} \gamma_i^T\right|} \qquad \text{and } i = 1 \ldots 6 \qquad (4.3.1)$$

where $C_1 = H_1^{\psi} H_1$, γ_i is one of the matrix transformations from (2.4.3) and

$$\widetilde{C}_2 = E\{H_2^{\psi} H_2\} = \mathrm{E}\{(a_1 H_1 + W)^{\psi} (a_1 H_1 + W)\} \qquad (4.3.2a)$$

$$= a_1^2 H_1^{\psi} H_1 + E\{W^{\psi} W\} = a_1^2 C_1 + 4\sigma_w^2 \mathrm{I} \qquad (4.3.2b)$$

where I is a $4x4$ identity matrix and W is the noise matrix of the AR-1 model with $Mx4$ elements. Note that we estimate the next channel using the expected value of the AR-1 model that we defined earlier. If a third transmission is required (R_3), the new determinants that we have to check to decide among one of the five alternatives left are given by,

$$Det\left|C_{x_2}\right| = \frac{\left|Det\left|C_1 + \gamma_i^T C_2^* \gamma_i + \gamma_j^T \widetilde{C}_3^* \gamma_j\right|\right|^2}{Det\left|C_1^{\psi} + \frac{1}{2}\gamma_i C_2^T \gamma_i^T + \frac{1}{2}\gamma_j \widetilde{C}_3^T \gamma_j^T\right|} \qquad (4.3.3)$$

where i is the first vector selected and j is the second one. Note that j belongs a set of numbers which i is not included. In (4.3.3) C_2 is the actual channel matrix (H_2) through which the second transmission is received. Similarly for the third transmission we do not know H_3 yet, hence we estimate again its value

$$\widetilde{C}_3 = E\{H_3^{\psi} H_3\} = a_1^2 C_2 + 4\sigma_w^2 I \tag{4.3.4}$$

In case that more transmission were required, we would follow the algorithm like we did in previous sections, but at each step we would have to estimate the value of the future channel, as in (4.3.4). Note that in case that a seventh transmission (R_7) is required, we use the last alternative left.

4.3.2 Numerical Results

The next figures depict the performance of the multiple Alamouti scheme with the modified determinant algorithm for the case of a time varying channel in a $4x4$ MIMO system. The first one (Fig. 4.2) shows the BER versus E_b / N_0 when the channel that at each retransmission is quite correlated with the previous one, with $a_1 = 0.9$. In figure 4.3 we use a channel more varying in time ($a_1 = 0.5$). In both figures, the new selection algorithm is termed "Time Varying (T.V.) with proposed". For comparison also added in the figure what is termed "T.V. without proposed" i.e. the selection of $s^{(i)}$, $i = 2,...,7$ to send R_2 up to R_7 are done at the beginning, using the values of the first channel (H_1), despite change in the channel. Also for comparison, in both figures is added the performance of the algorithm in an invariant channel.

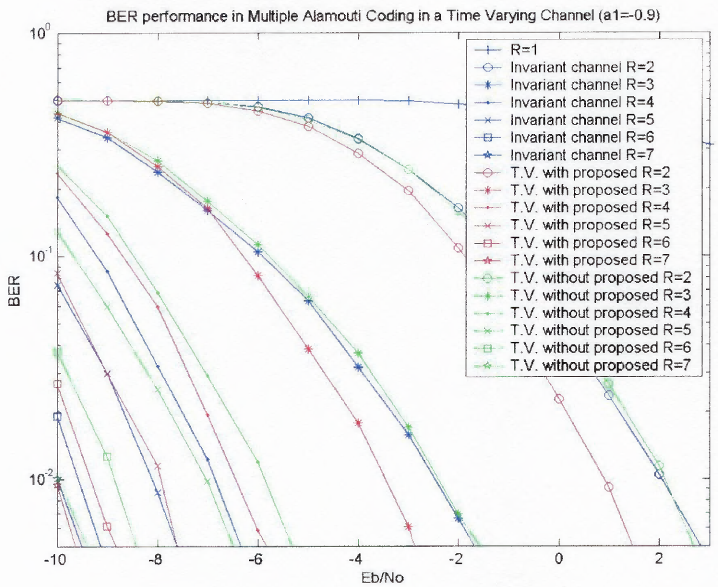

Figure 4.2 BER performance of the proposed modified retransmission order algorithm in a Time Varying Channel with $a_1 = 0.9$

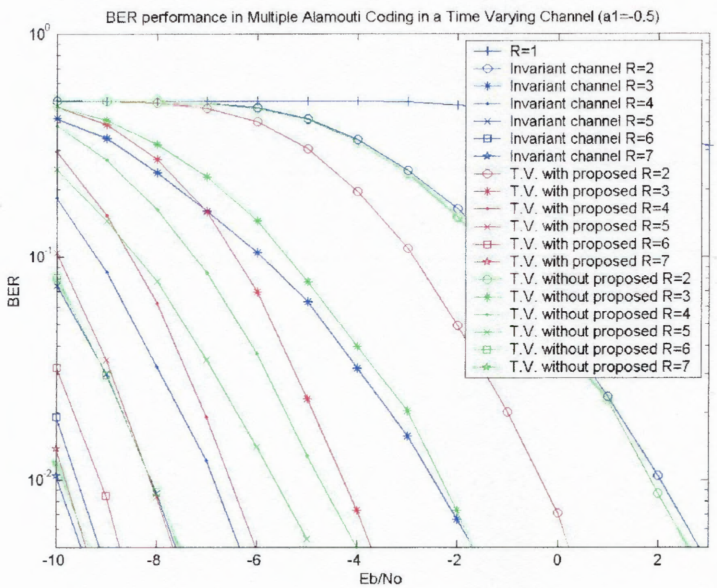

Figure 4.3 BER performance of the proposed modified retransmission order algorithm in a Time Varying Channel with $a_1 = 0.5$

The next two figures depict the throughput of the two cases described above. In brackets, we showed the maximum number of allowed transmissions.

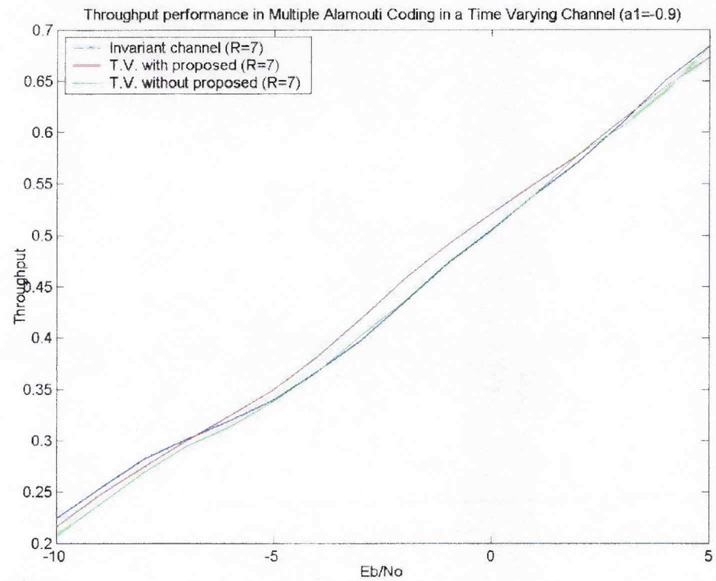

Figure 4.4 Throughput performance of the proposed modified retransmission order algorithm in a Time Varying Channel with $a_1 = 0.9$

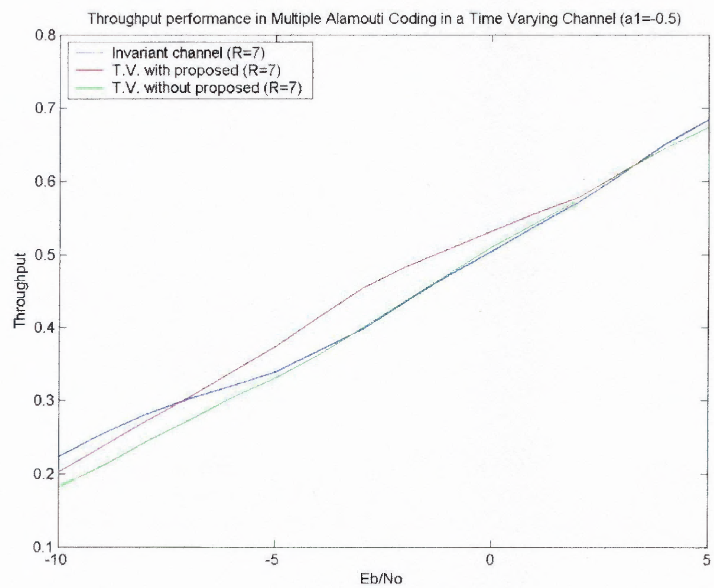

Figure 4.5 Throughput performance of the proposed modified retransmission order algorithm in a Time Varying Channel with $a_1 = 0.5$

The results show that, in a varying channel using the proposed modified retransmission order algorithm, the BER performance is better than an invariant channel in the first stages (for R_2 and R_3). This performance trend is also noticed in the throughput, where from -6 to 2 dBs of E_b / N_0 the variant channel outperform the invariant channel, since the chances to receiver the correct packet at the first's stages are 2 dBs higher in the variant channel than in the invariant (for the case of a time varying channel with $a_1 = 0.5$). This effect is due to the fact that in a varying channel the diversity of the channel change at each step from which the proposed algorithm we can benefit. For a high number of retransmissions, where in the channel matrix approaches diagonal, the invariant channel outperforms the variant channel. Note also that the proposed algorithm always has better performance than the case where in the retransmission are decided at the first stage using the value of the first channel (H_1) (although the channel is variant in time).

CHAPTER 5

CONCLUSION

HARQ is an important protocol used in packet transmission to provide reliable data communications. MIMO systems are also well known to increase the spectral efficiency and the capacity of a communication system. In chapter 2 scheme termed Multiple Alamouti Coding (MAC) was suggested for HARQ. At each retransmission data is sent on 2 antennas out of N to form with the corresponding previously sent data, an Alamouti code, hence eliminating the effect of the corresponding cross term of the channel matrix. In this thesis a Retransmission Order Algorithm supporting HARQ for MIMO is proposed for a NxM MIMO channel in two different scenarios: time invariant channel and time varying channel. This selection algorithm exploits the proprieties of the channel matrix to choose the better sequence for retransmission with the MAC. Simulation shows that in a slow varying channel with the proposed selection algorithm better performance is obtained than with the random selection for retransmission of the Multiple Alamouti sequences. For example, from Figure 2.11 (the case of 4 element transmitter) we noticed an improvement of up to 4 dBs. In a time varying channel, we modified the algorithm so that the better alternative to retransmit is decided based on estimation of the next channel. Results show that at the first stages of retransmission (R_2 for example), the performance in a varying channel is better than an invariant channel. This is due to the fact that in a varying channel the diversity of the channel changes is exploited by the modified algorithm. For a high number of retransmissions, the system with invariant channel outperforms the variant channel, since with the former an almost diagonal matrix is approached, while not so with the later.

In this thesis, we also compared the performance MAC scheme with another scheme, proposed by LG Electronic, noticing the former outperforms MAC, even though it does not finally approach a diagonalized channel matrix as with the latter. This indicates that such diagonalization is not crucial. In section 3.1 it is shown, for the case of 4 element transmitter, that after the first retransmission, the resulting matched filtered and combining the cross-correlation matrixes four off-diagonal terms are zeroed and all the values of the diagonal are increased. This result can explain the huge difference in Figure 3.9 in favour of LG scheme for R_2 in a time invariant channel. Nevertheless, for a high number of retransmissions this difference diminishes as the Multiple Alamouti scheme a diagonal matrix is reached with MAC.

The retransmission order algorithm proposed in this thesis was proposed for the multiple Alamouti scheme, however, we showed by simulation that this algorithm is also suitable for other schemes, such as LG. The key feature of the algorithm is that it depends only on the resulting matched filtered and combined cross-correlation matrix, hence, it does not matter what code is used to send the data.

In future we suggest processing in the frequency domain instead of the time domain to hand fast fading dispersive channel. Also, it would be interesting use different models for the time varying channel, and find out if the proposed selection algorithm would have better performance than an invariant channel. Finally, we suggest using the selection algorithm in other schemes, such as Basis Hopping and compare its performance with the other schemes discussed in this work.

APPENDIX

MATLAB SOURCE CODES

Main Program

```
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=3;
packet_size=522; %with this size we can split the packet in 2,3,4,5 and 6 parts
N=3;
M=3;
random=0; %1 we use random selection of the vectors, 0 we use the proposed algorithm
seed=45; %seed for the initiall state in the function rand
SNR=[-10:1:-5]; %SNR in dB
L=length(SNR);
Result=ones(2,L);%we save the BER and the throuhput for each value of SNR
if(random)
    filename = ['HARQ_random',num2str(N),'x',num2str(M),'_R',num2str(R)];
    for(i=1:L)
        Result(:,i)=HARQ(n,R,SNR(i),packet_size,floor(seed*rand),N,M,random)';
    end;
else
    filename = ['HARQ_proposed',num2str(N),'x',num2str(M),'_R',num2str(R)];
    for(i=1:L)
        Result(:,i)=HARQ(n,R,SNR(i),packet_size,floor(seed*rand),N,M,random)';
    end;
end;
save (filename,'Result','n');
return;
```

HARQ function

```
function [Result]=HARQ(n,R,SNR,packet_size,state,N,M,random);

%this function return the probability of error and the throughput
%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
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
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
    if (random)
        A=I; %we don't use the algorithm
    else
        A=decisorNxM(I,H,N); %this function returns the matrix I with the best order for
%transmission
    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
        x=send(V,H,noise,A(N*y+1:N*y+N,:),y); %returns the data after having been
processed
        Vest=Vest+x; %we combine all the vectors
        Vzf=ZF(Vest,H,A,r,y,N,P); %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;
        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;
```

Alamouti Generator function

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

columns=N;
rows=1+factorial(N)/(factorial(N-2)*2);
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-1
    for j=i+1:N
```

```
        aux(i,j)=(-1)^(power);
        aux(j,i)=(-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;
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;
```

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;
```

Encoder Function

```
function [X]=encoder(I);
%This function do a convolutional Trellis Code (2,1,3)

%we have to add 2 zeros in the packet I
L=length(I);
I(L+1)=0;
I(L+2)=0;
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;
```

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

```
function [A]=decisorNxM(I,H,N);
%I is the Alamouti random order matrix
%H is the Channel Matrix
%N is the number of transmitting antennas

A=0*I;
A(1:N,:)=I(1:N,:); %the first vector always will be the same
C0=H'*H;
C0x=conj(C0);
I(1:N,:)=[];
loops=factorial(N)/(factorial(N-2)*2);
for i=1:loops-1
    y=determinant(I,C0x,C0,loops+1-i,N); %y gives us the first vector
    A(N*i+1:N*i+N,:)=I(N*y+1:N*y+N,:);
    I(N*y+1:N*y+N,:)=[]; %we can delete this vector, because we don't need it more
```

```
    C0=C0+A(N*i+1:N*i+N,:)'*C0x*A(N*i+1:N*i+N,:);
end;
%finally...the last vector
A(N*loops+1:N*loops+N,:)=I;
return;
```

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] Mietzner, J.; Hoeher, P.A., "Boosting the performance of wireless communication systems: theory and practice of multiple-antenna techniques" *IEEE Communications Magazine*, Volume: 42, Issue: 10, Oct. 2004, pp. 40 – 47.

[2] G. J. Foschini, "Layered Space-time Architecture for Wireless Communication in a Fading Environment When using Multi-Element Antennas," *Bell Labs Tech. J.*, Autumn 1996, pp. 41–59.

[3] John G. Proakis and Masoud Salehi, "Communications Systems Engineering", pg. 702.

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

[5] Yoon, S. and Bar-Ness Y., " Packet Data Communications over coded CDMA, part II: Throughtput-bound of CDMA in Slotted Aloha with Hybrid Type II ARQ using Rate Compatible Punctured Turbo Coded" IEEE Trans. On Wireless Comm., Vol 3 , pp 1616 1625 Sept 2004.

[6] D. Chase, "Code combining- a maximum likelihood decoding approach for combining an arbitrary number of noise packets," IEEE Transactions on Communications, vol. COM-33, pp. 385-393, May 1985.

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

[8] V. Tarokh, Ming Jia, Weng Jong, Pelying Zhu, Jlanglel Ma, "Incremental Redundancy with Space-Time Codes", Patent Application Publication, Nortel Networks Limited, Num US2004/0057530A1, Mar. 2004.

[9] 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.

[10] M. Aguilar Igartua, "Codificación de Canal", Universidad Politécnica de Catalunya, class notes, pg. 32-33.

[11] John G. Proakis and Masoud Salehi, "Communications Systems Engineering", pg. 778-779.

[12] Y. Bar-Ness and K. Acolaste, "A New Hybrid-ARQ Scheme for MIMO Systems", January 2005.

[13] John G. Proakis and Masoud Salehi, "Communications Systems Engineering", pg. 784-788.

[14] Enhanced STC subpacket combining in OFDMA" LG Electronic, Inc. 3/16/05.

[15] John G. Proakis and Masoud Salehi, "Communications Systems Engineering", pg. 265-267.