INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS ESTADO DE MÉXICO

DESARROLLO DE UN SISTEMA DE CODIFICACION ESPACIO TIEMPO ORIENTADO A LAS REDES INALAMABRICAS DE TIPO WLAN.

TESIS QUE PRESENTA

ING. KARINA Y. SOSA GONZALEZ.

MAESTRÍA EN CIENCIAS DE LA INGENIERIA
ESPECIALIDAD COMUNICACIONES
MCI-C 04

JULIO, 2005

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS ESTADO DE MÉXICO

DESARROLLO DE UN SISTEMA DE CODIFICACION ESPACIO-TIEMPO ORIENTADO A LAS REDES INALAMBRICAS DE TIPO WLAN.

TESIS QUE PARA OPTAR EL GRADO DE
MAESTRO EN CIENCIAS DE LA INGENIERIA
PRESENTA

ING. KARINA Y. SOSA GONZALEZ.

Asesor: Dr. LUIS FERNANDO GONZALEZ PEREZ.

Comité de tesis:
Dr. Andrés David García García.
Dr. George Rodriguez Guisantes.
Dr. Luis Miguel Bazdresch Sierra.
Dr. Javier González Villaruel.
Dr. Luis Fernando González Pérez.

Jurado:		
	Dr. Andrés David García García.	Presidente
	Dr. Javier González Villaruel.	Secretario
	Dr. George Rodriguez Guisantes.	Vocal
	Dr. Luis Miguel Bazdresch Sierra.	Vocal
	Dr. Luis Fernando González Pérez.	Vocal

Atizapán de Zaragoza, Estado de México. Julio de 2005.

INDEX

AGRADECIMIENTOS

Primero que nada le agradezco al **ITESM CEM**, quien representado por todas las personas que aquí laboran, ha hecho realidad uno de los muchos sueños que conforman la gran montaña que quiero escalar en mi vida, la Maestría, sinceramente, gracias.

Al Dr. Luis Gonzalez Pérez, mi estupendo asesor, quien no solamente ha sido un gran compañero en los días buenos y malos, sino también la fuente de todo el conocimiento recibido. Es y ha sido un gran amigo durante la elaboración de este trabajo, y aunque no siempre hemos tenido los mismos puntos de vista, nunca perdió del todo la fe en mí. Te lo agradezco infinitamente.

A mi segundo asesor, Dr. Miguel Bazdresch, quien desde Francia me daba muchos buenos consejos de cómo hacer o investigar todo lo relacionado con este trabajo, gracias querido Dr. Mike, por las cosas buenas, por los regaños y por aquellas veces que me hiciste desatinar por medio del Chat con preguntas, cuando no sabia la respuesta, pero me hiciste pensar todo el tiempo, y eso me agrado sobremanera.

Zu meinem geehrten Ehemann, **Stich**, der sechs Jahre meines Lebens geteilt hat, gleich dieses ist Wunderbare, mit Ihrer Anstrengung und ununterbrochener Ermutigung, Du hast möglich gemacht, daß mein Leben in einem Berg erreichter Träume geworden ist, und von neuen Träumen, um zu kommen, daß wir zusammen erreichen werden. Diese These ist, teilen Sie sich von Ihnen auch, es ist einfach für Dich, ich liebe Dich, und Du sind wesentlicher Teil von meinem Leben, dem besten Teil. Danke für diesen ganzen Impuls, Unterstützung, Verständnis, Zuneigung, Sorgen und Liebe, die mich während dieses Jahres und der Hälfte bewegten. Danke, um zu sein damit meins! Diese Arbeit ist Ihres!! Es ist unser!! Wir bekommen es!

A mi madre, la mujer mas maravillosa creada en la tierra, de quien no solamente debo mi vida como tal, sino también lo bueno o malo que haya yo hecho sobre esta tierra, de ti he aprendido casi todo, desde hablar, caminar, hasta ser lo suficientemente perseverante para crear sueños, construir los caminos para llegar a ellos y tener la fuerza suficiente para aguantar los trayectos a pesar de las dificultades y de las personas que nos encontramos en el camino. Este trabajo es fruto tuyo también, porque has soportado mi mal genio y los días buenos con una sonrisa franca en tus labios y muchas palabras de aliento.

A ti mi querida hermana Yoya, porque todo el tiempo creíste en mis capacidades aun cuando yo misma lo dudaba en ocasiones. Tus palabras, tus consejos y en ocasiones tus jalones de oreja me hicieron darme cuenta que la maestría es mucho mas allá de un simple trabajo, es un paso que me acerca a ser una verdadera investigadora. Gracias por estar, por nunca dejar que me rindiera, por haber estado conmigo aun en las malas, también gracias

por ser quien eres, una gran hermana del alma. Prometo ayudarte cuando te decidas a hacer tu maestría, cuenta con eso!!

A mi familia en general, ya que sin ellos el sueño de estar en el Tecnológico de Monterrey nunca hubiese sido posible, **a Martita y Rusito (Alex)** por el continuo apoyo que me han dado durante toda la vida y por algunos patrocinios otorgados (porque por eso son mis tíos favoritos caray!), **a ti Daniel**, mi querido hermano, por el apoyo que siempre me diste durante todos los proyectos que nos hemos puesto en la vida, tu has sido mi ídolo desde siempre, y el motor que me ha impulsado a llevar esta vida de deportista que ha reinado desde siempre, gracias a ti y tu esposa, Suheim. **A mi padre**, que aunque eres el demonio mismo, te debo parte de lo que soy y mi bello rostro. A la futura heredera Sosa, **mi querida sobrina Sharon**, porque aunque los fines de semana no me dejabas hacer la tesis, si compartimos bellos momentos, esto es también un esfuerzo para ti, porque eres un grandioso motor que mueve mi mundo. También quiero dedicar esta tesis a mis hermosas hijas (Blacky, Fedora, XP y Rabit) quienes todas las noches de desvelo me acompañaron sin nunca repelar ni criticar mi trabajo. A mis abuelos maternos, quienes ya no están conmigo, pero que sin duda me dieron un apoyo moral desde donde quiera que se encuentren, en especial, a ti querido abuelo Carlos… ves? Lo he logrado!

A mis queridos amigos del alma… Primeramente**, a Reyna** del alma, de quien a partir de la vocacional (viva Batíz!) has sabido ser una guía muy importante en mi camino, y una impulsora de todas las ideas que se me han ocurrido no importando que estés o no de acuerdo. **A ti Araceli**, mi psicóloga favorita, por ayudarme en los momentos difíciles y ser una gran confidente durante toda mi vida. **A Jessi**, quien tus largas charlas hizo que mi cabeza dejara atrás los nubarrones de mi desesperanza cuando las cosas no iban muy bien, llenando de fe mi mundo agobiado, esto es también parte de tu esfuerzo. **A Fausto y su gran Familia**, especialmente su esposa Olga, quienes durante toda la carrera de profesional y esta maestría me han brindado su apoyo, confianza y sobre todo, su amistad sin condiciones, en verdad les agradezco profundamente todo lo que han hecho por mi. A todos aquellos que no son mencionados pero que compartieron conmigo sueños y logros, desengaños y desesperanzas, sin nunca perder su fe, gracias.

Al Dr. Ricardo Swain, quien si su apoyo este trabajo, y el grado en si no hubiesen sido posibles. Gracias por el apoyo y la confianza depositados en mí.

Al Dr. Alejandro Aceves, quien como director de la Maestría logro que los malentendidos fueran un asunto meramente trivial, haciendo que mi estancia fuese lo más cómodamente posible.

Y finalmente, **a todos los profesores que tuve, en especial al Dr. Andrés García García**, porque con su guía, regaños y correcciones, me ayudaron durante la carrera profesional y la maestría, querido Andie, en verdad aprendí mucho de ti y de los demás.

RESUMEN

La capacidad de canal en los canales inalámbricos se ha convertido en uno de los aspectos principales dentro de las comunicaciones inalámbricas. Mejoras significativas pueden lograrse con los canales MIMO (Múltiple Entrada - Múltiple Salida) incrementando el numero de antenas en las dos partes, tanto en el transmisor como en el receptor. La recepción en este tipo de canales puede ser realizada a través de la codificación en espacio-tiempo, donde la unión de los diseños de corrección de errores, modulación, diversidad en la transmisión y diversidad optima en la parte de recepción es realizada. Un importante incremento en el desempeño de los canales inalámbricos de banda limitada con esta técnica es posible. Uno de los primeros y mas importantes sistemas de codificaciones de canal para sistemas MIMO es la técnica de codificación en espacio-tiempo por capas creada por Bell Laboratorios (BLAST). Existen diferentes variantes de este algoritmo para tratar de obtener una reducción en la complejidad material que permita ser llevado a la realidad. Una de estas variantes es el algoritmo Vertical BLAST (VBLAST), mismo que fue escogido durante la realización de esta tesis debido a que presenta el mejor compromiso entre la complejidad y su desarrollo contra errores, adicionalmente es el único que no cuenta con codificación de canal, lo que hace posible la experimentación con él. La parte de procesamiento de la señal en la recepción puede realizarse de diferentes maneras. Uno de los objetivos de la tesis es el análisis de dichas variantes desde el punto de vista complejidad-desarrollo. Tres variantes del algoritmo son analizadas: Descomposición por Valores Singulares (SVD), Descomposición Ordenados QR (SQR) y mínimos cuadrado (LS). Una comparación con el algoritmo ML también es hecha. Se demuestra que el algoritmo LS contiene las mejores características para ser implementado físicamente. Por otro lado, una codificación de canal fue hecha para disminuir la probabilidad de error dentro de sistema, y se analiza también el incremento de complejidad dentro del sistema contra la versatilidad en cuanto a su forma de detectar los errores. Se presenta panorama del trabajo futuro donde se introduce la posibilidad de integrar técnicas de codificación modulada así como la implementación física del sistema en un DSP (Digital Signal Processor - Procesador de señales digitales).

ABSTRACT

Channel capacity in wireless channels has become the prime aspect in mobile communications. Significant improvements can be achieved with MIMO (Multiple Input - Multiple Output) channels by increasing the number of antennas at both the transmitter and the receiver. Reception in this type of channels can be done with space-time coding where a joint design of forward error correction, modulation, transmit diversity and optional receive diversity is made. An important increase in the throughput of band-limited wireless channels with this technique is possible. One of the first and most important space-time coding systems for MIMO channels is the Bell Laboratories Layered Space-Time (BLAST) coding technique. Different variants of this algorithm exist in order to reduce the computational complexity to make it realizable. One of these variants is the Vertical BLAST algorithm, which was chosen because presents the best tradeoff between complexity and performance of all; otherwise, it has not real channel codification. VBLAST has different realizations depending on the way the signal processing at the receiver is performed. One of the aims of the thesis is to analyze these realizations from a performance-complexity standpoint. Three variants of this algorithm are analyzed, the Singular Value Decomposition (SVD), the Sorted QR Decomposition (SQR) and the Least Square (LS) VBLAST. Comparison with the ML (Maximum Likelihood) algorithm is also made. It is shown that the LS algorithm is the best suited for hardware implementation. On the other hand, additional channel codification is made in order to decrease the error probability, and the posterior analysis of the additional complexity. We will present future work which includes the introduction of coded-modulation techniques and the possibility of make a physical implementation in a DSP (Digital Signal Processor) system.

LIST OF SYMBOLS

M	Transmit Antennas
N	Receive Antennas
H	Channel Matrix
F_c	Transmitter Frequency
Q	Angle with respect to the direction of the motion
f_d	Doppler Frequency
c	Light Speed
n	Mobil's Speed
y(t)	Received Signals
x(t)	Transmitter Signals
H(t,t)	Channel Transfer Matrix
V	Processing Operation
U	Receiver Incoming Operation
.'	Means Transpose Operator
I	Unitary Matrix
z(t)	Output Data Stream
l	Positive constant
C	Channel Capacity
$SINR_0$	Initial Signal to Noise plus Interference Ratio
B	Bandwidth
SINR	Signal to Noise Interference Ratio
E| |	Expected Value
s	Variance
PDF	Probability of Density Function
p(z)	Probability of Density Function
SNR	Signal to Noise Ratio
r	Code Rate
k	Number of Transmit Antennas
p	Transmission Intervals
*****	Conjugate Operation
Z	Input Vector
R	Euclidean Space
D	Dimensional Euclidean Space
L	Generator Matrix of a Lattice
.	Hadamard Product
||.||	Euclidean Norm
|a|	Magnitude of a scalar *a*
min	Minimum of various values
argmin	number with minimum squared magnitude
O_b	Number of Operations estimating one bit of information
L	Number of bits in stream

Add	Number of addition operations
Mul	Number of multiply operations
Div	Number of division operations
Sqrt	Number of square root operations
Mem	Number of memory location
S/P	Serial to Parallel
P/S	Parallel to Serial
$\mathbf{w}^{\mathbf{H}}$	Conjugate Transpose of w
$\mathbf{w}^{\mathbf{T}}$	Transpose of A
$\mathbf{A}^{+}$	Moore-Penrose inverse (pseudo-inverse) of A

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

3G	Third Generation of Communications
4G	Fourth Generation of Communications
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLAST	Bell Labs Layered Space Time
BLER	Block Error Rate
BPSK	Binary Phase Shift Keying
CC	Convolutional Channel Coding
CDMA	Code Division Multiple Access
dB	Decibels
DBLAST	Diagonal Bell Labs Layered Space-Time
DLST	Diagonal Layered Space-Time
DSP	Digital Signal Processor
EGC	Equal-Gain Combining
FPGA	Field Programmable Gate Architecture
GPRS	General Packet Radio Service
GSM	Global System for Mobile
HLST	Horizontal Layered Space-Time
IC	Integrated Circuits
IEEE	Institute of Electrical and Electronics Engineers
KZ	Korkine-Zolotareff
LAN	Local Area Network
LLL	Lenstra-Lenstra-Lovász
LOS	Line of Sight
LS	Least Square
LST	Layered Space-Time
MEA	Multiple Element Arrays
MIMO	Multiple Input - Multiple Output
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
MRC	Maximal Ratio Combining
PDF	Probability of Density Function
PLD	Programmable Logic Device
QAM	Quadrature Amplitude Modulation
SI	Spatial Interleaver
SINR	Signal to Noise Interference Ratio
$SINR_0$	Initial Signal to Noise plus Interference Ratio
SISO	Single Input - Single Output
SNR	Signal to Noise Rate
SQR	Sorted QR Decomposition
ST	Space-Time

STC	Space Time Coding
STBC	Space-Time Block Codes
STTC	Space-Time Trellis Codes
SVD	Singular Value Decomposition
TLST	Threaded Layered Space-Time
VBLAST	Vertical Bell Labs Layered Space Time
VLST	Vertical Layered Space-Time
WLAN	Wireless Local Area Networks Standard
ZF	Zero Forcing

1.

INTRODUCTION

Communications services have experienced and exponential increase in the past fifty years. Starting with the invention of transistor in 1947, and soon after the creation of integrated circuits (IC), the development of high integration technology, with small power requirements, low weight and high response speed were possible. This way, the creation of satellite communications in 1965 was made a reality [1], beginning with the era of wireless communication services.

At present time, wireless and mobile communication services are becoming a very important issue in information technology. We can highlight cellular and wireless networks (GSM – Global System for Mobile, CDMA – Code Division Multiple Access, IEEE802.11x and Hyperlan) as the main wireless and mobile communications means up to this day. These wireless technologies arose thanks to the simultaneous development of both digital communications techniques and semiconductor devices, which allow high integration circuit design of complex signal processing functions encountered in modern communications systems. Thus, efficient mobile communications systems are available, allowing the transmission of information which was unimaginable only ten years ago, and giving rise to huge economic implications (e-commerce, e-business, etc).

As a result, an explosion of new communication services came out, due to this important advance in wireless and mobile communications. Demands for capacity in wireless communications have been rapidly increasing worldwide in order to satisfy the requirements of these new services. Today, most fixed public communications services offer data, audio and multimedia communications. However, for mobile systems, things are not that easy since wireless channels present serious drawbacks of limited available spectrum, and capacity needs cannot be met without significant increase in the spectral efficiency (bit/sec/Hz) of these channels. The prime aspect in the further advance of wireless communications is inescapable: the development of new and efficient communications systems requires improved signaling schemes and a detailed and accurate measurement and characterization of the wireless channel [4].

Today, the main Wireless Local Area Network standard (WLAN) is IEEE802.11b which can achieve 11Mbps. This data rate is enough for applications such as voice, audio and video downloading in real time; nevertheless, for the new envisaged multimedia applications, such as interactive 3D video games, biomedical imaging applications, etc, these data rates are not enough. In addition, mobility is not a characteristic feature of this system. On the other hand, the main mobile communications standards are GSM and GPRS (General Packet Radio Service) which are capable of sending information at bit rate connections ranging between 5.35Kb/s and 171Kb/s. Mobility is the prime feature of these systems, but the bit rate offered is not enough so as to support multimedia communications

in real time and with high quality [14]. As we can see, the trend of wireless systems is to converge in such a way that a communications system supporting both mobility and high data rates for multimedia services is to be available in the short term. This way, wireless systems will become an excellent alternative to the wired counterparts.

As an example, the new IEEE802.16 standard, also known as WiMAX (World Interoperability for Microwave Access) has been created to try to fulfill this requirement. It is claimed that it will support transmission speeds of up to 30 – 130Mb/s in the IEEE802.16 version, and up to 170Mb/s in the IEEE802.16a version. Moreover, it is expected that the coverage area will be of 2 kilometers in IEEE802.16 and up to 30 kilometers in IEEE802.16a, as opposed to the IEEE802.11b which can only offer coverage areas of 50 up to 150 m [15][16]. The considerable increase in bit rate in this new standard makes possible mobile systems with communications services similar to the wired counterpart, i.e. multimedia mobile communications. Nevertheless, it is evident that we need new technologies that make an efficient use of the spectrum, which derives in the need for transmission techniques offering high spectral efficiency. In this thesis, this need is addressed where new transmission techniques which are spectral efficient are investigated.

At the moment, different alternatives exist trying to achieve the design of communications systems with high spectral efficiency. Among these techniques, we can distinguish one that is potentially important and which uses arrangements of antennas at both, transmitter and receiver end. It is known as MIMO systems (Multiple Input – Multiple Output) [5][6].

The MIMO systems theory has demonstrated channel capacities that depend linearly on the number of antenna pairs utilized (in the transmitter and receiver side) [7]. Nevertheless, the detection procedure is very complex since this implies the knowledge of the channel's spatial characteristics. Moreover, the omnidirectional multipath propagation and the Rayleigh fading models are no longer accurate in this kind of scenarios. This means that multiple antenna transmitters and receivers will not work properly if designed without a precise understanding of the spatial-temporal characteristics of the multipath channel.

The detection procedure in MIMO systems is generally known as Space-Time Coding (STC) [8]. MIMO systems were possible when space-time coding algorithms appeared. The principal idea of this approach is to jointly design modulation, coding and equalization. Space-Time coding can be divided into three types: Space-Time Block Codes (STBC) [8], Space-Time Trellis Codes (STTC)[9] and the so-called Bell Labs Layered Space-Time Architecture (BLAST) [10]. Layered Architectures are also referred to as Spatial Multiplexing techniques.

Among the different code techniques in space-time, the BLAST architecture is the one that has demonstrated the advantages of the MIMO systems in a practical way [17].

The BLAST system was proposed by Foschini at the end of the 80s [4]. This architecture approaches the theoretical limits and was called DBLAST (Diagonal Bell Labs Layered Space-Time) [11]. DBLAST architectures use multiantenna arrays and an elegant diagonal layered coding structure in which blocks of information are dispersed across diagonals in space–time. However, DBLAST suffers from computational complexity, making it

inappropriate for a hardware implementation. A simplified version of the DBLAST algorithm is the Vertical BLAST (VBLAST). This approach can reach tens of bits/s/Hz with multiple antennas and reduced complexity. In a VBLAST system, the uncoded data streams are demultiplexed into M substreams, each being transmitted simultaneously by one transmit antenna. At the receiver, the received signals from N received antennas are detected by a decision feedback algorithm. At present time, multiple realizations of the VBLAST exist whose aim is to reduce the computational complexity.

As stated above, the aim of this thesis is to analyze the so-called space-time coding systems for high-rate mobile communications systems. Specifically, one of the goals of this thesis is to analyze the VBLAST architecture from a performance-complexity standpoint. Different approaches of this algorithm are considered namely, the Singular Value Decomposition, the Sorted QR Decomposition and the LS algorithm [12]. These variants are compared to the near optimum Maximum Likelihood (ML) algorithm proposed by Agrell in [13]. This algorithm will be used as a reference to assess the performance of the different VBLAST approaches. This way, a selection of the VBLAST approach presenting the best tradeoff performance-complexity can be made and justified.

The second goal of this work is to envisage some improvements to the VBLAST algorithm without rendering it too complex. The main idea is to improve its performance, as compared to the conventional VBLAST and Agrell's algorithms, while maintaining its complexity as reduced as possible, in order to consider a hardware implementation. This idea is carried out with the addition of a simple convolutional channel coder to the VBLAST architecture. At the receiver, this convolutional code is decoded by means of the Viterbi algorithm.

The document presents simulation results which compare Agrell's algorithm and the VBLAST variants from a complexity and performance point of view. It will be shown that the LS algorithm achieves the best tradeoff. In addition, a new space-time coding scheme is proposed which comprises a convolutional code at the transmitter together with the BLAST encoder, and the Viterbi algorithm concatenated with the VBLAST decoder is used at the receiver. It is shown that this system can achieve a 2dB gain in terms of Block Error Rate (BLER) performance. The computational complexity was estimated and it is shown that a hardware implementation of this system is still possible.

The thesis is organized as follows. In this chapter, an introduction to the subject and a justification of this thesis is given. Chapter two presents an introduction to MIMO (Multiple Input – Multiple Output) systems and their performance in comparison to SISO (Single Input – Single Output) systems. It introduces some basic notations and mathematical tools. Also we present a brief introduction to space-time coding and its theoretical performance. Chapter three presents a near optimum algorithm based on Maximum Likelihood (ML) decoding/detection. This algorithm serves as a benchmark for the other VBLAST-based algorithms discussed here. The BLAST architecture for ST decoding is unveiled in chapter four. Its complexity is still quite important; as a result detailed descriptions of VBLAST variants with reduced computational effort are presented in chapter five. In this chapter we study the performance-complexity characteristics of each variant and discuss which algorithm is best suited for hardware implementations. In chapter

six, the design of a simple space-time coding scheme consisting of a conventional channel coding scheme and the VBLAST algorithm is proposed so as to reduce the BLER performance of the overall system. Finally, conclusions and future work/perspectives are discussed in chapter seven.

REFERENCES

[1] J. G. Proakis, M. Salehi, "Communication Systems Engineering", Second Edition, 2002.
[2] A. Paulraj, R. Nabar, D. Gore, "Introduction to Space-Time Wireless Communications", Cambridge Editors, 2003.
[3] WI-FI Planet: The source for WI-FI business and technology. http://www.wi-fiplanet.com/tutorials/
[4] G. J. Foschini: "Layered space-time architecture for wireless communication in environment when using multiple antennas". Bell Lab. Tech. J., 1996. I. (2), pp. 41-59.
[5] M. Dobler, "Space-Time Trellis Codes for Hiperlan2", IST-MIND Workshop. King's College London, 7 Oct 2002.
[6] R. Gaspa, J. R. Fonollosa, "Comparison of Different Transmit Diversity Space-Time Code Algorithm", Department of Singal Theory and Communications. Universitat Politecnica de Catalunya.
[7] A. Pascua, M. A. Lagunas, "Avances en Técnicas de Codificación Espacio-Temporal en Sistemas de Comunicaciones Móviles", URSI 2002.
[8] S. M. Alamount, "A simple transmit diversity technique for wireless communications", IEEE J. Sel. Arcas in Comm., pp. 1451 – 1458, Oct. 1998.
[9] V. Tarokh, H. Jafarkhani, A. R. Calderbank, "Space-time block codes for orthogonal designs", IEEE Trans. On Inf. The., pp. 1456 – 1567, July 1999.
[10] G. J. Foschini, "Layered Space-time Architecture for wireless communications in a fading environment when using multiple antennas", Bell Labs Technical Journal, Vol 1, No. 2, pp. 41 – 59, 1996.
[11] B. Vucetic, J. Yuan, "Space-Time Coding", WILEY, 2002.
[12] M. Bazdresch, R. Guisantes: "Least-Squares Techniques Applied to VBLAST Receivers". Admission in process.
[13] E. Agrell, T. Eriksson, A. Vardy: "Closest point search in Lattices". IEEE Transactions on Information Theory. Vol. 48. No. 8. August 2002, pp. 2201 – 2214
[14] Z. Hunaiti, V. Garaj, W. Balachandran, F. Cecelja, "An Assessment of 3G link in a Navigation System for Visually Impaired Pedestrians", 15th International Conference on Electronics, Communications and Computers, CONEILECOMP 2005, pp. 7-9.
[15] C. Kamath, "Hiperlan/2: High Performance Radio Local Area Network/2", Presentation available on: http://www.llnl.gov/CASC/people/kamath/
[16] "802.11 Wireless Ethernet Standards", http://stdsbbs.ieee.org/group/802.11
[17] M. J. Gans, N. Amitay, Y.S. Yeh, Y.S. Hao Xu, T.C. Damen, R.A. Valenzuela, T. Sizer, R. Storz, D. Taylor, W.M. MacDonald, C. Tran and A. Adamiecki; "Outdoor BLAST measurement system at 2.44 GHz: Calibration and initial results"; IEEE Selected Areas on Communications, vol 20., pp. 570-583, Apr 2002

2

CHARACTERIZATION AND PERFORMACE OF MULTIPLE-INPUT MULTIPLE-OUTPUT WIRELESS COMMUNICATION SYSTEMS

2.1 INTRODUCTION

In this chapter concepts of multiple-input multiple-output (MIMO) versus single-input single-output communications systems (SISO) will be analyzed. In MIMO systems, signaling goes beyond simple antenna diversity. Data streams of symbols are broken and separated into M new sub-streams (where M is the number of transmit antennas) which determine the lower number of multipaths in the free space.

This chapter presents an introduction to MIMO concepts and signal processing, emphasizing the potential gains in channel capacity by increasing the spatial signals on the transmitter and receiver. The second part of the chapter presents a brief panorama of space-time coding techniques as a preview to the near-optimum closest point (CP) decoder algorithm which will be described in the next chapter. We will see that simpler suboptimum decoding techniques such as the BLAST algorithm are needed in order to render MIMO systems feasible. The major topics discussed in this section are listed below:

- Section 2.2: Characterization of MIMO channel
- Section 2.3: Channel capacity for conventional antenna arrays
- Section 2.4: Performance limits of MIMO systems
- Section 2.5: Introduction to Space-Time coding
- Section 2.6: Conclusions.

2.2 CHARACTERIZATION OF MIMO CHANNELS

MIMO systems are defined as an array of multiple antennas at both, receiver and transmitter. This kind of systems is also called MEA (Multiple Element Arrays) [4]. Systems with a single antenna at the transmitter and receiver can obtain excellent performance with conventional channel coding and optimized modulation techniques at the expense of increased bandwidth. Nevertheless, MIMO systems can obtain similar performance with a couple of antennas and simple modulation and without increasing bandwidth. This makes the system efficient and with low required transmit power.

In MIMO systems, M antennas are used to transmit information signals and N antennas at the receiver detect them. The input data stream is divided into M sub-streams, each of which is transmitted by one of the M transmit antennas available. The M transmit antennas

are also synchronized, and use the same frequency band, which basically can be viewed as M virtual transmission channels with the same signal constellation. It is important to notice that in these systems the transmit power is divided into the M transmit antennas; therefore, the average symbol energy is the same. In a general MIMO system, any number of transmit and receive antennas can be used; however, in this thesis it is assumed that the number of received antennas is equal or greater than the number of transmit antennas, i.e. $N \geq M$.

The channel is assumed to be frequency-flat with block Rayleigh fading (some Rician and fast Rayleigh fading characteristics are discussed in appendix A). This means that the channel impulse response is random but it remains constant during the transmission of a block. This block comprises L information symbols. In addition, the channel is corrupted by AWGN noise. These assumptions are not unrealistic since this scenario is usually encountered in indoor applications such as WLAN environments. Mathematically, the channel can be represented by the channel matrix $\boldsymbol{H}$ whose elements $\boldsymbol{h_{ij}}$ represent the link between the *i*-th transmit antenna and the *j*-th receive antenna.

Next section will show us all the fundamentals of MIMO systems and some of the mathematical aspects. We must consider that for any WLAN system, the normal supposition is that distances are short and the fading is constant. Also, in indoor environments, line of sign is very rare, so it is correct and very close to reality consider block Rayleigh fading with AWGN noise.

2.2.1 BASIC DEFINITIONS AND FUNDAMENTALS

2.2.1.1 Multipath Propagation

In wireless communications, there are a lot of objects, such as houses, buildings, trees, which act as reflectors for the radio waves. They produce reflected waves with attenuated amplitudes and phases. As a modulated signal is transmitted, it could have reflections that create new waves with multiple directions and multiple propagation delays. These reflected waves are called multipath waves [7]. These multipath waves are collected by the receiver antenna at any point in space; they may be combined in a constructive or destructive form, depending on the random phases. The sum results in a spatially varying wave field. When the handset is moving, the receiver can get waves which vary widely in amplitude and phase, and when the handset is stationary, the amplitude variations are due to the movement of surrounding objects. This amplitude fluctuation of the received signal is called signal fading.

2.2.1.2 Doppler Shift

Doppler shift consist in a shift in frequency caused by the relative motion between transmitter and receiver. It is proportional to the speed of the mobile unit. Consider a single tone of frequency f_c which is transmitted; the received signal consists of only one wave

coming with an angle θ with respect to the direction of the vehicle motion. The Doppler shift denoted by f_d is given by [15]

$$f_d = \frac{v * f_c}{c} \cos\theta \qquad 2.1$$

where v is the speed of the vehicle and c is the speed of light. In a multipath propagation environment, the maximum Doppler shift is

$$f_{d\max} = \frac{v * f_c}{c} \qquad 2.2$$

This shift is referred to as maximum fade rate. As a result, the single tone transmitted gives rise to a received signal with a spectrum of nonzero width, which is called frequency dispersion of the channel.

2.2.1.3 Diversity

Diversity is used to reduce the multipath effects and improve the reliability of transmission without increasing the transmitted power or sacrificing bandwidth. This technique requires multiple replicas of the signal at the receiver.

The basic idea is that two or more independent samples of a signal are transmitted. These will fade in an uncorrelated manner, in such a way that some samples are severely faded while others are less attenuated. The probability of all the samples being simultaneously below a given level is lower than the probability of a single individual sample. Thus, combinations of the various samples result in a greatly reduced fading and improved reliability of transmission [9][10].

Diversity techniques are used in mobile communications in a number of ways in order to improve performance. According to the domain where diversity is introduced, the techniques are classified into time, frequency and space diversity. The reader is referred to appendix B for a more detailed discussion.

Now, we will discuss how these phenomenons are related to MIMO communications systems.

2.2.2 MIMO CHANNEL MATRIX.

In MIMO systems, it is useful to use a vector/matrix notation. A vector of received signals, ***y(t)*** (as seen at the input of the N received antennas), may be estimated from the transmitter vector, ***x(t)***. The output vector is related to the input vector by the channel transfer matrix ***H(τ,t)***, where $\boldsymbol{\tau}$ is the channel dispersion and $\boldsymbol{t}$ is the time where transmission occurs according to the following equation [1]

$$\boldsymbol{y(t)} = \frac{1}{2}\int_{-\infty}^{+\infty} \boldsymbol{H(\tau(t)x(\tau)}d\tau \qquad 2.3$$

where

$$\boldsymbol{y(t)} = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \dots \\ y_N(t) \end{bmatrix},\ \boldsymbol{x(t)} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \dots \\ x_M(t) \end{bmatrix},\ \boldsymbol{H(\tau(t)} = \begin{bmatrix} h_{1,1}(\tau,t) & h_{2,1}(\tau,t) & \dots & h_{M,1}(\tau,t) \\ h_{1,2}(\tau,t) & h_{2,2}(\tau,t) & \dots & h_{M,2}(\tau,t) \\ \dots & \dots. & \dots & \\ h_{1,N}(\tau,t) & h_{2,N}(\tau,t) & \dots & h_{M,N}(\tau,t) \end{bmatrix} \qquad 2.4$$

Matrix $\boldsymbol{H_{ij}(\tau,t)}$ is the channel impulse response of the virtual channel from the *i*th transmit antenna to the *j*th receive antenna. For simplicity, as stated previously, if we assume block fading, the MIMO channel can be represented as a time invariant channel model. In this case, there is no frequency or time dependency in the channel; it results in a constant $\boldsymbol{H}$ matrix, which makes the estimation of the received signal as

$$\mathbf{y}(t) = \frac{1}{2}\mathbf{H}\mathbf{x}(t) \qquad 2.5$$

2.2.3 PROCESSING THE MIMO SIGNAL.

The MIMO system is a complicated structure, whose better representation is based on matrix/vector analysis. Processing at the transmitter and receiver is needed in order to produce the set of received signals with highest overall capacity. All this operations will be discussed in a basic form in this section, and taking as a reference the block diagram shown in figure 2.1.

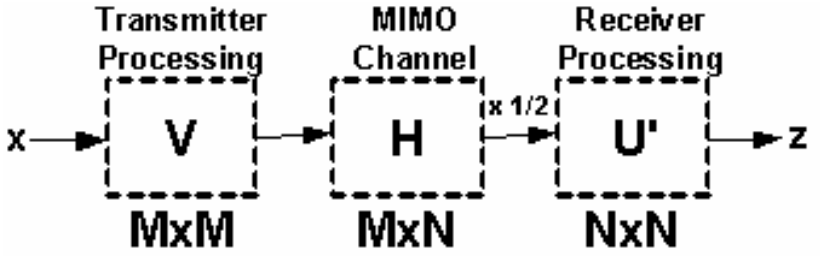


Figure 2.1: The matrix representation of signal processing operations in a MIMO system.

In communications systems, the magnitude of a signal vector represents the total signal power. In figure 2.1, block **V** is the processing operation on the transmitted signal vector, ***x(t)***, to produce a new set of signals to be fed into the array of transmitter antennas, **V*x(t)***. Matrix **V** is a MxM unitary matrix with the property **VV'=I** (where operator **'** means the transpose matrix), which means that no change of the geometrical length of vectors is incurred, i.e., we are not adding any power to the total transmitted signal. In MIMO systems, transmit power does not depend on the number of transmit antennas.

After the channel operates on the transmitted signal to produce **HV*x(t)***, the receiver operates in the incoming signal with matrix **U'**, so we get the final output signal vector

$$\boldsymbol{z(t)} = \frac{1}{2}\overbrace{\boldsymbol{U'HV}}^{\boldsymbol{D}}\boldsymbol{x}(t) \qquad 2.6$$

It is very important to remember that operator U' is an NxN unitary matrix with the property **U'U=I**. This condition, as we saw, assures that we have not added or subtracted any signal power.

Now, let us propose an algorithm on the MIMO channel which makes that operation ***U'HV*** may be replaced by a diagonal matrix ***D***. This new operation consists of diagonal elements, all of which are positive and constant. Matrices ***U*** and ***V*** are defined so that the transmission operation generates matrix ***D*** (recall that the number of transmit antennas must be lower or equal to the number of receive antennas $N \geq M$). This way, the estimation process at the receiver is greatly reduced.

Mathematically, this means that the MIMO channel can be viewed as a set of M separate channels from transmitter to receiver with the condition $N \geq M$. If we have M > N the diagonal matrix would support a maximum of N separate channels. Thus, the number of separate channels with M transmit and N receive antennas is $\min(M, N)$, which means that the number of separate channels is the minimum between M and N.

The signal processing steps described above can be done in several ways. In this thesis, we present four detailed descriptions: Maximum Likelihood (ML) based in the closest point algorithm and three variants of Vertical Bells Labs Layered Space-Time Architecture (VBLAST) [16]. They will be described in following chapters.

2.2.4 PHYSICAL INTERPRETATION OF MIMO CHANNELS

As we saw, a MIMO channel has the particular characteristic of creating separate virtual channels that depend only on the number of transmit and receive antenna pairs, which obviously is reflected in the channel capacity obtained by the system [19]

$$C \approx M * B * \log_2\left(1 + \frac{N}{M} SINR_0\right) \qquad 2.7$$

Notice that equation 2.7 is different to SISO systems because even though the SINR increases by a factor N/M, the overall capacity is multiplied by M channels. The main idea here is to transmit data using many different low-powered channels rather than using one single high-powered channel.

The physical interpretation of a MIMO channel is shown in figure 2.2. Matrix ***V*** operates on an input vector ***x(t)***. For example, if we have a frame with M bits of information, then

each bit in the frame will be transmitted by one transmit antenna. Each column in ***V*** contains the M amplitudes and phases for a symbol sent through the array of M transmit antennas. From matrix V is important to visualize that each antenna is defined by a unique spatial radiation pattern, so each of the columns have unique amplitudes and phase weightings. For instance, column 1 in ***V*** assigns a radiation pattern to $x_1(t)$, column 2 assigns a radiation pattern to $x_2(t)$, and so on.

Matrix ***U*** operates over the received signals from the N receiver antennas. Each column of ***U*** defines a radiation pattern that recognizes one of the original symbols sent by the transmitter (see Fig. 2.2). The weighting operation of ***V*** and ***U*** using unique antenna patterns to each separated data streams creates individual symbols from the others that come from the multipath channel allowing a correct estimation of each transmitted symbol by the corresponding receive antenna.

Remember that multipath channel is very complicated in practice, and antenna patterns have finite resolution, so the MIMO processing can not be simplified with a few dominant scatters and get perfect separation of symbols; however the basic concept (which is the main propose of this sections) has been explained.

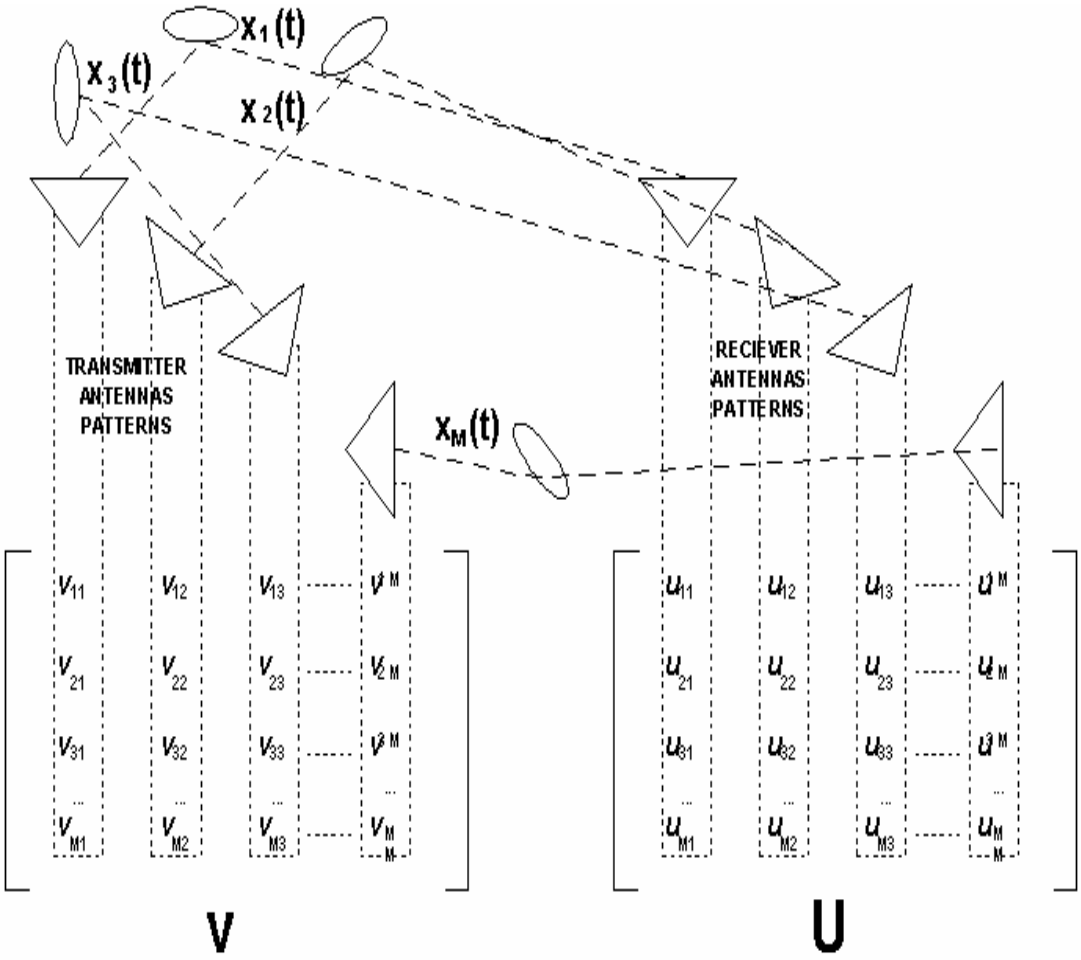


Figure 2.2: Physically, MIMO spatial coding sends different symbol streams to and from different directions of space.

2.3 CHANNEL CAPACITY FOR CONVENTIONAL ANTENNA ARRAYS

We have explained the fundamental concepts of MIMO systems and receive processing; however, the main question is the increase in the channel capacity achieved by these systems. This section discusses the performance of multiple antenna systems, starting from the SISO, SIMO and MISO systems, the ancestors of space-time block codes and MIMO architectures.

2.3.1 SINGLE-INPUT SINGLE-OUTPUT (SISO).

The simplest and the baseline for comparison will be SISO system, which is shown in figure 2.3. SISO systems use only one transmit antenna with P_T input power. We can estimate the Shannon channel capacity for this system by the expression [11]

$$C = B \cdot \log_2(1 + SINR_0) \qquad 2.8$$

Where B is the available bandwidth and $SINR_0$ is the average signal to noise plus interference ratio. Note that from digital communications theory, the available bandwidth limits the symbol rate, but not necessarily the bit rate. It is very common to forget that we can transmit more that one bit on a single complex baseband symbol. While symbols with more than two states are more susceptible to noise and interference, higher SINR makes reliable transmission possible and helps to approach Shannon capacity limit in systems.

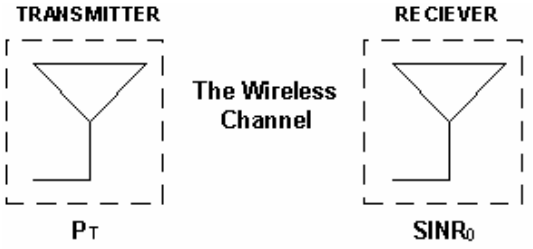


Figure 2.3: A SISO antenna configuration.

2.3.2 SINGLE-INPUT MULTIPLE-OUTPUT (SIMO).

A SIMO system is shown in figure 2.4. N number of antennas are used at the receiver, producing N copies of the faded signal at the receiver. Although it is almost impossible, we must say that if N signals have the same amplitude, they can produce an N^2 increase in signal power. Of course, there are not N sets of noise/interference that can add together as well. Noise and interference add incoherently (not with the same amplitude) to create an increase in noise power less that N. Therefore, the increase in SINR is [3]

$$SINR = \frac{N^2[SignaltoPower]}{N[Noise + Interference]} = NSINR_0 \qquad 2.9$$

So, channel capacity for this system is approximately:

$$C \approx B \cdot \log_2\left(1 + N \cdot SINR_0\right) \qquad 2.10$$

which is a little higher that SISO system. This difference at the receiver side implies an increase in the signal to noise interference ratio when multipath channels are used. As an example, when we are in open-space, the receiver antenna elements are phased together to form an array with maximum gain in the direction of signal arrival. This peak causes a subsequent increase in SINR and channel capacity. But if we are in a multipath channel, the direction of signal arrival may be dispersed throughout the azimuth. Figure 2.4 show the SIMO diagram.

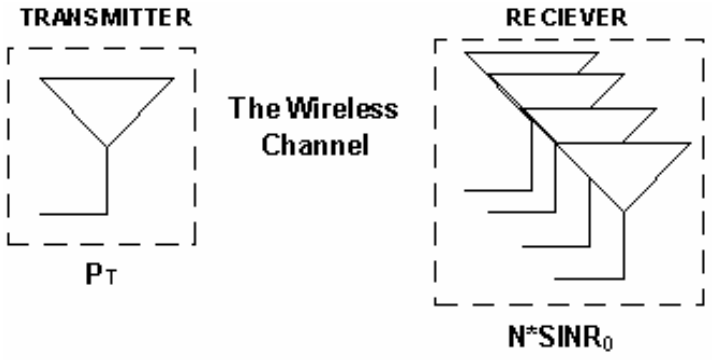


Figure 2.4: A SIMO antenna configuration.

2.3.3 MULTIPLE-INPUT SINGLE-OUTPUT (MISO).

In this system (see Fig. 2.5), M antennas are used at the transmitter. The total transmitted power is split into M transmitters. This way, although the power per antenna drops, signals are transmitted in such a way that they add coherently at the receiver, producing a net M-fold increase in SINR, as compared to the SISO case. Since we have only one receiver, noise/interference power should be the same. The overall increase in SINR becomes [1]:

$$SINR \approx \frac{M^2\left[Signal - Power/M\right]}{Noise + Interference} \equiv MSINR_0 \qquad 2.11$$

The channel capacity for this system is [3]:

$$C \approx B \cdot \log_2\left(1 + M \cdot SINR_0\right) \qquad 2.12$$

This channel capacity is higher than the SISO case. The physical interpretation at open-space is that the M transmitter antennas are phased and directed towards the receiver. The directivity created, causes an increase in the SINR and channel capacity. In a multipath scenario and with channel state information known at the transmitter, it is possible to change the phases and signal weighting on the antennas in order to maximize receiver SINR and channel capacity.

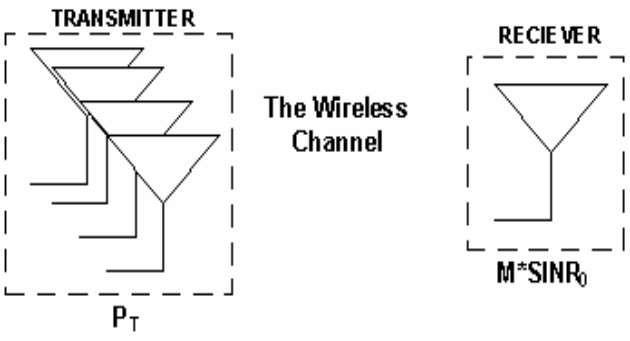


Figure 2.5: A MISO antenna configuration.

2.3.4 MULTIPLE-INPUT MULTIPLE-OUTPUT (MIMO).

Finally, the general structure of a MIMO system is presented in figure 2.6. It consists of M transmit and N receive antennas. Signals are to be transmitted and received with such phasing as to maximize the total signal power through the wireless channel. The result is (approximately) an M*N-fold increase in the SINR of the received signal. The channel capacity for this kind of systems is given by

$$C \approx M * B * \log_2\left(1 + \frac{N}{M} SINR_0\right) \qquad 2.13$$

From this equation we can observe that maximum channel capacity is given when multiple-inputs and multiple-outputs are placed in the same system. A physical interpretation at open-space is that both transmitter and receiver use their multiple antennas in a phased array configuration, increasing the SINR substantially. When in a multipath scenario, in order to achieve the maximum channel capacity, transmit and receive diversity is used to achieve the highest possible SINR in the overall link.

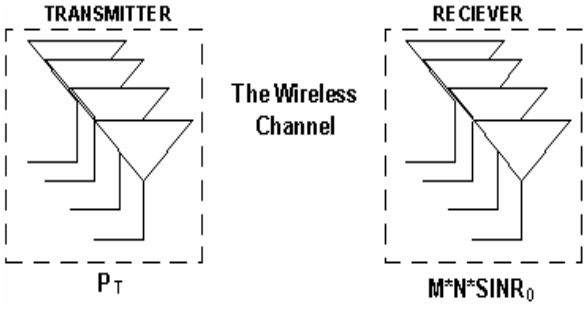


Figure 2.6: A MIMO antenna configuration.

As a comparison between all systems, consider a system with M=2 transmit antennas and N=4 receive antennas. This implies that the SIMO system has one transmit and four receive antennas, the MISO system has two transmit and one receive antennas and the MIMO system has two transmit and four receive antennas. Figure 2.7 shows the channel spectral efficiency as a function of the SNR at the receiver for each one of the four systems. We can observe from the figure that MIMO systems can obtain much better performance. The

number of bits we can transmit is increased almost four times with respect to SISO systems. Also, the figure shows that when we introduce diversity at the receiver, the efficiency is increased faster that if the diversity is introduced at the transmitter. It must be noted that MIMO systems can be either square (same number of receive and transmit antennas) or not; but for this thesis, we assume that the number of transmit antennas is smaller or equal than the number of receive antennas.

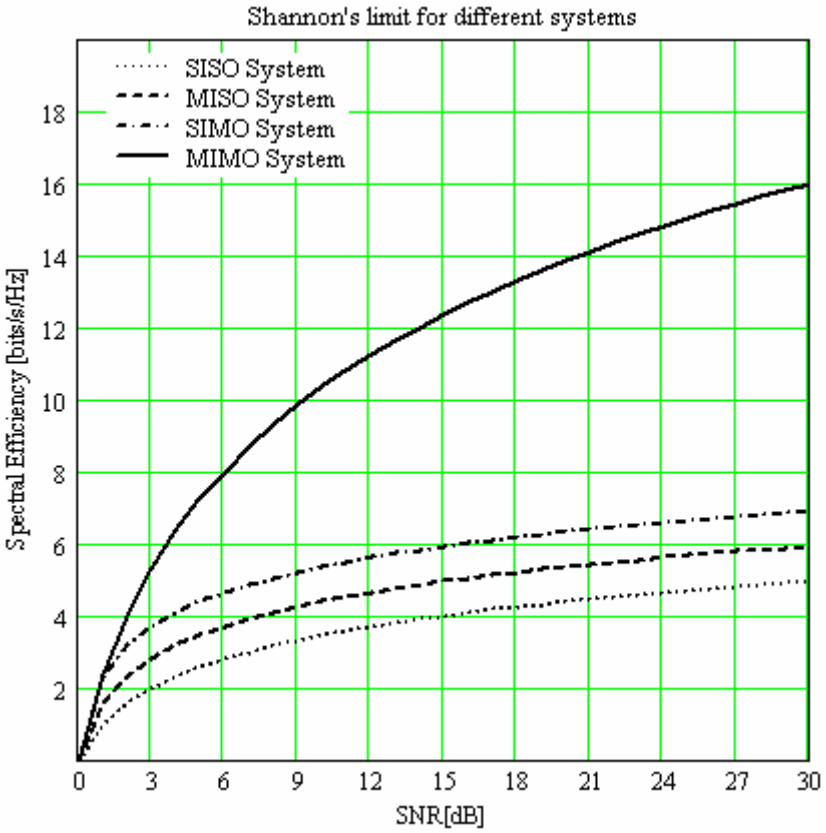


Figure 2.7: Comparison between different kinds of array system.

2.4 FADING ASSUMPTIONS IN MIMO SYSTEMS

As we saw in last section, MIMO systems set individual channels between pairs of transmit and receive antennas, which are modeled by an independent flat Rayleigh fading process. In this section, we limit our attention to the analysis of the narrowband channels, so that they can be described by frequency flat models.

Rayleigh models are realistic for environments with a large number of scatterers. In channels with independent Rayleigh fading, a signal transmitted from every transmit antenna appears uncorrelated at each of the receive antennas. As a result, signals corresponding to every transmit antenna have a distinct spatial signature at the receive antenna [2].

The analysis of the performance of MIMO systems in more realistic situations where the channel matrix coefficients are random distributed can assume that channel coefficients are

estimated at the receiver but unknown at the transmitter. Furthermore, we assume that the elements of channel matrix are zero mean Gaussian complex random variables, each with variance of ½. So, each entry of the channel matrix has a Rayleigh distributed magnitude, uniform phase and expected magnitude square equal to unity, i.e. $E\left[\left|h_{ij}\right|^2\right]=1$. The probability density function (PDF) for a Rayleigh channel with distributed random variable is $z=\sqrt{z_1^2+z_2^2}$, where z_1 and z_2 are two statistically independent Gaussian random variables, each with zero mean and variance σ_r^2 [3]

$$p(z)=\frac{z}{\sigma_r^2}e^{\frac{-z^2}{2\sigma_r^2}},z\geq 0 \qquad 2.14$$

where σ_r^2 is usually normalized to ½. According to how frequently the channel coefficients change, three kinds of scenarios can be distinguished, namely

1. *Fast Fading Channel.* The entries in the channel matrix change randomly at the beginning of every symbol interval T and are constant during one symbol interval.
2. *Block Fading Channel.* The entries in the channel matrix are random and constant during a fixed number of symbol intervals, which is shorter than the total transmission.
3. *Slow or Quasi-Static Fading Channel.* The matrix H is random but constant during transmission.

As indicated above, we will consider only block fading channels. Appendix A gives a more general treatment of fading channels in situations other than block fading.

2.5 INTRODUCTION TO SPACE-TIME CODING

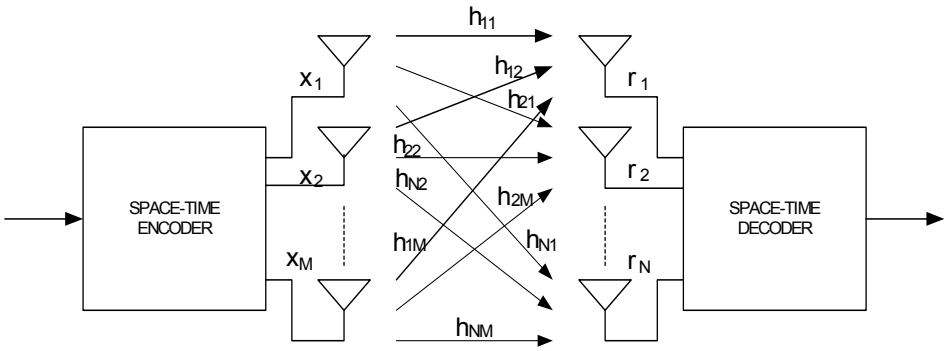
In previous section we saw that information capacity of wireless communications systems can be increased considerably by employing multiple antennas at the transmitter and receiver, where capacity grows linearly with the minimum number of antennas.

One fact of wireless communications is inescapable: the development of new systems requires that the channel is measured and modeled to an increasingly higher degree of detail. We can no longer make approximations about the spatial channel, such as omni-directionality in the multipath propagation and Rayleigh fading. Multiple antenna transmitters and receivers will not function properly if designed without an understanding of the spatio-temporal characteristics of the multipath channel.

The multipath propagation has two unpredictable types of behavior: frequency selectivity and spatial selectivity. The first of them is caused by multipath components which arrives from multiple and different delays. The second is caused by multipath components arriving from different directions in space. Frequency selectivity is a well known phenomenon, as opposed to spatial selectivity. Spatial selectivity is an alternative against fading and has been used a lot with the new space-time applications.

Advances in channel coding make it possible to approach Shannon capacity limits [17] in systems with single antenna links (Single Input – Single Output) [19]. However, as we described in previous paragraphs, even more significant improvements can be achieved with MIMO channels. It has been demonstrated that cellular SISO systems can achieve spectral efficiencies between 1-2 bits/sec/Hz whereas MIMO systems with 8 antennas at each side can obtain 37 bits/sec/Hz at an SNR of 20 dB [20].

In order to approach those capacities, MIMO channels employ space-time coding [5][6]. Space-time coding techniques are used with multiple antenna arrays. Coding is performed in spatial and temporal domains, spatial domain means multiple antennas and temporal domain means various time slots or time periods. The spatial correlation is used to exploit the MIMO channel fading and minimize transmission errors at the receiver. This kind of transmission can achieve transmit diversity and power gain over uncoded systems without sacrificing the bandwidth. There are basically three types of coding structures: space-time block codes (STBC)[18], space-time trellis code (STTC) [13] and space-time layered codes (LST) [16]. A general space-time coding system is shown in figure 2.8. The main process in all schemes is the exploitation of the multipath and to obtain performance gains [8].



2.8: General scheme of a space-time system

2.5.1 SPACE-TIME BLOCK CODES (STBC)

The first space-time coded system was created by Alamount [18] (figure 2.9). The general idea is to consider as input to the encoder a group of k symbols transmitted at different time slots. This can be seen as arranging the symbols into a code matrix, where columns represent the symbols transmitted by one of the transmit antennas and rows represent the time intervals of transmission (see figure 2.9). The code rate of the system is defined as $r = k / p$; where p is the number of transmission intervals and k is the number of transmit antennas (number of columns in the code matrix).

The construction of the transmission matrix consists of the original information symbols and their conjugates. This space-time coding has a unitary code rate (one symbol per transmission interval), that is, the code does not require any bandwidth expansion. Moreover, it is possible to obtain diversity gain and sometimes code gain which makes an efficient coded transmission.

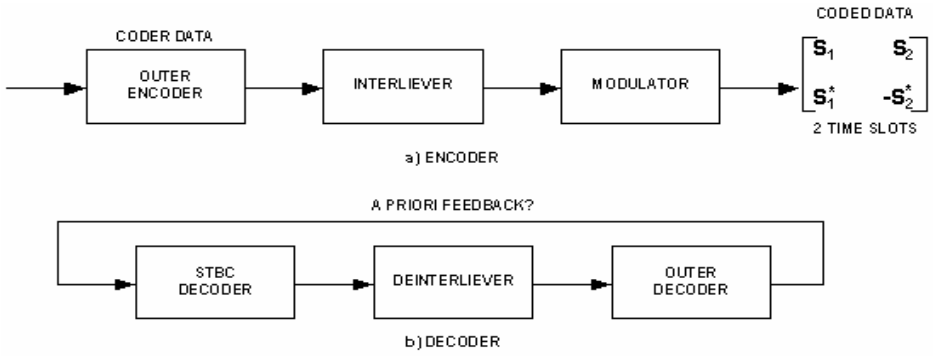


Figure 2.9: Coding and Decoding Alamount System.

2.5.2 SPACE-TIME TRELLIS CODES (STTC)

The second type of space-time coding systems were proposed by Tarok et al. [11] and called STTC (see figure 2.10). STTC develop channel coding, modulation and transmit diversity simultaneously. The system can obtain important gains but with increased bandwidth; in addition, compared to STBC, its complexity is much higher [8].

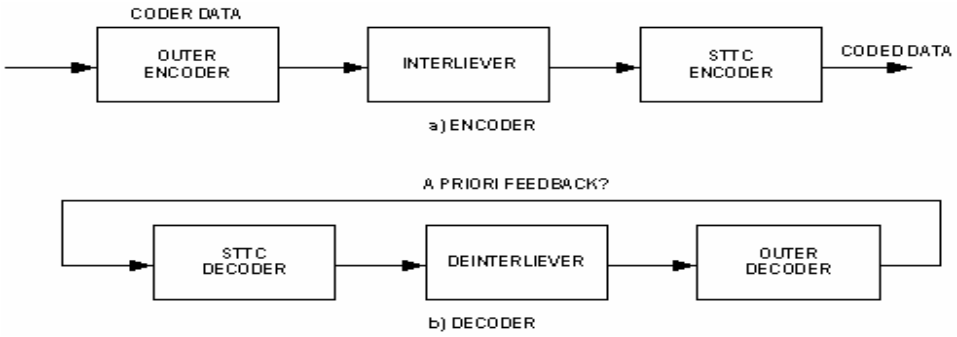


Figure 2.10: Coding and Decoding Tarok system.

2.5.3 LAYERED SPACE-TIME CODES (LST)

LST codes were proposed by Foschini in the 80's [16]. In order to increase the bit rate in the systems, LST introduced an array of MxN antennas, each one of them transmits a sub-stream. Consider the example of a 3x3 antenna array.

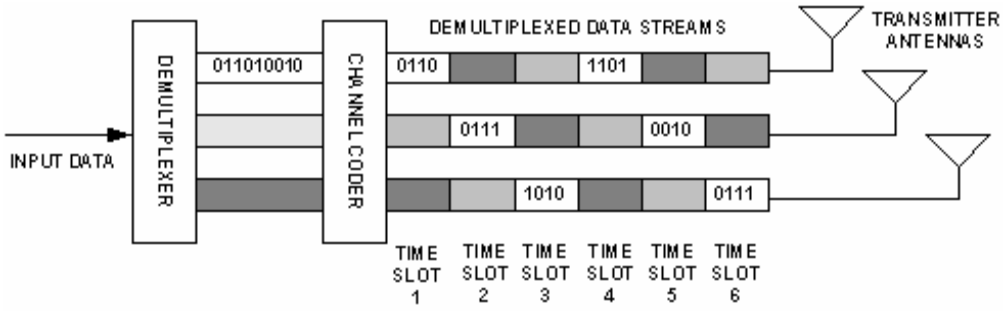


Figure 2.11: 3x3 MIMO system with Layered Architecture at the transmitter.

The 3x3 MIMO system is shown in Figure 2.11. In order to increase the bit rate, the input stream is introduced to the demultiplexer, which creates at its output three new data sub-streams, each with equal number of bits. The data sub-streams are arbitrarily coded onto digital symbol streams. For sake of simplicity, the constellation used is BPSK (Binary Phase Sift Keying).

It is possible to take these three data streams and modulate them directly to the carrier frequency and send them out through the transmit antennas, but the basic problem here is the presence of low-powered channels through one or more of the transmit antennas. The channels can not be separated with equal SINR at the receiver by simple inversion of the channel matrix $\boldsymbol{H}$. We need a channel estimation feedback because the transmitter has no way to know which of the separate channels is able to support the data rate of the transmitted streams.

In order to solve this problem, Foschini introduced a transmitter architecture that cycles the sub-streams [16]. Each transmitted stream takes turn through one of the transmit antennas, spending a time-slot at each antenna element and cycling back to its original antenna element every three time slots. After sufficiently large time slots, each transmitted data stream experiences the same average channel conditions, which can be viewed as being transmitted with equal power.

When information has passed through the channel, each stream is detected from the receive antennas, one at a time. For antenna one, the algorithm extracts the first signal according to the expression [1]

$$\boldsymbol{x}_1(t) = \boldsymbol{u}_1 \cdot \boldsymbol{y}(t) = \boldsymbol{u}_1 \cdot \left[\boldsymbol{H}_1 \quad \boldsymbol{H}_2 \quad \boldsymbol{H}_3\right]\boldsymbol{x}(t) = \left[1 \quad 0 \quad 0\right]\boldsymbol{x}(t) \qquad 2.14$$

We apply this equation to the other two receive antennas. However, we can obtain signal gain through interference cancellation. Since we already know $\boldsymbol{x}_1$, this value is subtracted from vector $\boldsymbol{y(t)}$. Next expression shows how the process is affected with this

$$\boldsymbol{x}_2(t) = \boldsymbol{u}_2 \cdot \boldsymbol{y}(t) = \boldsymbol{u}_1 \cdot \left[0 \quad \boldsymbol{H}_2 \quad \boldsymbol{H}_3\right]\boldsymbol{x}(t) = \left[0 \quad 1 \quad 0\right]\boldsymbol{x}(t) \qquad 2.15$$

We can note that the decision criterion is diminished in one less interferer. A simple point of view is that we are solving a linear equation system; thus, removal of the first estimated symbol provides an extra degree of freedom for constructing $\boldsymbol{u}_1$. Since two independent estimates are available, the system can use combining algorithms to achieve diversity gain. The potential performance for this second data stream is equivalent to maximum ratio combining (MRC) diversity using two branches [12].

For the last signal it is clear that three independent estimates will exist. MCR diversity gain using three signal branches is possible. This interference cancellation algorithm strips away previously detected signals and in the process produces increasingly reliable estimates of the next detection with higher average SINR.

Figure 2.12 shows the signal extraction with interference cancellation at the receiver. As we can observe, the receiver buffers the incoming streams and extracts them in steps, each of them with duration of one time slot. At every step, the receiver detects a layer of data and spans other three time slots when the receiver has returned to detect the layer corresponding to stream 1.

The layers are decoded and multiplexed onto a final stream of output data. When the receiver detects stream 1, each time slot has had a different interference removed from the data stream. In time slot 1, two other layers have been subtracted during previous time slots, so this layer contains the most reliable data for layer 1. In time slot 3, none of the other layers have been subtracted, so this layer is the least reliable layer.

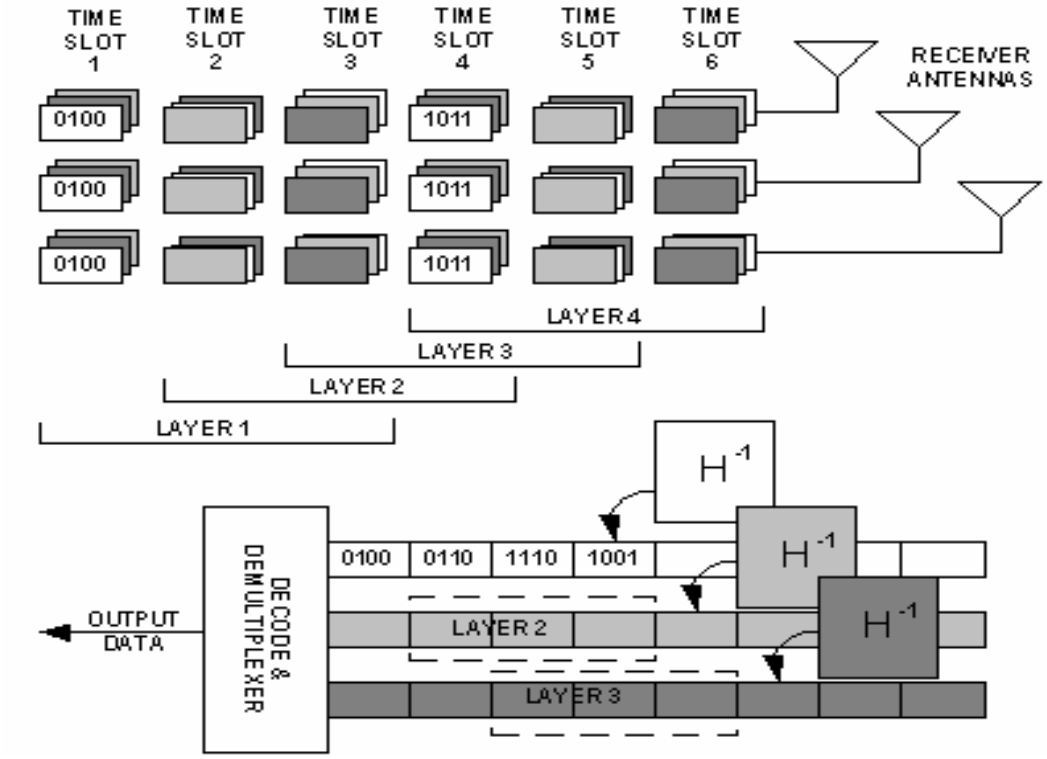


Figure 2.12: 3x3 MIMO system with Layered Architecture

2.6 CONCLUSIONS

In this chapter, MIMO systems have been defined as arrays of antennas in both transmitter and receiver, which can let us achieve performance improvements in both bit error rate and bit rates that other systems cannot obtain. The most important characteristic of these systems is diversity in space and time. Those techniques, applied together, let systems transmit information by multiple antennas and multiple slots of time. The diversity and multipath fading channel performance have been analyzed.

In the next chapter, we turn our attention to a near optimal ML decoder based on the closest point algorithm proposed by Agrell.

REFERENCES

[1] G. D. Durgin, "Space-Time Wireless Channels", Prentice Hall PTR. 2003.
[2] P. VanRooyen, "Advances in Space-Time Processing techniques open up mobile APPS", EE Times, Nov. 8, 2002. http://www.commsdesign.com/story/OEF20021107S0021
[3] B. Vucetic, J. Yuan, "Space-Time Coding". WILEY. 2004.
[4] D. Chizhik, G. J. Foschini, M. J. Gans, R. A. Valenzuela, "Keyholes, Correlations and Capacities of Multiantenna Transmission and Receive Antennas ", IEEE Transactions on Wireless Communications, Vol. 1, No. 2, April 2002.
[5] V. Tarokh, N. Seshadri and A.R. Calderbank, "Space-time codes for high data rate wireless communications: performance criterion and code construction". IEEE Trans. Inform. Theory, vol. 44, no. 2, pp. 744-765. Mar. 1998.
[6] N. Seshadri, V. Tarokh, A. R. Calderbank, "Space-Time Codes for Wireless Communications: Code Construction", 0-7803-3659-3, 1997.
[7] J. G. Proakis, "Digital Communications", 4th Ed., Mc. Graw-Hill, New York, 2001.
[8] V. Tarokh, A. Naguib, N. Seshadri, A. R. Calderbank, "Space-Time Codes for High Data Rate Wireless Communication: Performance Criteria in the Present of Channel Estimation Errors, Mobility and Multiple Paths", IEEE Transactions on Communications, Vol. 47, No. 2, February 1999.
[9] K. F. Lee and D. B. Williams, "A Space-Frequency Transmitter Diversity Technique for OFDM Systems," in *Proc. IEEE GLOBECOM*, EU, November 2000, pp. 1473-1477.
[10] Michigan Technological University, 2001. http://www.geocities.com/hansadhwani8/smartantennas/diversity.html
[11] V. Tarokh, H. Jafarkhani, R. Calderbank, "Space-Time Block Coding for Wireless Communications: Performance Results", IEEE Journal on Selected Areas in Communications, Vol. 17, No. 3, March 1999.
[12] M. A. Fkirin, A. F. Al-Madhari, "Prediction of time-varying dynamic processes", The international Journal of Quality & Reliability, Vol. 14, iss. 5, pp. 505, 1997.
[13] Z. Hong, K. Liu, A. M. Sayeed, R. W. Hearth Jr., "A General Approach to Space-Time Trellis Codes", Office of Naval Research #N00011-01-1-0825 and National Science Foundation CCR-9875805.
[14] www.mathcad.com
[15] H. D. Young, R. A. Freeman, "University Physics", Addison-Wesley Publishing Company. INC. 1996.
[16] G. J. Foschini, "Layered Space-time Architecture for wireless communications in a fading environment when using multiple antennas", Bell Labs Technical Journal, Vol 1, No. 2, pp. 41 – 59, 1996.
[17] C. E. Shannon, "Mathematical Theory of Communications", University of Illinois Press, 1963.
[18] S. M. Alamount, "A simple transmit diversity technique for wireless communications", IEEE J. Sel. Arcas in Comm., pp. 1451 – 1458, Oct. 1998.
[19] L. Hanzo, T. H. Liew, B. L. Yeap, "Turbo Coding, Turbo Equialisation and Space-Time Coding for Transmission Over Fading Channels", John Wiley & Sons. 2002.
[20] G. D. Golden, G. J. Foschini, R. A. Valenzuela and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the VBLAST space-time communications architecture", Electronics Letters, Vol. 35, No. 1, Jan. 7, 1999, pp. 14-15.

3

MAXIMUM LIKELIHOOD DETECTION OF SPACE-TIME CODING SYSTEMS

3.1 INTRODUCTION

In this chapter, we consider the mathematical framework for Maximum Likelihood (ML) decoding of space-time coded systems. Matrix computation concepts will be highlighted and used to search for the correct signal sent from the transmitter to the receiver in a ML criterion. ML decoding is the optimum procedure for computing the point $\bar{\boldsymbol{x}}$ in a lattice which is closest to the received symbol, $\boldsymbol{x}$. Nevertheless, this procedure is extremely complex, giving rise to suboptimum algorithms among which we can find the BLAST algorithm for Layered Space-Time coding. Hence, the aim of this chapter is to introduce the Closest Point (CP) decoding algorithm proposed by Agrell [2] and analyze its performance and complexity in order to elucidate the reasons why BLAST algorithm is the preferred choice for decoding Layered Space-Time coded systems. It is important to notice that CP uses all the lattice points in the space and ML decoding uses only the lattice points that belong to the constellation points in use. ML decoding is extremely complex to implement when the number lattice points is high; that is the reason why CP algorithms are commonly used since they can be seen as a suboptimum but a very good approximation to ML decoding with much less computational effort.

The outline of the chapter is the following

- Section 3.2: Introduction to Lattice points.
- Section 3.3: Preliminary of lattice calculations.
- Section 3.4: Agrell algorithm
- Section 3.5: Performance of CP algorithm.
- Section 3.6: Conclusions.

3.2 INTRODUCTION TO LATTICE POINTS

As an introduction, we can say that a lattice code is a finite subset of points of a lattice set (or of a lattice translate) within a bounded region containing the origin, so that the energy of each signal is bounded [1]. The lattice points are used to create the constellations points in the modulation process and attain the Shannon capacity bound.

Decoding a d-dimensional lattice consist basically in finding the lattice point closest to any given vector Z in the Euclidean space R^d. The main idea of this section is to understand

lattices, its meaning and importance in communications systems, and the decoding algorithm.

The main application of lattice points in communications systems is decoding modulated symbols which are corrupted by additive white Gaussian noise (AWGN) noise and multipath. That is, given a d-dimensional lattice A and a point Z in the d-dimensional Euclidean space R^d, we must find the lattice point in A closest to Z and consider this point as the transmitted symbol we used. Before considering CP lattice decoding, let us review the basic lattice definitions.

3.2.1 LATTICES

Let $v_1 \quad v_2 \quad ... \quad v_m$ be m linearly independent vectors of the d-dimensional Euclidean space R^d (where m ≤ d). A lattice is the set of vectors defined as follows

$$\Lambda = \lambda_1 v_1 + \lambda_2 v_2 + ... + \lambda_m v_m ; \lambda_i \in Z \tag{3.1}$$

where λ_i are integers. The set of vectors v is called basis of Λ and is defined as $v_i = (v_{i1}, v_{i2}, ..., v_{id}); i = 1, ..., m$.

Lattice Λ can be written as $\Lambda = \{u = bG : b \in Z^m\}$, where G is the generator matrix defined as

$$G = \begin{bmatrix} v_{11} & ... & v_{1d} \\ ... & & ... \\ v_{m1} & ... & v_{md} \end{bmatrix} \tag{3.2}$$

From this equation we can see that generator matrix G has real entries whose rows are linearly independent over R^d. We can denote m and d as the number of rows and columns of G.

The closest point problem consists in finding for a given received point $x \in \Re^d$, the closest lattice point, $\bar{x}$, such that for any given vector $\bar{X} \in \Lambda$ the distance between point $\bar{x}$ and x is smaller than for any other lattice point $\bar{x}$. Mathematically this is represented as

$$\left\| x - \bar{x} \right\| \leq \left\| x - \bar{X} \right\| \tag{3.3}$$

where $\|\bullet\|$ denotes the Euclidean norm [2]. In channel coding theory, we refer to equation 3.3 as the ML decoding problem. In order to define the closest distance to the lattice point, we must define a concept called sphere packing.

3.2.2 THE PACKING OF SPHERES

The general idea of the packing of *n*-dimensional spheres in mathematics is equivalent, in communications theory, to the design of a finite set of digitally encoded messages that do not waste power and do not cause confusion in transmission, that is, they can be received with low error probability.

There are basically two concepts related to the packing of spheres: the kissing number (the number of spheres that can be arranged around a central sphere) and covering (the least dense arrangement of identical spheres with the property that any point is inside or on the boundary of at least one sphere).

The detailed design and characteristics of packing of spheres are not reviewed here. The interested reader is referred to [3]. The principal idea of using high dimensional sphere packing is the construction of signals to be used on a noisy channel with minimum decoding error probability and high spectral efficiency. In digital communications, we can construct a list of distinct coded symbols that can be transmitted with maximum reliability and minimum power. Each of them can be represented as modulation symbols.

These spheres packing will help us to find the closest lattice point and the corresponding symbol, making possible the minimization of errors. If a lattice is used as a code for the Gaussian channel, then maximum likelihood decoding in demodulation will help to find the closest point.

In order to understand the problem, we will describe in the next section the principal idea of the decoding problem, which means, finding the closest lattice point.

3.2.3 BASICS ON CLOSEST LATTICE POINT DECODING

The decoding problem consists in finding the lattice point closest to the received symbol. A direct and “naive” process would consist in doing an exhaustive search over all the points that form the lattice and select the point with the smallest Euclidean distance from the received point.

The problem of this direct search comes out when lattices are too big, which is usually the case, since the decoding procedure will be very complex and too long. As a result, more efficient search procedures have to be envisaged. One of these algorithms consists in restricting the search to a finite number of lattice points lying within a bounded region. The first chosen region is made large, and whenever a shorter vector is found, the search region is consequently reduced, and so on until only one dimension is explored.

This problem can be stated mathematically as

$$\min_{U\in\Lambda}\|z-u\| = \min_{W\in z-\Lambda}\|w\| \qquad 3.4$$

where $u = xG$ with $x \in Z^d$, $z = \xi G$ with $\xi = (\xi_1,...,\xi_d)$, and $w = \zeta G$ $\zeta = (\zeta_1,...,\zeta_d)$, both vectors are real. The operation described in the equation can be rewritten as $w = \sum_{i=1}^{d} \zeta_i v_i$ where $\zeta_i = \xi_i - x_i$ with *i*=1 to d. From this, we chose the shortest non-zero vector of the translated lattice $z - \Lambda$.

In words, what equation 3.4 indicates is that instead of searching all lattices points closest to the received vectors z, it is possible to translate the lattice by an amount $z - \Lambda$, since it is much easier to search for the lattice point closest to the origin. As a result, a ball or sphere of radius $\sqrt{\bar{X}}$ can be constructed, centered at z and then we can start the search in all lattice points inside this ball. This is the typical approach encountered in several closest point algorithms [1].

In practical applications, the radius of the searching region could be adaptively adjusted according to the noise level. That is, if no lattice point is detected inside the ball, the radius must be increased. On the contrary, if distance of the received point from lattice is small, then the radius can be decreased [4].

3.2.4 BASICS ON MAXIMUM LIKELIHOOD DECODING

In communications theory, it is common practice to assume that the transmitted signal is corrupted with noise components which are independent and identically distributed with zero mean Gaussian random variables.

The information signal *x* is uniformly distributed over a discrete and finite set $K \in \Re^d$ which represent the codebook of modulation constellations. Under that conditions and assuming that we know perfectly ***H*** at the receiver, we can say that the optimum detector could minimize the average error probability $P(k) \cong P\left(\bar{x} \neq x\right)$ which is known as the maximum likelihood (ML) detector and is given by

$$\bar{x} = \arg\min_{x \in K} |y - Hx|^2 \qquad 3.5$$

Notice that *K* is the set of all possible signals that exist in the transmission codebook, and it is possible that not all the closest points from the received point belong to this codebook. As a consequence, the decoding process not only has to detect and decide for the closest point but also it has to take into consideration the allowable lattice point that form the constellation and which are to be taken into account in the searching process. This issue is the principal difference between ML decoding and Closest Point decoding in lattices [5].

Although closest point implies extra work and in some cases, the closest point found is not a constellation point, the time used in the searching process is diminished considerably.

3.3 PRELIMINARY CALCULATIONS OF CLOSEST POINT

Closest point is a suboptimal algorithm with exponential computational complexity. The matrix reductions used in the Agrell algorithm of closest point [3] make the time process faster. Roughly speaking, closest point decoding consists basically in performing changes in the coordinate system used to represent a lattice so as to considerably decrease the computational burden needed to decode the received symbols.

3.3.1 BASIC CONCEPTS

Two lattices are identical if and only if their lattice points are the same. Two generator matrices G_1 and G_2, generate identical lattices $\Lambda(G_1) = \Lambda(G_2)$ if and only if

$$G_1 = G_2 W \tag{3.6}$$

where W is a square matrix with integer entries satisfying the condition $|\det(W)| = 1$.

When a generator matrix G_2 is rotated and reflected, then we have another representation of G_1, which is the transformation of a coordinate system. If G_2 is square and lower triangular, it is said to be a lower triangular representation of G_1, i.e.

$$G_1 = G_2 Q \tag{3.7}$$

Two lattices are congruent or equivalent if one can be obtained from the other trough scaling, rotation and reflection, that is,

$$G_1 = cWG_2 Q \tag{3.8}$$

The constant term c must be real and positive, while W and Q obey the same conditions as in equation 3.6 and 3.7, respectively.

There are several ways lattice basis vectors for the generator matrix can be defined. The process of selecting a good basis for a given lattices is called reduction. The aim is to find a generator matrix representation of a lattice with the smallest basis vectors so that computations can be reduced. Basically there are two types of reductions that are more widely used in practice: the Korkine-Zolotareff (KZ) reduction and the Lenstra-Lenstra-Lovàsz (LLL) algorithm [6].

The LLL reduction is used principally to reduce the time required in the search process. The general idea is to reduce the operations required by some transformations (or matrix operations) in the channel matrix estimation which allow decreasing the search in one less dimension (from an *n*-dimensional to a (*n*-1)-dimensional lattice representation). This implies important reductions in the number of lattice points we search.

The LLL algorithm basically suggests the corresponding reduction criteria with minimum time consumption. This algorithm needs that the generator matrix be given as a lower triangular matrix as below

$$G = \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \begin{bmatrix} v_{11} & 0 & \dots & 0 \\ v_{21} & v_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ v_{n1} & v_{n2} & \dots & v_{nn} \end{bmatrix} \tag{3.9}$$

In order to obtain an LLL reduced matrix, we must consider the following conditions:

- $\|v_1\| \le \frac{2}{\sqrt{3}} \|v_2\|$
- $|v_{k1}| \le \frac{|v_{11}|}{2}$ for k=2,…,n
- Submatrix $\begin{bmatrix} v_{22} & \dots & 0 \\ \dots & \dots & \dots \\ v_{n2} & \dots & v_{nn} \end{bmatrix}$ is LLL reduced.

For a thorough description of the LLL reduction, refer to [6]. As indicated above, the KZ and the LLL reductions are the preferred lattice reduction algorithms. However, the LLL reduction is preferred since, even though the KZ implies a higher lattice reductions, when considering the decoding process as well, the LLL algorithm presents lower processing time.

3.4 AGRELL ALGORITHM

This section presents the Agrell algorithm based on Closest Point decoding [2]. The goal is to simplify the discussion in [2]. It is assumed that transmitted symbols are drawn from a 16-QAM constellation.

3.4.1 GENERAL DESCRIPTION

The goal is to decompose a lattice generator matrix G in the form

$$G = \begin{bmatrix} G^* \\ v_n \end{bmatrix} \tag{3.10}$$

where the top (n-1) rows of matrix G describe a new matrix G* and the new vector v_n is defined as $v_n = v_\perp + v_\parallel$, with $v_\parallel$ in the row space of G* and $v_\perp$ in the null space. When G is a lower triangular matrix, we can apply the QR decomposition in order to make the

decomposition of G into G* and v_n simpler, i.e., we can make $v_{\|} = (v_{n,1},...,v_{n,n-1},0)$ and $v_{\perp} = (0,...,0,v_{n,n})$.

From equation 3.10, we can say that any n-dimensional lattice can be decomposed as

$$\Lambda(G) = \bigcup_{u_n=-\infty}^{\infty} \{c + u_n v_{\|} + u_n v_{\perp} : c \in \Lambda(G^*)\} \quad 3.11$$

We can notice that this equation contains (n-1) dimensional hyper planes that contain these sublattices. These are referred to as (n-1) dimensional layers.

The index u_n denotes the layer to which the lattice point belongs. The vector $v_{\|}$ is the offset by which one sub lattice is translated within its layer, with respect to an adjacent sublattice. The vector $v_{\perp}$ is normal to the layers, and the distance between two adjacent layers is ($\|v_{\perp}\|$). If the generator matrix is lower triangular, then $\|v_{\perp}\| = |v_{nn}|$.

Remember that all search algorithms for an n-dimensional lattice will be described recursively as a finite number of (n-1)-dimensional search operations. Let $x \in \Re^d$ be a vector to be decoded in the lattice. The orthogonal distance from x to the layer with index u_n is given by [3]

$$y_n \stackrel{def}{=} \left| u_n - \bar{u}_n \right| \cdot \|v_{\perp}\| \quad 3.12$$

where

$$\bar{u}_n \stackrel{def}{=} \frac{x v_{\perp}^t}{\|v_{\perp}\|^2} \quad 3.13$$

Let us denote $\bar{x}$ as the closest lattice point to x, and suppose that an upper bound ρ_n on $\left\| \bar{x} - x \right\|$ is known. Then, in order to ensure that the x point will be found, we must consider a finite number of layers

$$u_n = \left\lceil \bar{u}_n - \frac{\rho_n}{\|v_{\perp}\|} \right\rceil ... \left\lfloor \bar{u}_n - \frac{\rho_n}{\|v_{\perp}\|} \right\rfloor \quad 3.14$$

The layer with $u_n = q\left(\bar{u}_n\right)$ has the shortest orthogonal distance to x, where function $q(z)$ denotes the closest integer point to $z \in \Re$.

The search method to identify the closest lattice point is defined as the Schnorr-Euchner strategy [2]. From both we assume that $\bar{u}_n \le q\left(\bar{u}_n\right)$. Then the sequence orders the layers according to non-decreasing distance from *x*. A trivial counterpart holds when the distance we propose is bigger than previous calculations.

The advantages of the layers are that the volume of a layer decreases with increasing y_n, the chance of finding the correct layer soon is maximized. Other advantage is that the search can safely be terminated as soon as y_n exceeds the distance to the best lattice point found so far, that is

$$u_n = q\left(\bar{u}_n\right), q\left(\bar{u}_n\right) - 1, q\left(\bar{u}_n\right) - 2, \ldots \qquad 3.15$$

which is the sequence order of the layers according to no decreasing distances from x. The very first lattice point found will be referred as the Babai point [5] and corresponds to the case where the radius of the searching ball is infinite. Furthermore, since the ordering in equation 3.15 does not dependent on ρ_n, no initial bound is needed. Instead, the bound will be automatically updated during the search through the Babai point.

3.4.2 DESCRIPTION OF THE DECODE FUNCTION

The Agrell algorithm needs a last processing step referred to as DECODE, which finds the closest point in the lattice. The DECODE function is first explained since it is the core of the algorithm. Indeed, a preprocessing step in the Agrell algorithm is used only to modify the generator matrix so as to alleviate the DECODE function, rendering much faster its decoding process.

The DECODE process uses the H and $\bar{u}$ variables as input and output parameters respectively, we define $G = H^{-1}$ and $\bar{x} = \bar{u}G$.

In the algorithm (figure 3.1), *k* is the dimension of the sub layer structure we are investigating. There are three possible cases while searching for the closest lattice point. In Case A, each time the algorithm finds a *k*-dimensional layer, the distance is less than the currently smallest distance. This layer is expanded into (*k*-1)-dimensional sublayers.

In case C, as soon as the distance to the examined layer is greater than the lowest distance, the algorithm moves up one step in the hierarchy of layer. When Case B happens, the algorithm has successfully moved down all the way to the zero-dimensional layer (that is, a lattice point) without exceeding the lowest distance. Then, the lattice point is stored as a potential output point, the lowest distance is updated, and the algorithm moves back up again, without restarting. The conventions are:

- $u = (u_1, u_2, ..., u_n)$
- $c_k = (c_{k1}, c_{k2}, ..., c_{kk})$ for k=1,…,n
- $H = \begin{bmatrix} h_{11} & 0 & ... & 0 \\ h_{21} & h_{22} & ... & 0 \\ ... & ... & ... & ... \\ h_{n1} & h_{n2} & ... & h_{nn} \end{bmatrix}$
- sgn $= (z)$ returns -1 if $z < 0$ and 1otherwise.

<table>
<tr><th colspan="2">DECODE(H,x)</th></tr>
<tr><td colspan="2">Input: an NxN lower triangular matrix H with positive diagonal elements, and an N-dimensional vector $x \in \Re$ to decode in the lattice $\Lambda\left(H^{-1}\right)$.</td></tr>
<tr><td colspan="2">Output: an N-dimensional vector $\bar{u} \in Z^n$ such that $\bar{u} H^{-1}$ is a lattice point that is closest to x.</td></tr>
<tr><td>n:= The size of H /* dimension */</td><td>} else { CASE B</td></tr>
<tr><td>bestdist := ∞ /* current distance record */</td><td>$\bar{u} := u$ /* best lattice point so far */</td></tr>
<tr><td>$k := n$ /* dimension of examined layer */</td><td>$bestdist := newdist$ /* update record */</td></tr>
<tr><td>$dist_k := 0$ /* Distance to examined layer */</td><td>$k := k + 1$ /* move up */</td></tr>
<tr><td>$e_k := xH$ /* Used to compute point */</td><td>$u_k := u_k + step_k$ /* next layer */</td></tr>
<tr><td>$u_k := \lceil c_{kk} \rfloor$ /* examined lattice point */</td><td>$y := \frac{c_{kk} - u_k}{h_{kk}}$</td></tr>
<tr><td>$y := \frac{c_{kk} - u_k}{h_{kk}}$ /* see 3.12 */</td><td>$step_k := -step_k - \text{sgn}(step_k)$</td></tr>
<tr><td>$step_k := \text{sgn}(y)$ /* offset to next layer in 3.15 */</td><td>}</td></tr>
<tr><td>(loop)</td><td>} else { CASE C</td></tr>
<tr><td>$newdist := dist_k + y^2$</td><td></td></tr>
<tr><td>If $newdist < bestdist$ then {</td><td>else{</td></tr>
<tr><td>If $k \neq 1$ then { CASE A</td><td>$k := k + 1$ /* move up */</td></tr>
<tr><td>$c_{k-1,i} := c_{k,i} - y h_{k,i}$ for i=1,…,k-1</td><td>$u_k := u_k + step_k$ /* next layer */</td></tr>
<tr><td>$k := k - 1$ /* move down */</td><td>$y := \frac{c_{kk} - u_k}{h_{kk}}$</td></tr>
<tr><td>$dist_k := newdist$</td><td>$step_k := -step_k - \text{sgn}(step_k)$</td></tr>
<tr><td>$u_k := \lceil c_{kk} \rfloor$ /* closest layer */</td><td>}</td></tr>
<tr><td>$y := \frac{c_{kk} - u_k}{h_{kk}}$</td><td>}</td></tr>
<tr><td>$step_k := \text{sgn}(y)$</td><td>Goto {loop}</td></tr>
</table>

Figure 3.1: DECODE algorithm.

As stated above, this function is the most important part in the search algorithm. Next, we will present the other part of the Agrell closest point algorithm.

3.4.3 DESCRIPTION OF AGRELL ALGORITHM

The DECODE algorithm described above needs the generator matrix to be presented in lower triangular matrix form, with positive diagonal elements. The closest point algorithm computes the received vector as fast as possible. However, the algorithm works in a different way and decoding speed depends on a pre-processing of the generator matrix.

If we assume that generator matrix G and input vector *x* are given, we can make linear integer row operations in order to transform G into another matrix (G_2), which generates an identical lattice. This transformation has the purpose of rendering the DECODE algorithm faster. Next we rotate and reflect G_2 into a lower triangular form G_3, so that

$$\Lambda(G_3) \cong \Lambda(G_2) = \Lambda(G) \qquad 3.16$$

We must remark that vector *x* has to be rotated and reflected in same way as G_2, so that we construct a vector x_3, in same relation as in equation 3.16. Previous operations simulate a change of the coordinate system. Now, we have a form that makes easier and faster the DECODE algorithm, which is responsible of finding the closest lattice point x_3 in this "new" coordinate system. Before we find the point, we must reverse the operations of rotation and reflection that produced $\bar{x}$, the lattice point closest to *x* in $\Lambda(G)$. Following the steps, we have the complete algorithm shown in figure 3.2.

The explanation of this "new" coordinated system is as follows:

a) Is a basis reduction. This step is optional: it is possible to select W as a identity matrix, which basically gives as a result no reduction at all. The algorithm works better with a reduction of dimensional layers, the speed and the numerical stability can be improved significantly by an appropriate reduction.
b) Implies rotation and reflection of G_2 into a lower triangular form. The method we used is QR decomposition (refer to appendix A for a description of the QR decomposition). Basically, given an arbitrary MxN matrix *M*, its QR decomposition is a factorization of M of the form $M = QR$, where R is an NxN upper triangular matrix, and *Q* is an MxN orthonormal matrix, that satisfies Gram Matrix [3]. The transformation made above implies that the matrix can be seen as a coordinate system with square and lower triangular matrices.
c) The DECODE algorithm requires all diagonal elements of G_3 to be positive. QR decomposition sometimes does not make it automatically; if one element is negative, all the column is multiply by -1.
d) In the remaining steps, input vectors are processed. The coordinate system is inverted in order to get the point in the old coordinate systems.

An n-dimensional lattice consists of parallel (n-1)-dimensional sub lattices, translated and stacked in top of each other. The decomposition is controlled by the reduction method. The principal properties of that decomposition for a lattice are:

- The (n-1)-dimensional layers should be as far apart as possible, which minimizes the necessary number of layers and only the layers within a certain distance range need to be scanned. We can suppose that the spacing between (n-1)-dimensional layers is much larger than any other k-dimensional layer spacing in the lattice. Then the closest point will lie in the closest (n-1)-dimensional layer, and the dimensionality of the problem is essentially reduced by one.
- The zero-dimensional layer (i.e., the lattice point) should be densely spaced as possible from the one-dimensional layer (line). The reason is that the denser they are, the higher is the probability that the closest point will belong to the lattice line. If the distance is smaller than the other distances, then the closest point will always be uncertain in the closest line, so the problem will now be in only one dimension.

These two observations are highlighted because recursive methods can be applied, i.e., it makes the problem with high-dimensionality large, while low-dimensional should be small. So, the algorithm presented will have high priority in: a) sequentially maximizing the distance between *k*-dimensional layers, starting with *k=n-1*, and b) minimizing the same distances, starting with *k=0*.

CLOSEST POINT (G, x)	
Input: an NxM generator matrix G, and an M-element vector $x \in \Re^m$ to decode in $\Lambda(G)$	
Output: a lattice pint $\bar{x} \in \Lambda(G)$ that is closest to *x*.	
Let $G_2 := WG$; where W is an NxN matrix with integer entries and determinant ± 1.	Let $x_3 := xQ^T$
Compute an NxM orthonormal matrix Q such that $G_2 := G_3 Q$; where G_3 is an NxN lower triangular matrix with positive diagonal elements.	Let $\bar{u}_3 := DECODE(H_3, x_3)$
Let $H_3 := G_3^{-1}$	Return $\bar{x} := \bar{u}_3 G_2$

Figure 3.2: Closest Point algorithm.

In order to explain the last points, we must see that they are very similar. Observe in figure 3.2, that we can reduce the algorithm by choosing the numbers v_{kk} in many ways for a given lattice, but their product will be invariant, which means that the product will always be the volume of the Voronoi region [4]. The smallest value of v_{11} can be obtained using the LLL procedure, which will be automatically the shorter vector in the lattice. However, we must know that the LLL method is only an approximated (making faster) solution because of its inherent sorting mechanism.

3.4.4 CONVERSION FROM CLOSEST POINT TO CONSTELLATION POINT

When we have the closest point, this point might not correspond to a codebook given by the modulation constellation we are using. As a result we have to search and find the closest point from the lattice point into the modulation codebook. The easier form to do this is by

knowing the modulation constellation. Then, we search and compare the closest point found with all constellation points.

3.5 PERFORMANCE OF AGRELL ALGORITHM

In this section we will analyze the performance of the Agrell decoding algorithm. This section will serve as a reference for assessing other suboptimal decoding algorithm such as the VBLAST algorithm. In this section, it will be shown that the computational complexity of the Agrell algorithm will make difficult a hardware implementation. The analysis will be done in two parts. First part is dedicated to analyze the computational complexity and storage requirements. The results obtained show that neither material complexity nor error probability will be affected if we use the LLL reduction method.

Simulations were carried out on a Toshiba Satellite Laptop computer with Intel Pentium 4 processor, 2.66MHz, 256 MB in Ram and Operator System Fedora Core 2 (Linux 11). Msim simulator was used [7].

3.5.1 PARAMETER USED IN SIMULATION.

During simulation we can identify different important parameters. Among these, the block sizes, the number of transmit and receive antennas, and the use of the LLL reduction are the most important. In addition, simulations were done until 2000 block errors occur. Recall that a block error occurs when at least one bit in the block is in error.

The LLL reduction method can be useful for the simulation performance, but first, we have to choose the correct number of bits per block. If the number of bits per block is too small, LLL reduction will make simulations slow; therefore, the first analysis we will be made so as to find the correct number of bits per block. We will limit our search only for block of length L equal to 1,5, 10 and 100 symbols per block.

3.5.2 COMPLEXITY ANALYSIS

3.5.2.1 Determination of Block Size

We simulated all the possible antenna combinations. From results, we can see that block error probability (BLER) is not affected significantly when the number of symbols in a block increases or decreases. However, hardware complexity is affected when the number of symbols per block is varied (Figure 3.3a, 3.3b).

From these figures we can observe that when the block size is one, complexity is the highest. On the other hand, when the block size is 100, complexity decreases in approximately ten arithmetic operations per decoded bit. This indicates that the block size is very important in order to make a hardware implementation feasible. If we compare the

Agrell algorithm with and without LLL reduction we can see a small difference between them, when the block size is 100, the number of operations with or without LLL reduction is almost the same, while for a block size of 10 increase only one operation. Hence, we will use a block size of 10 in order to keep the simulation time acceptable.

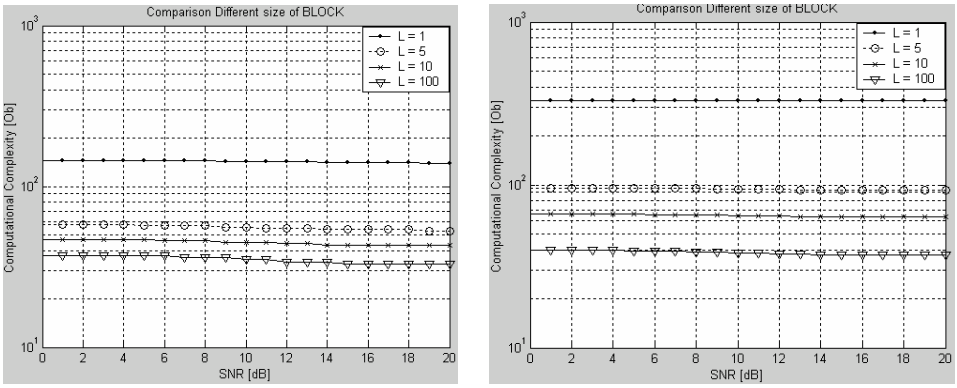


Figure 3.3: Comparison Complexity vs. SNR with size of block. a) Without LLL Reduction and b) With LLL Reduction.

3.5.2.2 Agrell Algorithm Without LLL Reduction.

In this section, performance without LLL reduction is presented. In all future analysis, the number of operations required by each algorithm for estimating one bit of information is called Ob. In this analysis, every arithmetic operation is referenced to addition operations. This way, a multiplication and division operation is considered to be equivalent to two additions, and square root operation is equivalent to four addition operations. Once again, simulations were carried out with L = 10 symbols per block, 16-QAM constellations and they were run until 2000 block errors occurred.

Figure 3.4 presents the total computational complexities with two transmit antennas and 2, 4, 6 and 8 receive antennas. We can see that there is an increase of approximately twenty operations every time the receive array increases; for instance, in the 2x2 system, the number of operations (and the number of memory locations) equals 50, while the number of operations for the 2x4, 2x6 and 2x8 arrays is 66, 82 and 101 respectively.

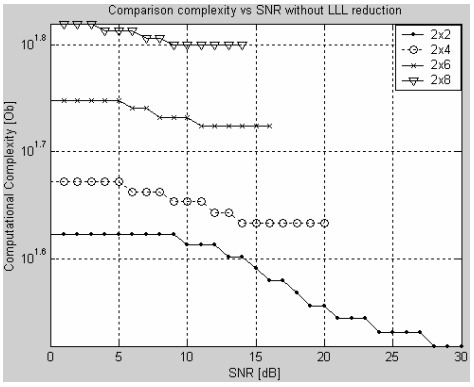


Figure 3.4: O_b vs. SNR_{avg} M = 2; N = 2,4,6,8.

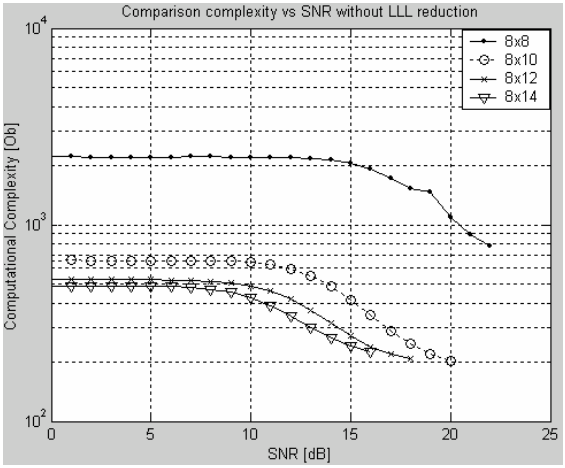


Figure 3.5: O_b vs. SNR_{avg} M = 8; N = 8,10,12,14.

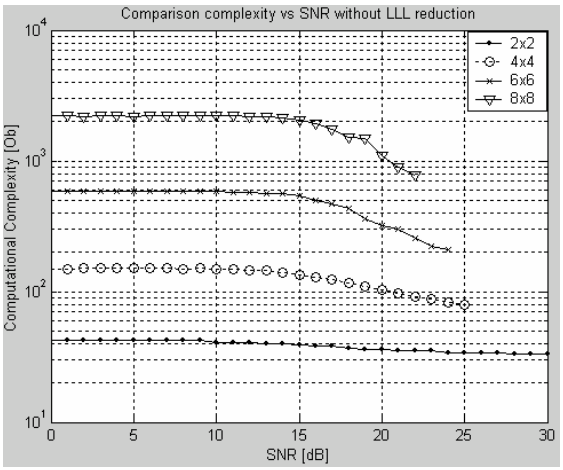


Figure 3.6: O_b vs. SNR_{avg} 2x2, 4x4, 6x6, 8x8.

The closest point algorithm depends on the SNR. When we have a noisy channel, the number of operations increases considerably, and the reason is very simple; the search areas where the possible lattice point could be are larger. This is the reason why complexity in the previous figures are not constant.

When the number of transmit antennas are larger, as in figure 3.5, the complexity increases significantly with the number of transmit antennas. Finally, systems with square arrays (i.e., equal number of transmit and receive antennas) increase considerably the number of operations and storage requirements. This is shown in figure 3.6.

3.5.2.3 Agrell Algorithm with LLL Reduction.

We now turn our attention to the LLL reduction method in order to determine if hardware complexity is affected. As indicated before, BLER is not modified, only the processing time is. From figures 3.7 through 3.9 we can see that computational complexity is not considerably increased with LLL reduction as compared to figures 3.4 through 3.6. However, simulation time is greatly reduced with this algorithm.

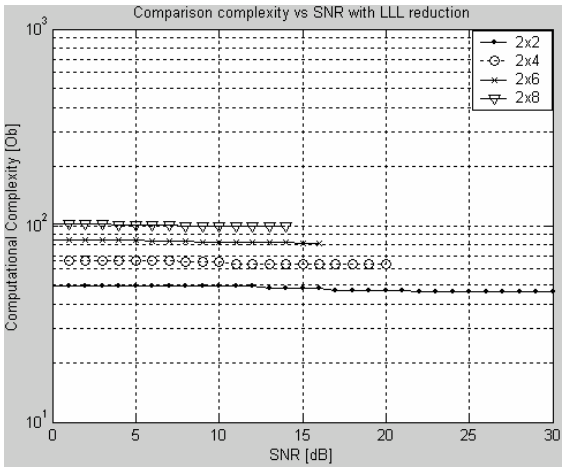


Figure 3.7: O_b vs. SNR_{avg} M = 2; N = 2,4,6,8.

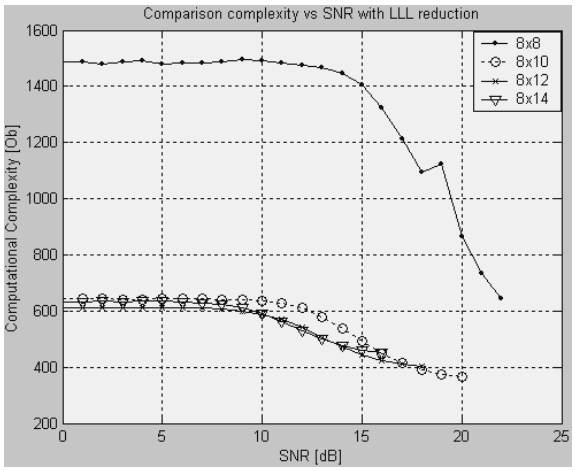


Figure 3.8: O_b vs. SNR_{avg} M = 8; N = 8,10,12,14.

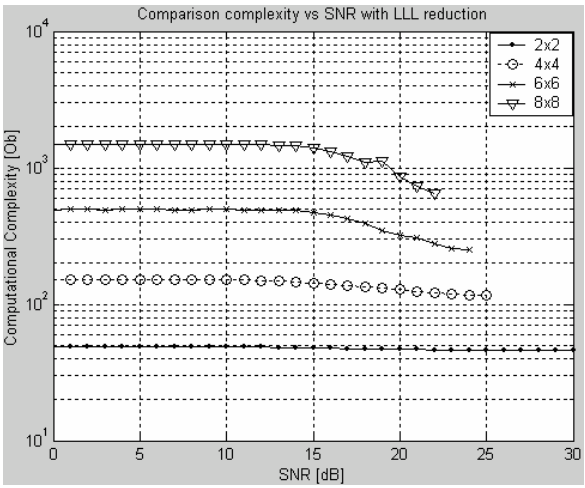


Figure 3.9: O_b vs. SNR_{avg} 2x2, 4x4, 6x6, 8x8.

As a conclusion, we can determine an approximation of the computational requirements as follows:

$$Add = 2MN(12N + 2L + 7) + 4M^2(24M + 3L - 22) + 2M(5L + 9) + N(11N + 2L + 3) + 5L - 7$$
$$Mult = MN(24N + L - 1) + 2N(6N^2 + 8N - 6) + M^2(40M + 6L - 44) - M(14 - L) + L$$
$$Div = M(12N + L) + 2N(2N^2 + 3N + 4) + 3L - 1$$
$$Sqrt = 3N + 2$$
$$Mem = 8MN(7N + L + 2) + 4N(19N + L + 1) + 4M^2(42M + 9L + 5) + 2M(4L - 13) + 10(L - 1)$$

Where Add represent the addition operation, Mult is multiplication, Div is division, Sqrt is square root and Mem is the memory required. L is the number of bits per block. These operations are only for one block, if we want the number of operations per bit, we have to divide the results by the number of bits per block (a block is defined as the number of bits per symbol – 4 bits/symbol –, multiplied by the number of bits per block, multiplied by the number of transmit antennas).

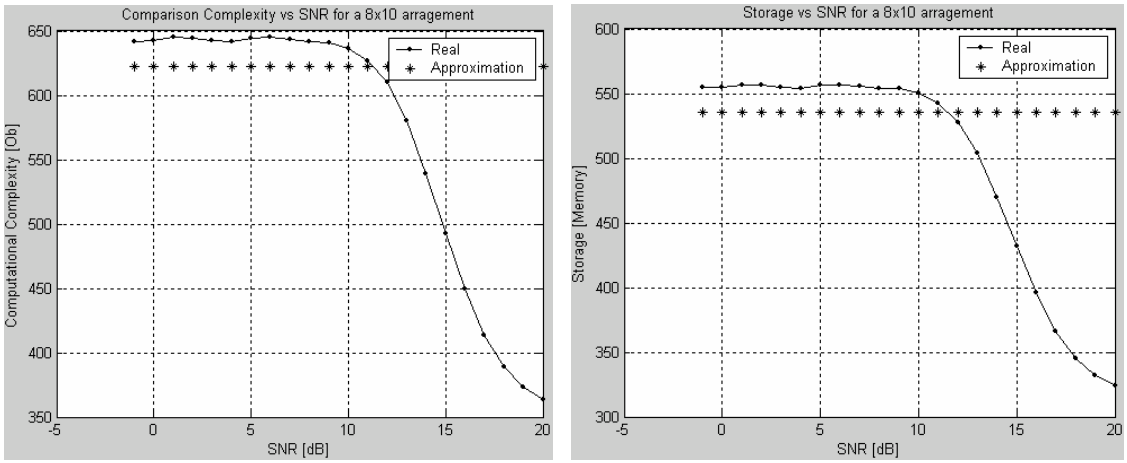


Figure 3.10: a) Computational Complexity Simulation vs. SNRavg, b) Storage requirements vs SNRavg Approximation for a 8x10 array.

Figure 3.10 shows a comparison between the approximation formula and the number of calculations obtained by simulation for a 8x10 array of antennas. As we can see, this approximation gives a good idea of how many calculations we need and the number storage requirements when channel is very noisy. Calculations and storage for the closest point are not very good approximation when noise in channel is greater than 13dB. The reason is very simple, the searching area is diminished.

3.5.3 PERFORMANCE ANALYSIS

Figures 3.11, 3.12 and 3.13, give the block error probability performance for different MIMO systems decoded with the Agrell algorithm. As we can see, BLER is reduced as we increase the number of receive antennas. In order to illustrate this, first we take two transmit antennas and increase the number of receiver antennas by two (see fig. 3.11). Results show that from two to four receive antennas, a gain of 10dB is achieved. For the 2x6 array system, we have a 14dB in gain, while for a 2x8 arrangement, the coding gain is 14dB.

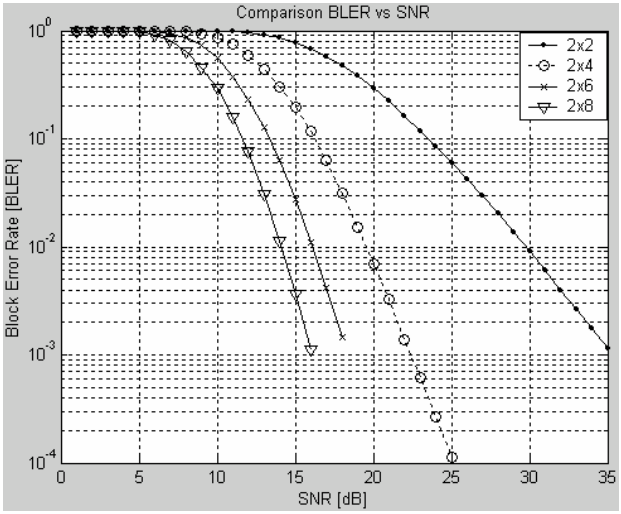


Figure 3.11: BLER vs. SNR_{avg} M = 2; N = 2,4,6,8.

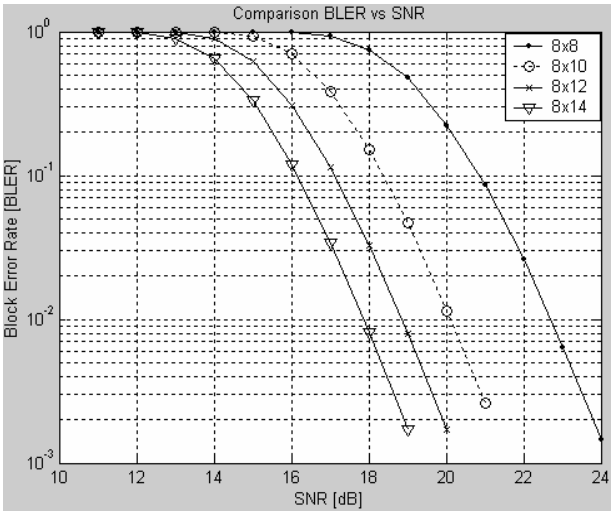


Figure 3.12: BLER vs. SNR_{avg} M = 8; N = 8,10,12,14.

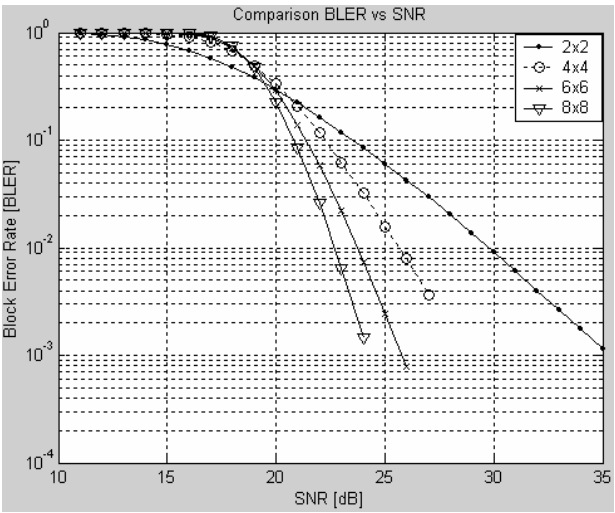


Figure 3.13: BLER vs. SNR_{avg} 2x2, 4x4, 6x6, 8x8.

Figure 3.12 presents the BLER performance for systems with M=8 transmit antennas and N=8,10, 12 and 14 receive antennas. We can see that for a BLER of 10^{-2}, gains of 3 dB, 4 dB and 5dB are obtained as the number of receive antennas is increased. From these figures we can see that the number of receive antennas is the most important parameter.

Last comparison shows the performance of square MIMO systems. We can see that for high SNR, BLER is reduced as the array increases.

3.6 CONCLUSIONS

In this chapter near-optimal ML decoding based on Agrell Algorithm (closest point algorithm) with and without LLL reduction has been presented. The performance and complexity have been analyzed and compared.

We already determined the correct number of bits used in future analysis. It was shown that closest point algorithm has a large computational complexity which makes its hardware implementation impossible. This obligates the designer to search a new decoding way where efficiencies will not be affected by the SNR and the number of operations required. BLAST system is presented in next chapter as an alternative.

Next chapter makes an introduction to the BLAST system and all of its variants in order to find the one that makes possible an implementation with minimal hardware requirements.

REFERENCES:

[1] E. Viterbo, E. Biglieri, "A universal Decoding Algorithm for Lattice Codes", Quatorzieme Colloque Gretsi – Juan Les Pins. Du 13 AU 16. Septembre 1993.
[2] E. Agrell, T. Eriksson, A. Vardy, K. Zeger, "Closest Point Search in Lattice". IEEE Trans. On Information Theory, Vol. 48, No. 8, August 2002, pp. 2201 – 2214.
[3] J. H. Conway, N. J. A. Sloane, "The packing of Spheres: Lattice and Groups". New York: Springer-Verlag, 1998.
[4] G. D. Forney, A. Vardy, "Generalized Minimun-Distance Decoding of Euclidean-Space Codes and Lattices". IEEE Trans. On Information Theory, Vol. 42, No. 6, November 1996.
[5] M. O. Damen, H. E. Gamal, G. Caire, "On Maximum-Likelihood Detection and the Search for the Closest Lattice Point". IEEE Trans. On Information Theory, Vol. 49, No. 10, October 2003.
[6] A. Wang, "Lattice Reduction and LLL Algorithm", IMA Summer Program, University of Notre Dame. June 8 2004. www.nd.edu/~cam/ima/CourseMaterial/Wang1.pdf
[7] M. Bazdresch and J. Rodriguez-Guisantes: "The msim simulator". [Online], Available: http://www.comelec.enst.fr/~rodriguez/msim/index.html
[8] G. Golub, C. Van Load, "Matrix Computations", The Johns Hopkins University Press. 1996.

4

LAYERED SPACE-TIME CODING SYSTEMS

4.1 INTRODUCTION

In this chapter, a brief introduction to BLAST systems is done. The main goal of these systems is to achieve the best tradeoff complexity-performance [9]. Space-Time Trellis algorithms have the potential drawback that complexity grows exponentially with the number of bits per symbol. This limits the achievable data rates. In [1], Foschini proposed a layered Space-Time (LST) architecture that could attain the lowest tight bound on the MIMO channel capacity.

The principal and most important difference between the BLAST algorithm with respect to others (STTB or STBC) is that BLAST allows processing of multidimensional signals in the space domain by one-dimensional processing steps, where one-dimensional refers to only one dimension in the space. The method gives us a powerful signal processing technique at the receiver and a conventional one-dimension channel code.

In the original architecture, the information stream is divided into M transmit antennas and transmitted simultaneously, in the same frequency band.

The receiver uses the same number of received antennas and detects the M transmitted symbols.

The process used in the separation of symbols involves a combination of interference suppression and interference cancellation. This way signals are decoded (using simple decoding algorithm), leading to much lower complexity compared to maximum likelihood decoding. The most important characteristic is that complexity grows linearly bandwidth.

This chapter presents an introduction to the BLAST systems. Then we review the received processing techniques used to decouple and detect the LST signals. These processing techniques are based on the Zero Forcing (ZF) and Minimum Mean Square Error (MMSE) interference suppression methods [12][13][16].

Topics are considered as follows

- Section 4.2: LST Transmitters
- Section 4.3: LST Receivers
- Section 4.4: Comparison between LST Architectures
- Section 4.5: Conclusions.

4.2 LST TRANSMITTERS

Four types of layered space-time architectures exist, and those depends on whether error control coding is used or not and the way the modulated symbols are assigned to transmit antennas. There are two kinds of layered architectures, with and without coding. In next sections, we will describe both. We must remark that coding structures will be considered in chapter six in order to improve the overall performance.

4.2.1 HORIZONTAL LAYERED SPACE-TIME (HLST)

The Horizontal Layered Space-Time is a coded form of transmitting information using multiple transmit and receive antennas [3]. As illustrated in figure 4.1, the process consists in separately encoding the signals transmitted by each antenna. The main drawback this technique presents is the difficulty in canceling interference. The process can be described as the sequence being first encoded by a channel code and subsequently demultiplexed into M sub-streams (figure 4.1a). Each substream is modulated, interleaved and assigned to a transmit antenna. Modulator output symbols are denoted by x_t^i, where *i* represents the layer number (the number of transmit antenna) and the *t* is the time interval. The transmission matrix at the modulator outputs is given by equation 4.1 for a system consisting of three transmit antennas

$$X = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & x_4^1 \ldots \\ x_1^2 & x_2^2 & x_3^2 & x_4^2 \ldots \\ x_1^3 & x_2^3 & x_3^3 & x_4^3 \ldots \end{bmatrix} \qquad 4.1$$

From this matrix we can observe that sequence $x_1^1 \quad x_2^1 \quad x_3^1 \quad x_4^1 \ldots$ is transmitted from antenna 1, second row in the matrix represents the sequence transmitted from antenna 2, and the third row from antenna 3. We can found two main types of architectures for HLST, one with the encoder before the demultiplexer, and the other with M encoders after the demultiplexer, one for each antenna (see figure 4.2).

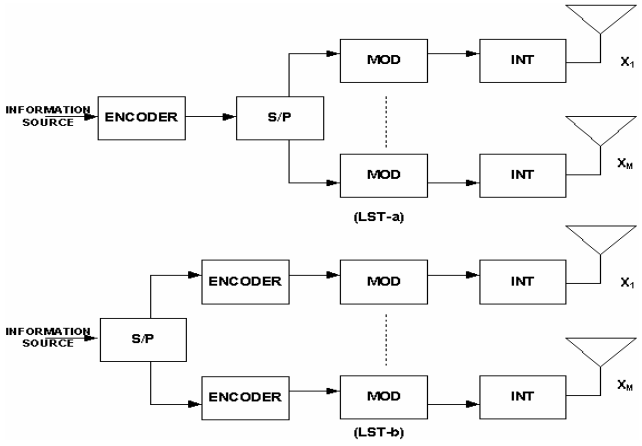


Figure 4.1: A HLST Architecture.

4.2.2 DIAGONAL LAYERED SPACE-TIME (DLST)

This is the original version of BLAST architecture. This structure used a cyclic association of data streams, called layers, thereby producing an "averaged" channel which is the same for all layers. DLST permits that a modulated codeword of each encoder is distributed among the M transmit antennas along the diagonal of the transmission array [1][10][11].

$$\begin{bmatrix} x_1^1 & x_2^1 & x_3^1 & x_4^1 & x_5^1 \ldots \\ 0 & x_1^2 & x_2^2 & x_3^2 & x_4^2 \ldots \\ 0 & 0 & x_1^3 & x_2^3 & x_3^3 \ldots \end{bmatrix} \Rightarrow \begin{bmatrix} x_1^1 & x_1^2 & x_1^3 & x_4^1 & x_4^2 \ldots \\ 0 & x_2^1 & x_2^2 & x_2^3 & x_5^1 \ldots \\ 0 & 0 & x_3^1 & x_3^2 & x_3^3 \ldots \end{bmatrix} \quad 4.2$$

In a three transmit antenna system, we can form the transmit matrix as indicated in equation 4.2, where the *i*-th row is delayed (*i*-1) time units, so that the first entries are always zero from antenna two to M on a diagonal fashion in $\boldsymbol{X}$. This way, the first diagonal is transmitted from the first antenna, the second diagonal from the second antenna and so on as indicated in the right hand side of equation 4.2. Hence the codeword symbols of each decoder are transmitted over different antennas. This operation can be represented by the introduction of a spatial interleaver (SI) after the modulators (Figure 4.2). The spatial operation for the DLST can be represented by equation 4.2.

The rows of the matrix in the right side are obtained by concatenating the corresponding diagonals of the matrix on the left hand of the equation. The first row of this matrix is transmitted from the fist antenna, and so on.

The diagonal layering introduces spatial diversity and a better performance than HLST. It is important to note that DLST has a loss in its spectral efficiency, since a portion of the transmit matrix is padded with zeros [4][5][6].

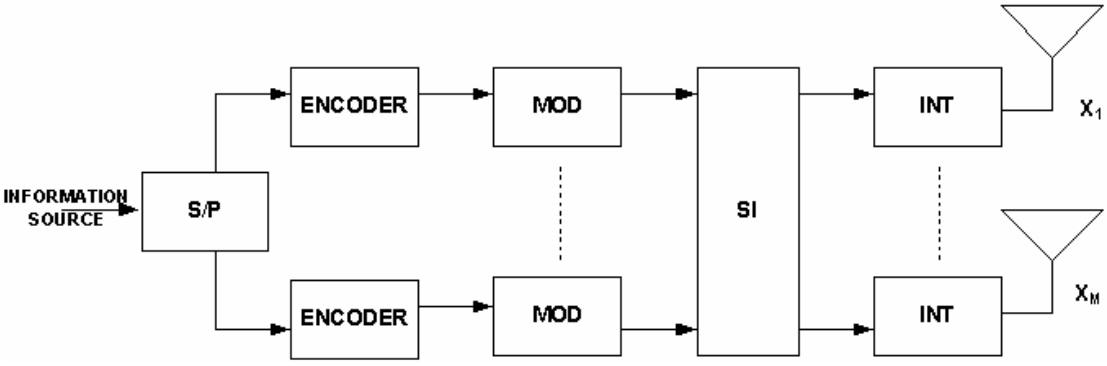


Figure 4.2: A DLST Architecture.

4.2.3 VERTICAL LAYERED SPACE-TIME (VLST)

This architecture was presented by Golden and Foschini in 1999 [2]. This is an uncoded LST structure and is illustrated in figure 4.1. The input information sequence is

demultiplexed into M sub-streams, each of them is modulated by an M-ary level modulation scheme and transmitted from the transmit antennas to the receive antennas. The signal processing for detecting the signals is made over the individual sub-streams (which are called layers). The modulated symbols are arranged into a transmission matrix, denoted by **X**, which consists of M rows of L columns, where L is the transmission block length. The column is the modulated symbols. At a given time *t*, the transmitter sends the column from the transmission matrix, one symbol from each antenna. So, the transmission process could be combined with conventional block or convolutional one-dimensional coding in order to improve the performance of the system.

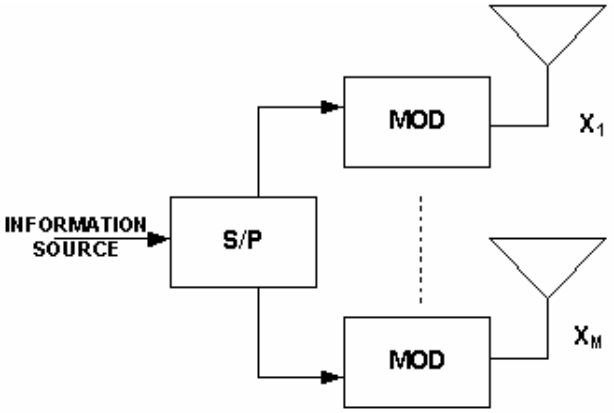


Figure 4.3: A VLST Architecture.

The general description of the transmission matrix is shown in equation 4.3, where the super index is the number of transmit antenna and the sub index is the number of transmit element. The input is describe by the vector $x = [x_1 \quad x_2 \quad ... \quad x_9]$.

$$X = \begin{bmatrix} x_1^1 & x_4^1 & x_7^1 \\ x_2^2 & x_5^2 & x_8^2 \\ x_3^3 & x_6^3 & x_9^3 \end{bmatrix} \qquad 4.3$$

Now let us focus on the signal processing steps for receiving LST coded signals.

4.3 RECEPTION IN LST SCHEMES

The main challenge in LST systems is the detection and decoding problem. For any of the types reviewed above, the decoding processing is different only in the transmit matrix arrangement. This means that the signal processing for reception is the same. In order to simplify the analysis, we will consider horizontal layering with binary channel codes and BPSK modulation.

The transmit diversity always introduce spatial interference. As we saw in chapter two, the transmitted signals from various antennas propagate over independently scattered paths and interfere with each other upon reception at the receiver. The interference can be represented as

$$r_t = Hx_t + n_t \qquad 4.4$$

where r_t is an N-component column matrix representing the received signals across the N receiver antennas, x_t is the t-th element in the transmission matrix X and n_t is an N-component column matrix of the additive white Gaussian noise (AWGN) signals from the receive antennas, with variance σ^2. In order to simplify the notation, we are going to eliminate the sub index.

An uncoded LST architecture can be received and detected with ML multiuser detector [17], which computes ML statistics. The complexity of this detector is exponential in the number of the transmit antennas. For coded LST schemes, the optimum receiver performs joint detection and decoding obtaining the layered space-time coded and the channel code. The complexity of the receiver is an exponential function of the product of the number of antennas and the code memory order.

It is important to understand that the exponential increase in implementation complexity may make the optimal receiver impractical even for a small number of antennas. In [2], it was shown that the best tradeoff between complexity and performance was presented by VLST.

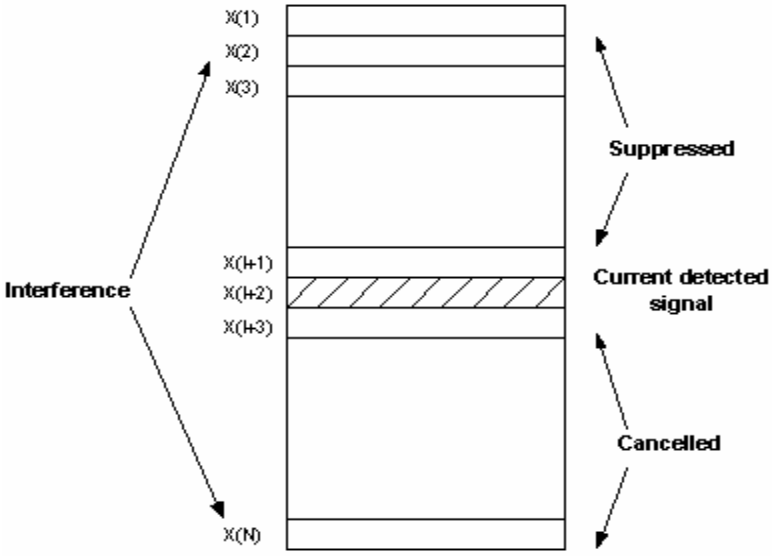


Figure 4.4: VLST detection based on combined interference suppression and successive cancellation.

VLST is based on a combination of interference suppression and cancellation. The basic idea is that each transmitted sub-stream is considered in turn to be the desired symbol and the remainder substreams are treated as interferers. These interferers are suppressed by zero forcing (ZF) approach [2][14][15]. The algorithm produces ZF-based decision statistics for a desired sub stream from the receiver vector r (which contains the transmitted and the multipath signals with interference). Subsequently, from top to bottom, a decision on the desired sub-stream is made from the decision statistic and its interference contribution is regenerated and subtracted out from the receiver vector r. This vector then, contains a lower level of interference and this will increase the probability of making the best detection.

The operation is illustrated in figure 4.4, where the first detected sub-stream is N. The detected symbol is subtracted from all other layers. The operation is repeated for the lower layers, finishing with layer 1. Assuming that all symbols at previous layers have been detected correctly, they will be free from interference.

VBLAST systems always use soft decision statistics at the detector in each layer. When we have coded LST schemes, the decision statistics are passed to the channel decoder, which makes the hard decision on the transmitted symbol in this sub-stream. The hard symbol estimated is used to reconstruct the interference from this sub-stream, which is then fed back to cancel its contribution while decoding the next sub-layer.

We must remark that ZF is only possible when the number of transmit antennas is equal or smaller that the number of receiver antennas. If the ZF is used in removing interference and if N received antennas are available, it is possible to remove interference with diversity order of d_0.

$$d_0 = N - n_i \tag{4.5}$$

where n_i is the number of transmit layers plus one. So, if the interference suppression starts at layer M, then, interference is needed to be suppressed until layer (M-1). If we say that the number of transmit and receive antennas are the same, then the diversity order is one.

From the original BLAST we consider ZF and MMSE (maximum mean square error) detectors, which with a single architecture makes possible the best tradeoff between computational complexity and performance, because the complexity grows linearly with the number of transmit antennas and transmission rate [8].

In the next section, the decoding process with MMSE based on QR decomposition is presented.

4.3.1 QR DECOMPOSITION INTERFERENCE SUPPRESSION COMBINED WITH INTERFERENCE CANCELLATION

A channel matrix MxN, where M≤N, can be decomposed as

$$H = U_R R \tag{4.6}$$

where U_R is a MxN unitary matrix and R is an MxM upper triangular matrix of the form

$$R = \begin{bmatrix} (\mathrm{R}_{1,1})_t & (\mathrm{R}_{1,2})_t & .... & (\mathrm{R}_{1,M})_t \\ 0 & (\mathrm{R}_{2,2})_t & .... & (\mathrm{R}_{2,M})_t \\ .... & .... & .... & .... \\ 0 & 0 & .... & (\mathrm{R}_{M,M})_t \end{bmatrix} \tag{4.7}$$

The decomposition of matrix H in the form 4.6 is called QR factorization. BLAST systems use this factorization in order to get less complexity in its calculations and signal detections. QR Decomposition introduces an M-component matrix y obtained by multiplying from the left the receive vector r, given by equation 4.4, by

$$y = U_R^T r \text{ or } y = U_R^T Hx + U_R^T n \tag{4.8}$$

Substituting the QR decomposition of H from 4.6 into 4.8, we get

$$y = Rx + n' \tag{4.9}$$

where $n' = U_R^T n$ is an M-component column matrix of AWGN noise signals. Since R is an upper triangular matrix, the i-th component in y depends only on the i-th and higher layer of transmitted symbols at time t as follows

$$y_t^i = (R_{i,t})_t x_t^i + n_t^i + \sum_{j=i+1}^{M} (R_{i,j})_t x_t^j \tag{4.10}$$

Considering x_t^i as the current desired detected signal, equation 4.10 shows that y_t^i contains a lower level of interference than in the original received signal r_t, as the interference from x_t^l, for $l<I$, has been suppressed. The third term represents the contribution from other interferers, which can be cancelled by using the available decisions assuming that they have been detected. The final decision statistics on x_t^i can be rewritten as

$$y_t^i = \sum_{j=i}^{M} (R_{i,j})_t x_t^j + n_t^j; i = 1,2,...,n_T \tag{4.11}$$

i.e., the estimate on the transmitted symbols is given by

$$\bar{x}_t^j = q\left(\frac{y_t^i - \sum (R_{i,j})_t \bar{x}_t^j}{(R_{i,i})_t} \right); i = 1,2,..,n_T \tag{4.12}$$

where q(x) denotes the hard decision on x. As additional information, appendix A shows a brief introduction to the QR decomposition [18].

In the MMSE detection algorithm, the expected value between the transmitted vector x and the linear combination of the receiver vector $w^H r$ is minimized

$$\min E\left[\left(x - w^H r \right)^2 \right] \tag{4.13}$$

where w is an NxM matrix of linear combination coefficients given by

$$w^H = \left[H^H H + \sigma^2 I_{n_T}\right]^{-1} H^H \quad 4.14$$

The variance is denoted by σ^2, I_M is an MxM identity matrix. The decision statistics for the transmitted symbol from antenna i at time t are obtained as

$$y_t^i = w_t^H r \quad 4.15$$

where i is the number of columns we are reviewing. The estimate of the symbol sent by antenna i, denoted as $\bar{x}_t^i$, is obtained by making a hard decision on y_t^i

$$\bar{x}_i^t = q\left(y_i^t\right) \quad 4.16$$

In detection algorithms with interference suppression only, the detector calculates the hard decisions estimates by using equations 4.15 and 4.16 for all transmit antennas.

In the case where interference suppression and interference cancellation are used, the receiver starts from antenna M and computes its signal estimates by using equation 4.15 and 4.16. The received signal, r, in level M is denoted by r^M.

For the next received signal, the calculation subtracts the interference contribution of the hard estimate of the previous layer and this modified received signal is used for the computation of the decision level corresponding to antenna (M-1).

The interference from level M-1 is subtracted from the received signal and this signal is used to calculate the decision statistics in 4.15 for antenna (M-2). This process continues until the level one is reviewed (layer for the first antenna). After detection of level i, the hard estimate $\bar{x}_t^i$ is subtracted from the receiver signal to remove its interference contribution, giving rise to the received signal for level i-1

$$r^{i-1} = r^i - \bar{x}_t^i h_i \quad 4.17$$

where h, is the i-th column in the channel matrix H, corresponding to the path attenuations from antenna i. The operation $\bar{x}_t^i h_i$ copies the interference contribution caused by $\bar{x}_t^i$ in the received vector. The unique element with no interference is the r^{i-1}. For estimation of the next antenna signal, the no interference element is used in equation 4.15 instead of r. Finally, a version of the channel matrix is calculated, denoted by H_d^{i-1}, by deleting column i from H_d^i. The deflated matrix at the (M-i+1)-th cancellation step is given by

$$H_d^{i-1} = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,i-1} \\ h_{2,1} & h_{2,2} & \dots & h_{2,i-1} \\ \dots & \dots & \dots & \dots \\ h_{n_R,1} & h_{n_R,2} & \dots & h_{n_R,i-1} \end{bmatrix} \qquad 4.18$$

This deflation is needed as the interference associated with the current symbol has been removed. In real situations when channel coefficients are random, the deflation is used in equation 4.14 or when we are computing the MMSE coefficients and the signal estimate from antenna *i*-1. Once the symbols from each antenna have been estimated, the receiver repeats the process on the vector r_{t+1} received at time (t+1).

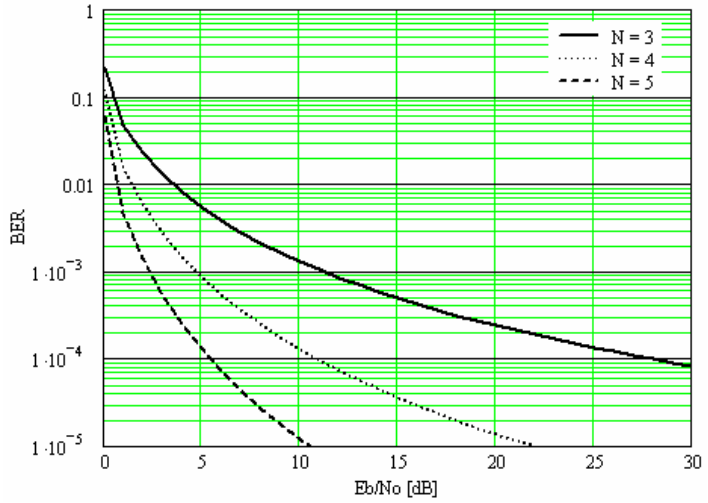


Figure 4.5: VBLAST example with M=4 and N=3,4,5, with QR decomposition, MMSE interference.

Figure 4.5 shows the interference free estimation at a singular layer. It can be described by the expression [19]

$$P_b = \left[\frac{1}{2}(1-\mu)\right]^N \sum_{k=0}^{k=N-1} \left[\frac{1}{2}(1+\mu)\right]^k \qquad 4.19$$

where $\mu = \sqrt{\dfrac{E_b/N_0}{N}}{1+\dfrac{E_b/N_0}{N}}$.

Finally, from figure 4.5, we can see that the coded system with multiple antennas have several improvements with respect to systems with only one receive antenna. We can also observe that, the higher the number of transmit and receive antennas, the better the performance in this kind of systems in terms of BER in environments where the SNR is degraded.

4.4 COMPARISON BETWEEN LST ARCHITECTURES

In BLAST systems (LST), the most important characteristic is that transmitter does not need any channel information. It is the receiver that uses this information in order to estimate the received signals. The principal feature in LST systems is the transmission of different information simultaneously with different transmit and receive antennas pairs in the same frequency band. It has been shown that initial Diagonal BLAST architecture is theoretically capable of approaching the theoretical spectral efficiency, but at a high complexity cost [1][2]. DLST uses multielement antenna arrays, and an elegant diagonally layered coding structure, in which code blocks are dispersed across diagonals. The diagonal processing structure leads to theoretical rates which grow linearly with the number of transmit antennas, with these rates approaching 90% of Shannon capacity.

This is the reason why Foschini developed Vertical BLAST, which still achieves a portion of that efficiency, where every transmit antenna radiates an equal-rate independently encoded stream of data. The independence enables the utilization, at the receiver, of interference rejection and cancellation techniques with the added advantage that the multiple streams are precisely synchronized. A VBLAST receiver can be regarded, therefore, as a multi-stage synchronous multiuser detector. This type of successive cancellation methods have already proved very effective in other context {referencias]. VBLAST, implemented by Bell Labs in real-time laboratory, have demonstrated spectral efficiencies as high as 40bits/s/Hz in an indoor slow fading environment at averages SNR's ranging from 24 to 34 dB [13].

The bit-stream in BLAST systems is demultiplexed into different sub streams (one per each transmit antenna), encoded into symbols and fed to its respective transmit antenna. Depending on the type of LST system (V-BLAST or D-BLAST), the stream will either introduce redundancy or not. V-BLAST encodes each sub stream independently from the others without redundancy; hence each antenna transmits different bits independently from the others. On the contrary, DBLAST introduces redundancy between all sub streams through specific coding strategies. This characteristic offers higher spectral efficiencies that VBLAST, but increases the computational complexity [6].

Making a comparison with horizontal BLAST, we can see that this architecture encodes the signals transmitted by each antenna. The performance of HLST with interference cancellation using previously decoded symbols is limited by the weakest code [3].

4.5 CONCLUSIONS

In this chapter it has been shown that VLST systems present the best tradeoff between performance and computational complexity as compared to other space-time coding schemes. We decided to work directly with this architecture because it allows us to evaluate improvements to this system. In this thesis, we are going to introduce some variations of the VLST in the decoding process in order to reduce computational complexity. In the next chapter, these variations will be compared from a performance and complexity point of view.

REFERENCES

[1] G. Foschini, "Layered Space-Time Architectures for Wireless Communications in a fading environments when using multi-elements antennas", Bell Labs Technical Journal, Autumn 1996, pp. 41-59.
[2] G. D. Golden, G. J. Foschini, R. A. Valenzuela and P. W. Wolniansky, "Detection algorithm and initial laboratory results using the V-BLAST space-time communication architecture", Electronics Letters, vol. 35, no. 1. Jan. 7, 1999, pp. 14-15.
[3] E. Biglieri, "Decoding Space-Time Codes with BLAST Architectures". IEEE Transactions on Signal Processing, vol. 50, no. 10, October 2002.
[4] B. Hassiby, "An efficient square-root algorithm for BLAST", in Proc. IEEE Int. conf. Acoust., Speech, Signal Process., Istanbul, Turkey, June 5-9, 2000, pp. 11737-11740.
[5] S. Bäro, G. Bauch, A. Pavlic, A. Semmler, "Improving BLAST performance using space-time block codes and turbo decoding", in Proc. IEEE Globecom, San Francisco, CA, Nov. 2000, pp. 1067-1071.
[6] R. Gaspa, J. R. Fonollosa, "Comparison of Different Transmit Diversity Space-Time Code Algorithms",
[7] H. El Gramal, A. R. Hammons, "The Layered Space-Time Architecture: a New Perspective", IEEE Trans. Inform. Theory, vol. 47, pp. 2321-2334, Sept. 2001.
[8] P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela, "VBLAST: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel", in Proc. ISSSE, Pisa, Italy, Sept. 1998.
[9] R. Gonzali, R. M. Buehrer, B. D. Woerner, "On the Performance of Scheduling over Space-Time Architecture", 0-7803-7468-1/02, 2002, IEEE.
[10] N. Prasad, M. K. Varanasi, "Optimizing the Performance of D-BLAST Lattice Codes for MIMO Fading Channels", in Proc. Allerton Conf. on Comm., Control and Comput., Monticello, IL, Oct. 2002, University of Illinois.
[11] J. Du, Y. G. Li, "Channel Estimation for D-BLAST OFDM Systems", NSF under Grant CCR-0121565 and the research gift Hughes Network System and Bell Labs of Lucent Technologies.
[12] T. L. Manzetta, "BLAST training: Estimating Channel Characteristics for High Capacity Space-Time Wireless", Proceedings 37th Annual Allerton Conference on Communications, Control and Computing, Monticello, IL, Sept. 22-24, 1999.
[13] G. Bauch, "Introducing to Multi-Antenna Systems and Space-Time Codes", Intitute for Communications Engineering, http://www.intel.tum.de.
[14] T. Guess, H. Zhang, T. V. Kotchiev, "The Outage Capacity of BLAST for MIMO Channels", NSF grant CCR-0093114.
[15] A. Lozano, C. Papadias, "Space-Time Receiver for Wideband BLAST in Rich-Scattering Wireless Channels", VTC2000 IEEE.
[16] A. Ranheim, A. P. des Rosiers, P. H. Siegel, "An Iterative Receiver Algorithm for Space-Time Encoded Signals", UC CORE Grant C98-07.
[17] E. Agrell, T. Eriksson, A. Vardy, K. Zeger, "Closest Point Search in Lattice". IEEE Trans. On Information Theory, Vol. 48, No. 8, August 2002, pp. 2201 – 2214.
[18] G. Golub, C. Van Load, "Matrix Computations", The Johns Hopkins University Press. 1996.
[19] B. Vucetic, J. Yuan, "Space-Time Coding", WILEY, 2002.

5

VERTICAL BELL LABS LAYERED SPACE-TIME ARCHITECTURE VARIANTS: PERFORMANCE AND COMPLEXITY COMPARISON.

5.1 INTRODUCTION

As we have shown in last chapter, VBLAST has better tradeoff between performance and computational complexity. Its principal characteristic is the uncoded data stream, which is demultiplexed into M sub-streams, each being transmitted simultaneously by one transmit antenna. At the receiver, received signals from the N received antennas are detected by a decision feedback algorithm. At present time, several implementations of VBLAST have been presented in the literature since the original idea was published. These implementations aim to reduce the computational complexity of the VBLAST algorithm while maintaining the error probability performance [2].

In this chapter, three variants of the VBLAST architecture are analyzed from a performance-complexity standpoint. The variants considered are the Singular Value Decomposition (V-SVD) [6], Sorted QR Decomposition (V-SQR) [8] and the Least-Square algorithm (V-LS) [9]. In addition, these variants are compared to the near-optimum Maximum Likelihood (CP) algorithm proposed by Agrell [12].

The computational complexity will be obtained empirically, by running each variant in a computer and keeping a record of the arithmetic operations and memory requirements needed to decode each information bit. Nevertheless, a general formula which mathematically approximates the simulated complexity will be obtained.

Topics are considered as follows:

- Section 5.2: System Description
- Section 5.3: VBLAST with Singular Value Decomposition variant
- Section 5.4: VBLAST using Shorter QR Decomposition variant
- Section 5.5: VBLAST over Least Square variant
- Section 5.6: Comparison between different variants with CP algorithm
- Section 5.7: Conclusions

5.2 System Description

The V-BLAST system is shown in figure 5.1. In our case, the system consists of M 16-QAM transmitters operating at the same frequency band and at a symbol rate of 1/T symbols/s with synchronized symbol timing. For simplicity, we assume that the

transmission is organized in bursts of L symbols. The block size is L=10 symbols. Power in each transmitter is proportional to 1/M and the total power is constant and independent of M [4].

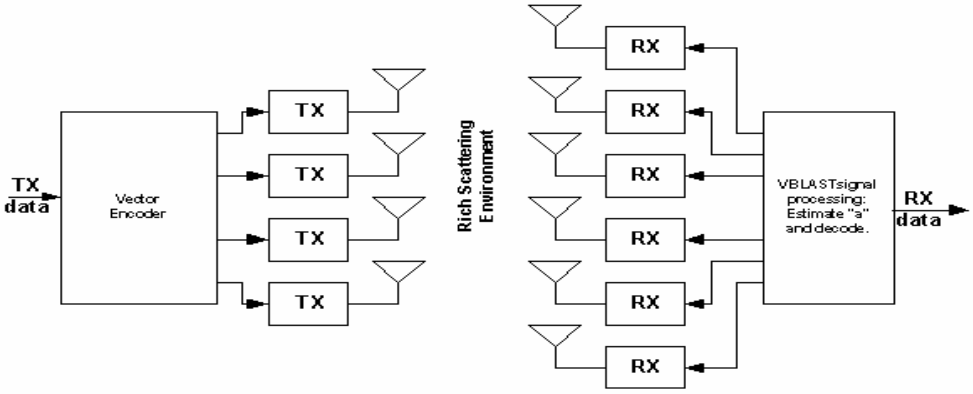


Figure 5.1: VBLAST architecture.

The receiver consists of N conventional QAM receivers. They receive the signals from all M transmit antennas. Flat fading is assumed, and the matrix channel transfer function is $\boldsymbol{H}_{\mathrm{MxN}}$, where $h_{i,j}$ is the complex channel transfer function from transmit antenna i to receive antenna j (N≥M). The channel is Gaussian-distributed with zero mean and variance 0.5 [3].

In the model, it is assumed that the detection process is symbol-synchronous. Letting $\boldsymbol{x} = (x_1, x_2, ..., x_M)^T$ denote the vector of transmitted symbols, the corresponding received N-vector, r is

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} \qquad 5.1$$

where v is a white-noise Gaussian vector. In addition, the channel is quasi-stationary implying negligible variations over the L symbol periods, and it is estimated accurately by means of a training sequence embedded in each burst. Hence, we will not make distinction between $\boldsymbol{H}$ and its estimate. This assumption is referred to as block-fading channel [1]. Each set of L symbol vectors is known as a block. Next, the analysis of the different decoding approaches is discussed. Several variants of VBLAST are described, followed by an analysis of these and the CP decoding algorithm.

5.3 VBLAST WITH SINGULAR VALUE DECOMPOSITION

The Singular Value Decomposition (SVD) approach is formed by a QR Decomposition algorithm which is used to reduce the size of the channel matrix, simplifying in turn the complexity. The channel matrix must have more rows than columns ($\Re^{N \geq M}$). Nevertheless, using QR decomposition may produce losses in the channel matrix representation which affect the detection process. An alternative method of computing the SVD is described by Golub and Kahan [11]. This alternative technique finds matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ simultaneously by implicitly applying a symmetric QR algorithm to $\boldsymbol{A}^T\boldsymbol{A}$, where $\boldsymbol{A}$ is any matrix in the system. This step reduces $\boldsymbol{A}$ into an upper bi-diagonal form using equation 5.2 as follow

$$\boldsymbol{U}_{\boldsymbol{B}}^{\boldsymbol{T}}\boldsymbol{A}\boldsymbol{V}_{\boldsymbol{B}} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \rightarrow \boldsymbol{B} = \begin{bmatrix} d_1 & f_1 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & d_n \end{bmatrix} \tag{5.2}$$

The SVD algorithm uses the eigenvalues. The eigenvalues of a matrix $\boldsymbol{A}$ in a space are the n roots of its characteristic polynomial $p(z) = \det(z\boldsymbol{I} - \boldsymbol{A})$. The set of these roots is called the spectrum and is denoted by $\boldsymbol{\lambda(A)} = \{\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n\}$.

The problem here is to calculate the SVD of $\boldsymbol{H}$ (matrix $\boldsymbol{B}$ in equation 5.2). Applying an implicit shift to the triangular matrix $\boldsymbol{T}=\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{B}$ we can compute the eigenvalues with next expression, which is closer to $d_n^2 + f_m^2$.

$$\boldsymbol{T}(m:n, m:n) = \begin{bmatrix} d_m^2 + f_m^2 & d_m f_n \\ d_m f_n & d_n^2 + f_n^2 \end{bmatrix}; \rightarrow m = n - 1 \tag{5.3}$$

The second step is to compute $c_1 = Cos(\phi_1)$ and $s_1 = \sin(\phi_1)$ such that

$$\begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix}^T \begin{bmatrix} d_1^2 - \lambda \\ d_1 f_1 \end{bmatrix} = \begin{bmatrix} * \\ 0 \end{bmatrix} \tag{5.4}$$

and set $\boldsymbol{G_1} = \boldsymbol{G}(1,2,\phi)$.

Third step consist in computing the Givens rotations [5] $\boldsymbol{G_2}$, … , $\boldsymbol{G_{n-1}}$ so that if $\boldsymbol{Q} = \boldsymbol{G_1 \dots G_{n-1}}$ then $\boldsymbol{Q^T T Q}$ is tri-diagonal and $\boldsymbol{Q e_1} = \boldsymbol{G_1 e_1}$. Notice that these calculations require the explicit formation of $\boldsymbol{B^T B}$.

Algorithm V-SVD

a. Let i = 1
b. Repeat M times.
   - Identify w[j] that are too small, and invert others. $H_3 = G_3^{-1}$.
   - Let V = W^{-1}.
   - Let T = V*U'
   - Apply VBLAST algorithm to all columns
   - Let i = i + 1
c. Quantizes a number to its closest coordinate.

Figure 5.2: V-SVD algorithm.

Once an upper bi-diagonal matrix $\boldsymbol{A}$ is constructed, the V-BLAST algorithm can be carried out. The V-BLAST algorithm iterates over each symbol vector, estimating each symbol in

turn. It starts by choosing the symbol with best SNR, and then subtracts the estimated symbol from the remaining ones. Furthermore, at each step a whole column of matrix $\boldsymbol{A}$ is removed; thus, the pseudo-inverses are calculated on matrices of reduced size. These pseudo-inverses are calculated only once requiring thus a much lower computational complexity than that of the CP algorithm. The SVD approach is described in figure 5.2 [6].

Step b summarizes the calculations described before, and the VBLAST algorithm is described in figure 5.3. The input to the algorithm is the channel matrix after being processed by the SVD algorithm.

Algorithm VBLAST

a. Let $i = 1, A = H$
b. Repeat M times:
   - Find $G_i = A^+$
   - Let $k_i = \text{argmin}_j \|(G_i)_j\|$
   - Let o_i equal to corresponding column in H
   - Remove column k_i of A
   - Let $i = i + 1$
c. Repeat L times:
   - Let i = 1
   - Repeat M times:
     - Let w equal to row k_i of G_i
     - Let $y = w^H r_i$
     - Let $a_{i,o_i} = fq(y)$
     - Let z equal to column o_i of H
     - Let $r_i = r_i - a_{i,o_i} z$
     - Let $i = i + 1$

Figure 5.3: VBLAST algorithm.

In the figure, $(\boldsymbol{G})_j$ is row j of $\boldsymbol{G}$, $\boldsymbol{H}^+$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{H}$ [10], $\boldsymbol{w}^H$ is the conjugate transpose of vector $\boldsymbol{w}$, and $f_q(.)$ is the appropriate quantizing operation for the constellation in use (16-QAM). VBLAST algorithm, as we said, iterates over each symbol vector, estimating each symbol in turn. It starts by choosing the symbol with best SNR, and then subtracts the estimated symbol from the remaining ones. It can be seen that that only M pseudo-inverses are needed per block.

One important point is that at each step, one whole column of matrix $\boldsymbol{A}$ is removed; thus, the pseudo-inverses are calculated on matrices of decreasing size, which makes the entire algorithm faster. Third part of the step b in figure 5.2 is needed to preserve the symbol ordering relative to the original channel matrix $\boldsymbol{H}$ and not to matrix $\boldsymbol{A}$.

The SVD-BLAST complexity can be divided into two parts: the block setup phase, which is computationally expensive but is performed only once per block, and the symbol estimation phase, which is simpler but is performed many times. Hence, if the number of transmit and receive antennas increase, computational complexity grows accordingly. The storage requirements and computational complexity of this algorithm can be approximated by the following expressions

$$Mem = 32MNL + 24ML \quad 5.5$$
$$Add = 16MNL + 16MN + 80MN^4 + 64M^2N^2 \quad 5.6$$
$$Mult = 18MNL + 60MN + 64M^2N^2 + 4M^2 \quad 5.7$$
$$Div = 6MNL + 8(M^2 + N^2) \quad 5.8$$
$$Srt = 1.5L + 8MN \quad 5.9$$

where Add, Mult, Div and Srt represent addition, multiplication, division and square root operations per decoded block, respectively, and Mem is the memory required per decoded block as well; L is the number of bits per block. If we want the number of operations per decoded bit, we have to divide these equations by the number of bits per block (i.e., a block is defined as the number of bits per symbol – 4 bits/symbol –, multiplied by the number of bits per block, multiplied by the number of transmit antennas).

5.4 VBLAST USING SORTED QR DECOMPOSITION

In this V-BLAST variant, instead of calculating the channel matrix pseudo-inverses, it estimates the length of the rows of a matrix $\boldsymbol{H}^+$, using the length of columns of matrix $\boldsymbol{H}$ ($\boldsymbol{H}^+$ is orthogonal to the vector space $\boldsymbol{H}$). This algorithm is an extension of the modified Gram-Schmidt algorithm by ordering the columns of $\boldsymbol{H}$ in each orthogonalization step [7].

The algorithm applies the Gram-Schmidt algorithm to compute matrix $\boldsymbol{R}$ line by line from top to bottom and matrix $\boldsymbol{Q}$ column by column from left to right. This is done by computing the elements of matrix $\boldsymbol{Q}$ so as to compute the elements of matrix $\boldsymbol{R}$ in a recursive manner.

The algorithms works as follow [8]:

For a given $\boldsymbol{H}$, it calculates q_1 of unit length and $r_{1,1} = |h_1|$ to fulfill $h_1 = r_{1,1}q_1$. In the next step, the components of h_2 in the direction of q_2 are cancelled and q_2 of unit length and $r_{1,2}$ and $r_{2,2}$ are computed to fulfill $h_2 = r_{2,1}q_1 + r_{2,2}q_2$. The computation of the next step has the same form, thus the diagonal elements $r_{k,k}$ form the length of h_k, orthogonal to vector $\boldsymbol{q}$, and vector $\boldsymbol{r}$ describes the projection of $\boldsymbol{H}$ into the vector space spanned by vector $\boldsymbol{q}$. Consequently the diagonal elements are calculated from $r_{1,1}$ to $r_{M,M}$, but it would be optimal to maximize the $|r_{k,k}|$ in every decoding step. To maximize the diagonal elements from M to 1 by applying only one QR decomposition, the idea of this algorithm is to find the permutation of $\boldsymbol{H}$ that minimizes each $|r_{k,k}|$ with k running from 1 to M, leaving all $r_{j,j}$ with $j<k$ unchanged.

This operation minimizes the diagonal elements in every decomposition step and thereby estimates maximal diagonal elements in the succeeding steps. The only change to the modified Gram-Schmidt algorithm consist in a reordering of the columns of $\boldsymbol{H}$ according to their minimum length, orthogonal to the vector space already spanned by vector $\boldsymbol{q}$. This is performed in step d in figure 5.4.

Algorithm V-SQR

b. Find the sorted-QR decomposition of H store it in Q and R.
c. Let i = 1.
d. Repeat L times:
   - Let $x = Q^H \cdot r_i$
   - Find $\hat{a}_i$ that solves $R\hat{a}=x$
   - Re-arrange $\hat{a}_i$ to correspond to original ordering of columns of H.
   - Let i = i + 1
e. Quantizes a number to its closest coordinate.
f. Reorder estimated symbols.

Figure 5.4: Sorted QR Algorithm.

In order to get a good approximation of the number of operations made in this algorithm, we found the following formulas

$$Mem = 4MNL + 2ML + 7M^2L + 10L + 10(MN + M^2N) \quad 5.10$$

$$Add = 6MNL + 6M^2L + 8L + 10M^2N \quad 5.11$$

$$Mult = 5MNL + 4M^2L + 4L \quad 5.12$$

$$Div = 3ML + 2MN \quad 5.13$$

$$Srt = M \quad 5.14$$

5.5 LEAST SQUARE VBLAST

In comparison to SVD-VBLAST, this algorithm needs an extra QR decomposition (Orthogonal Matrix Triangulation); nevertheless, pseudo-inverses are performed on an MxM matrix, which renders the computational complexity smaller. The symbol estimation has an extra vector-matrix multiplication, but the size of all other operations depends exclusively on M instead of M and N.

The estimation problem can be seen as solving a system of linear equations perturbed by noise, with the added constraint that the solution must be an element of the modulation constellation. To do this, the algorithm modifies the conventional QR decomposition to solve equation 5.1 as follows

$$\boldsymbol{r} = \boldsymbol{QRa} + \boldsymbol{v} \quad 5.15$$

$$\boldsymbol{Q^H r} = \boldsymbol{Ra} + \boldsymbol{Q^H v} \quad 5.16$$

$$\boldsymbol{x} = \boldsymbol{Ra} + \boldsymbol{v} \quad 5.17$$

where $\boldsymbol{Q}$ is an NxM matrix with orthonormal columns, $\boldsymbol{R}$ is an MxM upper-triangular matrix and $\boldsymbol{v}$ is a Gaussian-noise vector. It is important to note that some multiplications, divisions and square root operations are transformed into additions, reducing the overall complexity in a considerable manner as compared to the other algorithms [9]. The Least Square algorithm is described in figure 5.5.

Algorithm V-LS.

a. Compute an NxM orthonormal matrix Q and an MxM upper-triangular matrix R, such that H = QR.
b. Let i = 1, A = R.
c. Repeat M time:
   i. Find $G_i = A^+$
   ii. Let
   $$k_i = \arg\min_j \left\| (G_i)_j \right|$$
   iii. Let o_i equal to corresponding column in H.
   iv. Remove column k_i of A
   v. Let i = i + 1
d. Let i = 1
e. Repeat L times:
   i. Let $x = Q^H \cdot r_i$
   ii. Repeat M times:
   - Let W equal to row k_i of G_i
   - Let $y = W^H x$
   - Let $\ddot{o}_{i,o} = f_q(y)$
   - Let z equal to column o_i of H
   - Let $x = x - \ddot{o}_{i,o} z$
   - Let i = i + 1

Figure 5.5: V-Least-Square algorithm.

The computational complexity of this algorithm is approximating made in follow equations

$$Mem = 35MNL + 14ML + 4L + 10MN^2 \quad 5.18$$
$$Add = 16MNL + 16ML + 4L + 10M^2N \quad 5.19$$
$$Mult = 16MNL + 12ML \quad 5.20$$
$$Div = MNL \quad 5.21$$
$$Srt = M \quad 5.22$$

5.6 COMPARISON BETWEEN VBLAST VARIANTS AND CLOSEST POINT ALGORITHM

In this section, a comparison between the VBLAST variants discussed above is made in terms of BLER performance and computational complexity. These approaches are also compared to the CP algorithm described in the last chapter.

5.6.1 BLER PERFORMANCE

Tables 5.1 through 5.3 present a performance comparison between the closest point algorithm and the VBLAST variants in terms of Bit Error Rate, Block Error Rate, storage requirements and computational complexity for SNR_{avg} = 15dB and arrays of 2x4, 2x6 and 2x8 antennas. From these tables we can observe that the best BLER performance is

obtained with the CP algorithm, while the VBLAST approaches all have the same performance. The difference between them is the computational complexity and storage requirements. We can see that the algorithm presenting the lowest complexity is the V-SQR approach followed by the V-LS algorithm.

2x4					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		5.22E-03	1.95E-01	61.77	63.71
V-SVD		5.70E-03	2.09E-01	39.35	62.29
V-LS		5.70E-03	2.09E-01	13.18	30.38
V-SQR		5.70E-03	2.09E-01	11.53	25.29

Table 5.1: Performance analysis for two transmit and four receive antennas.

2x6					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		4.45E-04	2.74E-02	76.16	81.49
V-SVD		5.17E-04	3.06E-02	56.80	88.46
V-LS		5.17E-04	3.06E-02	15.73	37.48
V-SQR		5.17E-04	3.06E-02	14.83	33.44

Table 5.2: Performance analysis for two transmit and six receive antennas.

2x8					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		4.95E-05	3.63E-03	90.65	99.43
V-SVD		5.73E-05	4.00E-03	65.96	102.74
V-LS		5.73E-05	4.00E-03	18.28	44.58
V-SQR		5.73E-05	4.00E-03	18.13	41.59

Table 5.3: Performance analysis for two transmit and eight receive antennas.

Figure 5.6 presents a BLER performance comparison between the CP algorithm and the VBLAST algorithm for the arrays discussed above (2x4, 2x6 and 2x8), as a function of received SNR. We can see that both algorithms present almost the same performance. Nevertheless, computational complexity is much higher with the CP algorithm. This means that VBLAST variants are excellent alternatives when computational complexity is at a prime.

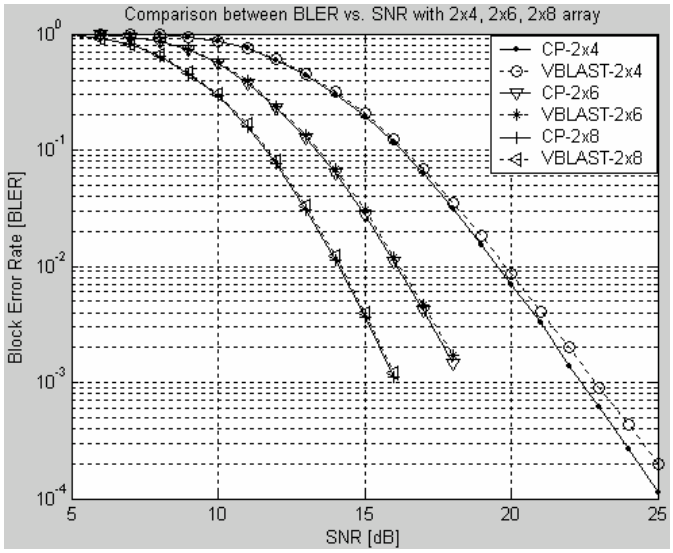


Figure 5.6: BLER vs SNR_{avg} M = 2; N = 4, 6 and 8

Tables 5.4, 5.5 and 5.6 present simulation results when the number of transmit antennas is increased to eight. One again, a comparison is made between the CP algorithm and the VBLAST variants from a performance and complexity point of view. Results in these tables are taken for SNR_{avg} = 17dB.

8x10					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		5.18E-03	3.87E-01	366.42	414.01
V-SVD		1.46E-02	7.04E-01	828.35	1521.25
V-LS		1.46E-02	7.04E-01	93.00	189.98
V-SQR		1.61E-02	7.16E-01	35.08	81.50

Table 5.4: Performance analysis for eight transmit and ten receive antennas.

8x12					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		6.16E-04	1.14E-01	360.94	409.97
V-SVD		2.75E-03	2.98E-01	973.62	1751.86
V-LS		2.75E-03	2.98E-01	97.05	200.68
V-SQR		2.98E-03	3.06E-01	40.18	94.15

Table 5.5: Performance analysis for eight transmit and twelve receive antennas.

8x14					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		1.28E-04	3.37E-02	388.65	445.97
V-SVD		5.67E-04	9.20E-02	1118.51	1981.15
V-LS		5.67E-04	9.20E-02	101.10	211.38
V-SQR		5.99E-04	9.37E-02	45.28	106.80

Table 5.6: Performance analysis for eight transmit and fourteen receive antennas.

These tables show, as expected, that the best performance is obtained by CP algorithm; we also note that V-LS and V-SVD algorithms have the same performance but the former has lower complexity.

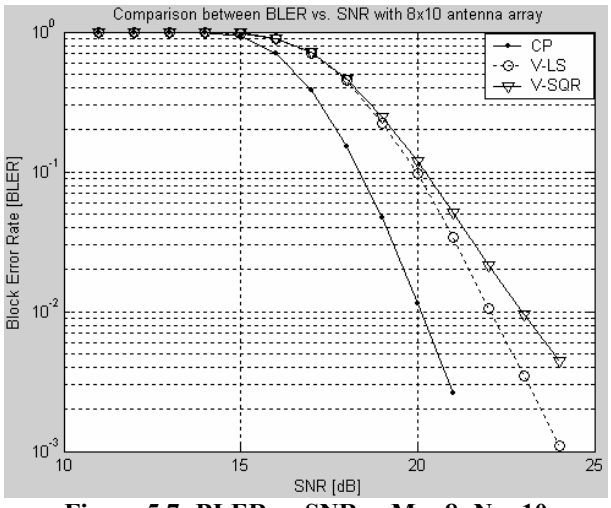


Figure 5.7: BLER vs SNR_{avg} M = 8; N = 10

Figures 5.7 and 5.8 show a performance comparison between the CP algorithm and the V-LS and V-SQR variants of the VBLAST algorithm. Notice that the V-LS approach has

better performance than the V-SQR approach. Also, the CP algorithm presents the best performance, but the V-LS approach is not that far from it. This means that the V-LS variant has a better tradeoff complexity-performance as compared to the CP algorithm. Finally, increasing the transmit antennas from 10 to 12, a performance gains of almost 4 dB are obtained.

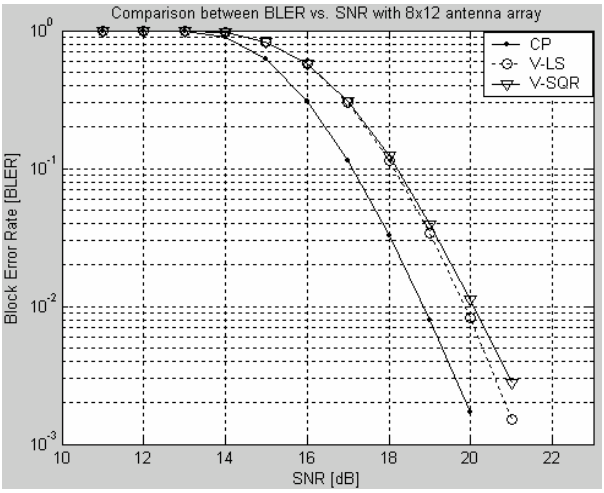


Figure 5.8: BLER vs SNR_{avg} M = 8; N = 12

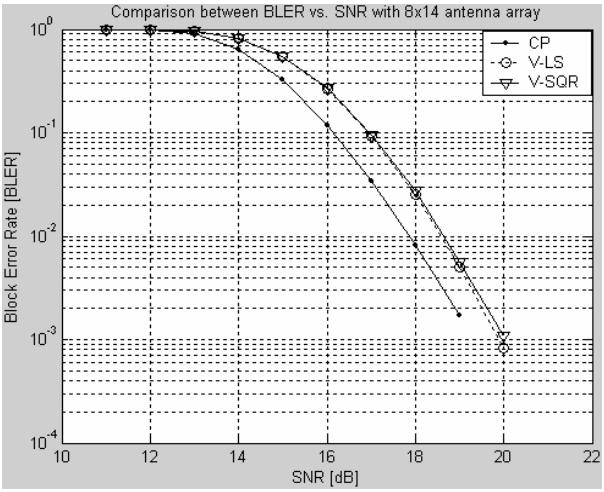


Figure 5.9: BLER vs SNR_{avg} M = 8; N = 14

Finally, performance and complexity results are presented for square MIMO systems. Tables 5.7, 5.8, 5.9 and 5.10 show a comparison between square arrays of 2, 4, 6 and 8 antenna pairs. From these tables we can observe that at a SNR_{avg} of 20dB, BLER is increased almost in 1dB per antenna pair.

2x2					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		2.07E-02	2.95E-01	47.93	46.68
V-SVD		2.34E-02	3.21E-01	22.00	36.49
V-LS		2.34E-02	3.21E-01	10.63	23.28
V-SQR		2.34E-02	3.21E-01	8.23	17.14

Table 5.7: Performance analyses for two transmit and receive antennas.

4x4					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		1.83E-02	3.33E-01	121.03	127.70
V-SVD		3.07E-02	4.75E-01	110.09	208.58
V-LS		3.07E-02	4.75E-01	26.50	55.08
V-SQR		3.23E-02	4.89E-01	14.28	31.44

Table 5.8: Performance analyses for four transmit and receive antennas.

6x6					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		1.37E-02	2.90E-01	283.61	318.30
V-SVD		4.00E-02	5.98E-01	314.41	598.44
V-LS		4.00E-02	5.98E-01	51.68	104.68
V-SQR		4.54E-02	7.00E-01	21.53	48.65

Table 5.9: Performance analyses for six transmit and receive antennas.

8x8					
Algorithm		**BER**	**BLER**	**Storage**	**Ob**
V-CP		9.69E-03	2.24E-01	749.47	867.26
V-SVD		4.54E-02	7.00E-01	680.74	1283.27
V-LS		4.54E-02	7.00E-01	88.95	179.28
V-SQR		5.13E-02	7.20E-01	29.98	68.85

Table 5.10: Performance analyses for eight transmit and receive antennas.

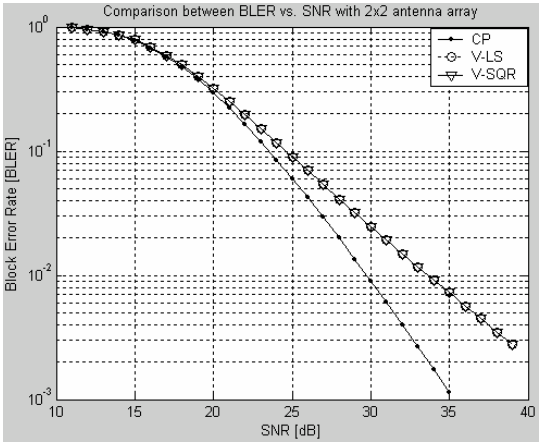


Figure 5.10: BLER vs SNR_{avg} M = 2; N = 2

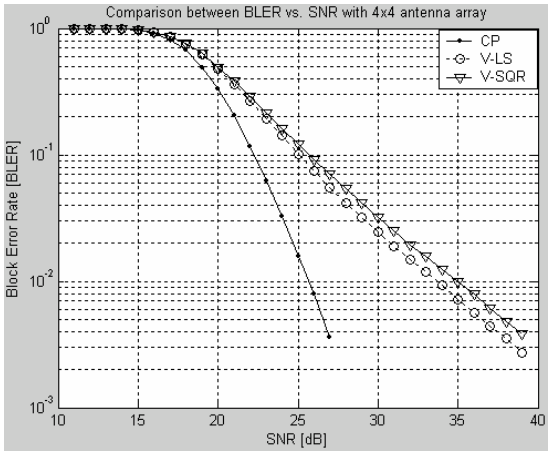


Figure 5.11: BLER vs SNR_{avg} M = 4; N = 4

Finally, figures 5.10 through 5.13 show the BLER performance of square MIMO systems decoded with the CP, V-LS and V-SQR algorithms. First, it is important to note that a significant increase in performance is obtained when the number of antenna pairs is increased and decoded with the CP algorithm. Nevertheless, when the VBLAST variants are used, BLER performance is degraded as the antenna pairs is increased.

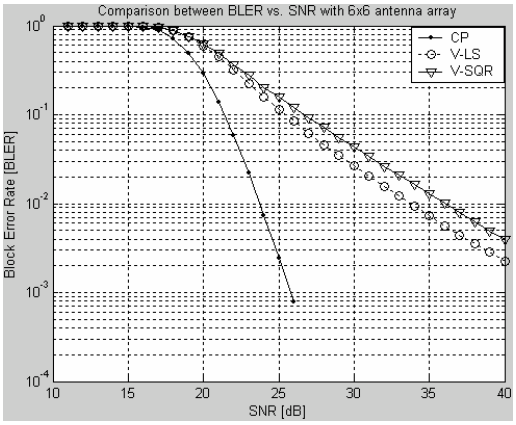


Figure 5.12: BLER vs SNR_{avg} M = 6; N = 6

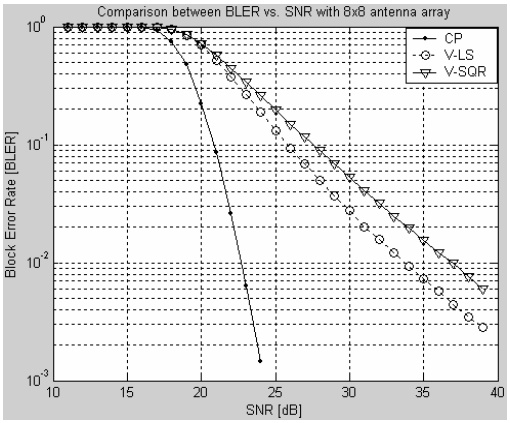


Figure 5.13: BLER vs SNR_{avg} M = 8; N = 8

5.6.2 COMPUTATIONAL COMPLEXITY

As we did in past section, we will make a complexity analysis between the VBLAST variants and the CP algorithm. As complexity regards, table 5.11, 5.12 and 5.13 show the computational requirements of each algorithms. First column indicates the total storage requirements, and the following columns indicate the number of additions, multiplications, divisions and square root operations. All of these for each decoded bit.

2x4						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	61.77	26.58	15.74	2.63	0.10	63.71
V-SVD	39.35	21.42	18.68	1.35	0.21	62.29
V-LS	13.18	10.95	9.30	0.34	0.04	30.38
V-SQR	11.53	9.09	7.35	0.70	0.03	25.29

Table 5.11: Performance analyses for two transmit and four receive antennas.

2x6						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	76.16	32.54	21.03	3.25	0.10	81.49
V-SVD	56.80	29.80	27.27	1.65	0.21	88.46
V-LS	15.73	13.25	11.60	0.44	0.04	37.48
V-SQR	14.83	11.74	10.00	0.80	0.03	33.44

Table 5.12: Performance analyses for two transmit and six receive antennas.

2x8						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	90.65	38.60	26.34	3.88	0.10	99.43
V-SVD	65.96	34.61	31.67	1.91	0.24	102.74
V-LS	18.28	15.55	13.90	0.54	0.04	44.58
V-SQR	18.13	14.39	12.65	0.90	0.03	41.59

Table 5.13: Performance analyses for two transmit and eight receive antennas.

It is clear that the V-LS algorithm has the best tradeoff. It is important to note that in this algorithm, multiplications, division and square root operations are reduced and replaced by addition operations.

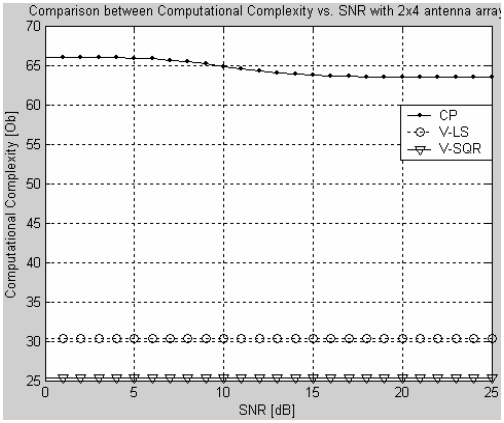


Figure 5.14: Computational Complexity vs SNR_{avg} M = 2; N = 4

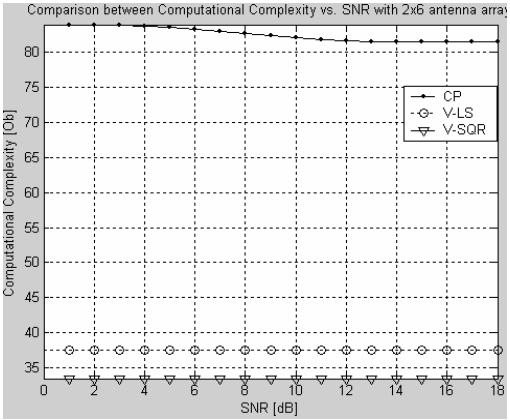


Figure 5.15: Computational Complexity vs SNR_{avg} M = 2; N = 6

Figures 5.14, 5.15 and 5.16 show the overall computational complexity, O_b, as a function of SNR_{avg} for 2x4, 2x6 and 2x8 MIMO systems. As we can see, the V-SQR algorithm presents the lowest computational complexity. Nevertheless, its BLER behavior is not quite

efficient. The second smallest computational complexity is due to the V-LS algorithm which, added to its BLER performance, becomes the best algorithm from a complexity-performance perspective.

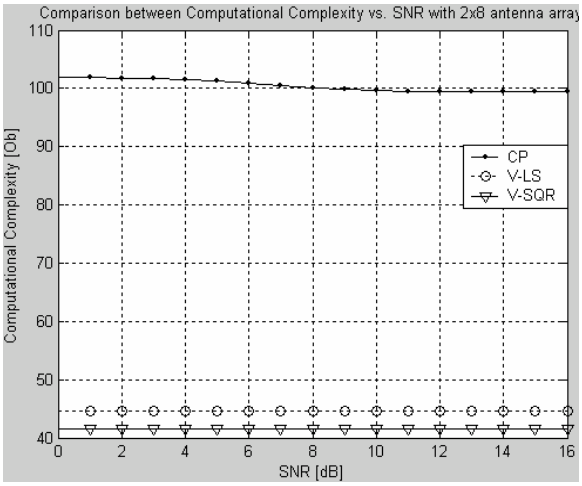


Figure 5.16: Computational Complexity vs SNR_{avg} M = 2; N = 8

Second analysis is made with same considerations in its architecture, so we will show that if we increase the number of transmit antennas, the V-LS algorithm is still our best option.

8x10						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	366.42	152.83	121.17	9.21	0.10	414.01
V-SVD	828.35	405.71	530.35	17.78	4.82	1521.25
V-LS	93.00	63.40	59.65	3.41	0.11	189.98
V-SQR	35.08	27.80	25.80	1.00	0.03	81.50

Table 5.14: Performance analyses for eight transmit and ten receive antennas.

8x12						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	360.94	145.44	124.32	7.75	0.10	409.97
V-SVD	973.62	469.75	612.55	18.76	4.87	1751.86
V-LS	97.05	66.90	63.15	3.51	0.11	200.68
V-SQR	40.18	31.95	29.95	1.10	0.03	94.15

Table 5.15: Performance analyses for eight transmit and twelve receive antennas.

8x14						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	388.65	156.16	136.73	7.98	0.10	445.97
V-SVD	1118.51	533.48	694.30	19.72	4.90	1981.15
V-LS	101.10	70.40	66.65	3.61	0.11	211.38
V-SQR	45.28	36.10	34.10	1.20	0.03	106.80

Table 5.16: Performance analyses for eight transmit and fourteen receive antennas.

Analyzing tables 5.14, 5.15 and 5.16, we can observe that V-CP has the highest arithmetic complexity; nevertheless, V-SVD needs more memory requirements, which basically means that for a high number of transmit antennas, the algorithm increase the complexity logarithmically, rendering its implementation practically impossible. The V-LS approach is still our better option as we can see in figures 5.17, 5.18 and 5.19.

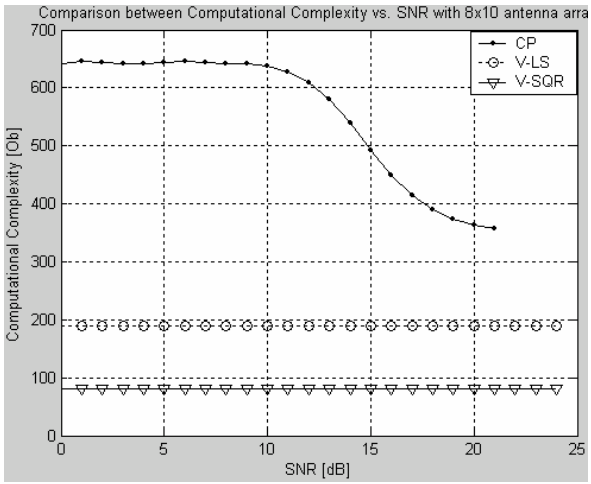


Figure 5.17: Computational Complexity vs SNR_{avg} M = 8; N = 10

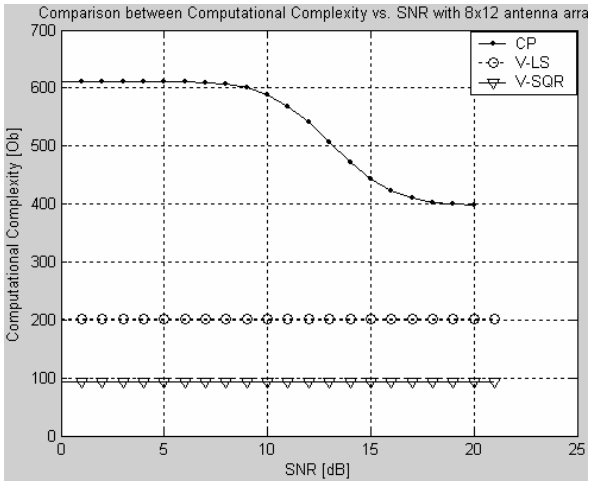


Figure 5.18: Computational Complexity vs SNR_{avg} M = 8; N = 12

It is interesting to observe that if we increase the number of transmit antennas, the computational complexity increases logarithmically. If the number of transmit antennas is small, the computational complexity is almost linear.

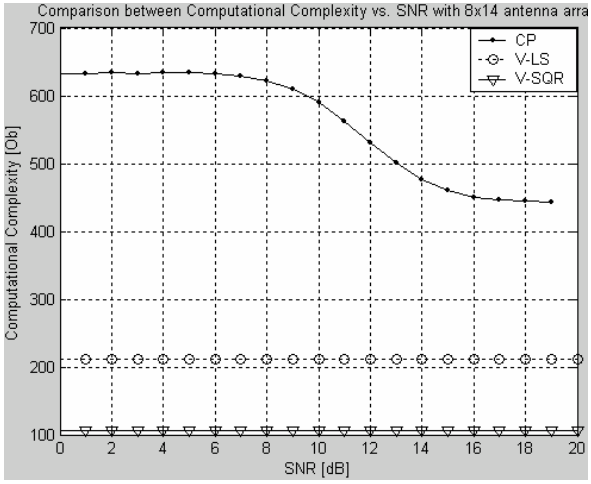


Figure 5.19: Computational Complexity vs SNR_{avg} M = 8; N = 14

2x2						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	47.93	21.03	10.56	2.06	0.10	46.68
V-SVD	22.00	13.14	10.20	1.05	0.21	36.49
V-LS	10.63	8.65	7.00	0.24	0.04	23.28
V-SQR	8.23	6.44	4.70	0.60	0.03	17.14

Table 5.17: Performance analyses for two transmit and receive antennas.

4x4						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	121.03	52.08	33.17	4.43	0.10	127.70
V-SVD	110.09	59.98	67.43	4.46	1.21	208.58
V-LS	26.50	19.00	17.05	0.86	0.06	55.08
V-SQR	14.28	11.24	9.35	0.70	0.03	31.44

Table 5.18: Performance analyses for four transmit and receive antennas.

6x6						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	283.61	129.14	83.92	10.46	0.10	318.30
V-SVD	314.41	162.26	202.90	9.72	2.73	598.44
V-LS	51.68	35.28	32.63	1.89	0.09	104.68
V-SQR	21.53	16.95	15.00	0.80	0.03	48.65

Table 5.19: Performance analyses for six transmit and receive antennas.

8x8						
Algorithm	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP	749.47	358.29	225.52	28.77	0.10	867.26
V-SVD	680.74	340.05	445.46	16.72	4.71	1283.27
V-LS	88.95	59.90	56.15	3.31	0.11	179.28
V-SQR	29.98	23.65	21.65	0.90	0.03	68.85

Table 5.20: Performance analyses for eight transmit and receive antennas.

Finally, we show in tables 5.17, 5.18, 5.19 and 5.20 results obtained for square antenna arrays with a SNR_{avg} equal to 20dB. These tables show that the CP algorithm is still higher in complexity than the V-LS and V-SQR algorithms.

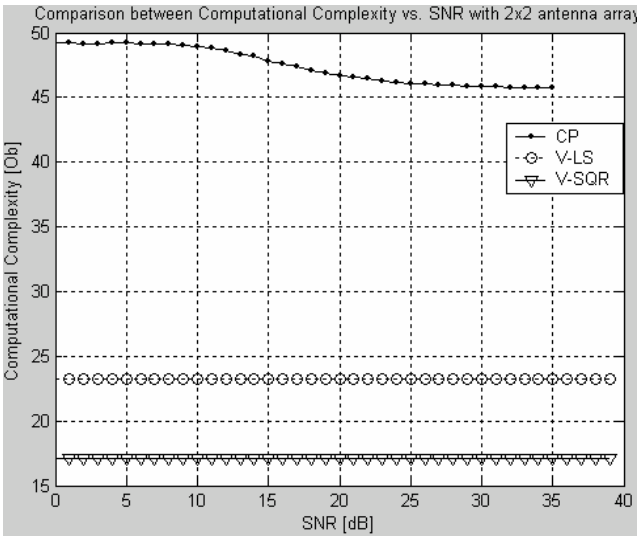


Figure 5.20: Computational Complexity vs SNR_{avg} M = 2; N = 2

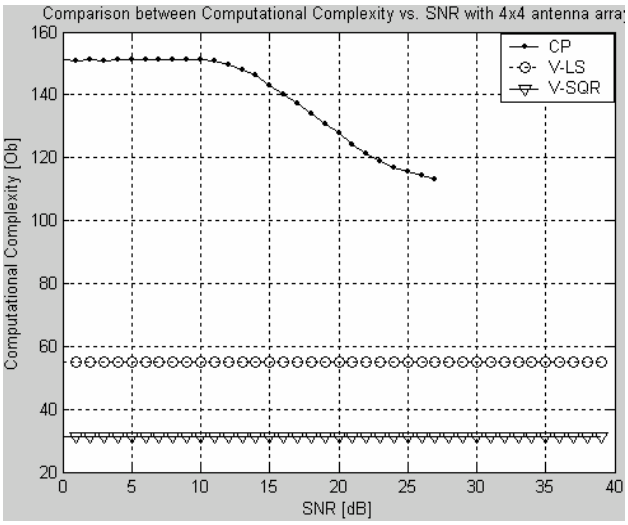


Figure 5.21: Computational Complexity vs SNR_{avg} M = 4; N = 4

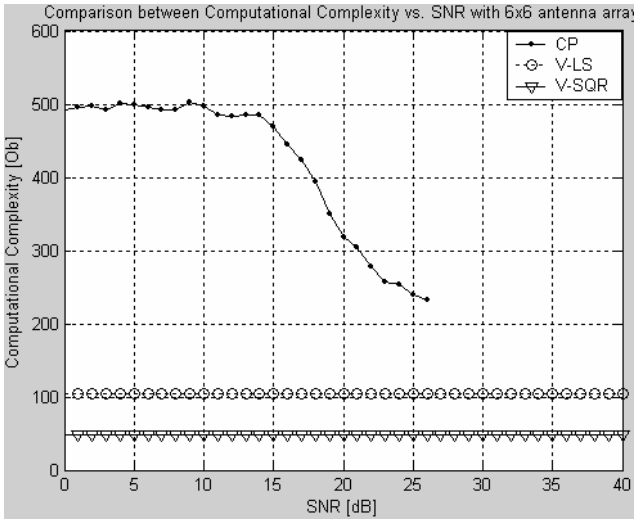


Figure 5.22: Computational Complexity vs SNR_{avg} M = 6; N = 6

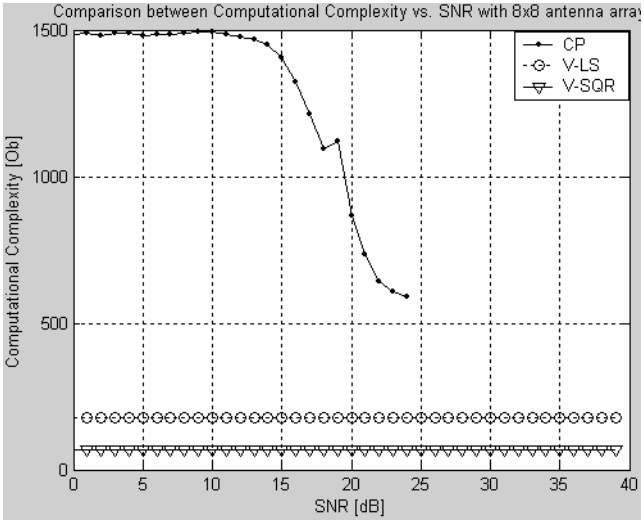


Figure 5.23: Computational Complexity vs SNR_{avg} M = 8; N = 8

5.7 CONCLUSIONS

In this chapter, three MIMO receiver procedures based on the V-BLAST algorithm and one based on the near-optimal CP algorithm have been presented. The performance and complexity of each algorithm have been analyzed and compared in order to determine the algorithm that is best suited for hardware implementation. It has been shown that MIMO systems can achieve much higher data rates than conventional SISO systems. This explains why space-time coding has been considered for standardization purposes in the third generation mobile communication systems (3G) where high rate multimedia services are required. It has also been shown that implementing an algorithm not only has to do with its overall performance but also with its computational burden.

Next chapter is focused on one aspect, additional channel coding in order to improve BLER performance and observe what happens with the computational complexity in a MIMO system.

REFERENCES

[1] E. Biglieri, J. Proakis, and S. Shamai, I: "Fading channels: Information-theoretic and communications aspects, *IEEE Trans.Theory*, vol. 44, no. 6, pp. 2619-2691, Oct. 1998.
[2] G. Golden, C. Foschini, R. Valenzuela, P. Wolniansky, "Detection algorithm and initial laboratory results using VBLAST space-time communications architecture", Electronic Letters, Vol. 35, No. 1, pp. 14-15, January 1999.
[3] L. Liu, L. Li, "A simple soft-detection for the BLAST system", IEEE 2002, 0-7803-7551-3/02
[4] Y. Wa, Y. Yang, X. Lue, "Improving the performance of VBLAST systems with STTC", IOCS 2002, 0-7803-7510-6/02.
[5] D. S. Watkins, "Fundamentals of Matrix Computations", New York, Wiley Inter-science, 2002.
[6] G. H. Golub, C. F. Van Loan, "Matrix Computations", Second Edition. The Johns Hopkins University Press. Baltimore and London. 1996.
[7] M. Bazdresch and J. Rodriguez-Guisantes: "The msim simulator". [Online], Available: http://www.comelec.enst.fr/~rodriguez/msim/index.html
[8] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, K. D. Kammeyer: "Efficient algorithm for decoding layered space-Time Codes". Electronics Letters. 25 October 2001. Vol. 37. No. 22
[9] M. Bazdresch, R. Guisantes: "Least-Squares Techniques Applied to VBLAST Receivers". Submitted for Publication.
[10] N. Boubaker, K. B. Letaief and R. D. Murch: "Transmit and Receive Spatial Multiplexing for Broadband Wireless Communitations". IEEE. 2001, pp. 224 – 229.
[11] G. H. Golub, W. Kahan, "Calculating the Singular Values and Pseudo-inverse of a Matrix", SIAM J., Num. Anal, Ser. B 2, 1965, pp. 205-224.
[12] E. Agrell, T. Eriksson, A. Vardy, K. Zeger, "Closest Point Search in Lattice". IEEE Trans. On Information Theory, Vol. 48, No. 8, August 2002, pp. 2201 – 2214.

6

IMPROVING VBLAST PERFORMANCE WITH CONVENTIONAL CHANNEL CODING

6.1 INTRODUCTION

In the last chapter an analysis of different realizations of the VBLAST architecture was performed in order to select that realization presenting the best complexity-performance tradeoff. We saw that the V-LS approach has this best tradeoff. In the purpose of improving the performance of these VBLAST approaches, while keeping its computational complexity as feasible as possible, a first attempt to concatenate the VBLAST architecture with conventional channel coding techniques is presented in this chapter. Indeed, the aim of this chapter is to analyze the performance improvement of the VBLAST algorithm when a channel codec (coder/decoder) is introduced in the transmission chain.

The channel coder utilized is a convolutional code [2] with hard-input Viterbi decoding [10]. This codec is concatenated to two approaches of the VBLAST architecture, namely the V-LS algorithm and the V-SQR approach, and compared to the CP algorithm. As in the case of chapter 5, BLER performance and computational complexity is analyzed in order to propose the MIMO system with the best trade off.

The analysis presented in this chapter consists in a description of the overall system followed by a presentation of performance and complexity results obtained by computer simulations. Then, a comparison with the near-optimum CP algorithm is done in order to determine which approach is best suited for a hardware implementation.

Topics are considered as follows

- Section 6.2: System Description
- Section 6.3: Performance Results
- Section 6.4: CP comparison
- Section 6.5: Conclusions

6.2 SYSTEM DESCRIPTION

The system analyzed in this chapter is illustrated in figure 6.1. It consists of a convolutional channel coder concatenated to the vertical BLAST system with M 16-QAM modulators which operate at the same frequency band and with symbol rate of 1/T symbols/s with synchronized symbol timing. As in previous chapter, transmission is organized into burst of L symbols from each antenna.

The receiver consists of N conventional 16-QAM receivers (one for each receive antenna). They receive signals from the M transmit antennas. Flat fading is assumed, and the matrix channel function is H_{MxN}. The channel is perturbed with Gaussian-distributed noise with zero mean and variance 0.5.

We can observe that in the transmitter side, the convolutional coder creates dependence between the coded bits; however, in order to detect correctly, VBLAST needs independence between the transmitted bits. Assuming MIMO systems with L=10 symbols per block and convolutional coders with small constraint lengths (3<K<7), the demultiplexer of the VBLAST encoder can be seen as an interleaver. In other words, in cases where the length of the channel code is sufficiently small, as compared to the block size and the number of the transmit antennas, the demultiplexer plays the additional role of an interleaver. In other cases where this condition is not fulfilled, the MIMO system demands the use of an interleaver between the convolutional coder and the demultiplexer shown in figure 6.1.

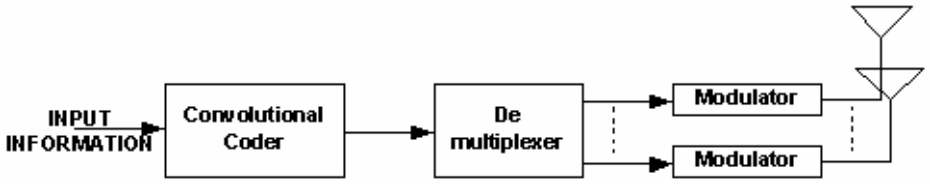


Figure 6.1: MIMO transmitter with convolutional coding.

At the receiver end, the channel matrix is estimated and transmitted symbols are detected and demodulated in the usual way by the VBLAST system. In addition, the detected symbols are further decoded by a convolutional decoder so as to refine/correct further errors that VBLAST decoder was not able to correct. The convolutional decoder used in this chapter is the Viterbi decoder [8]. The output of the Viterbi decoder is the information sequence.

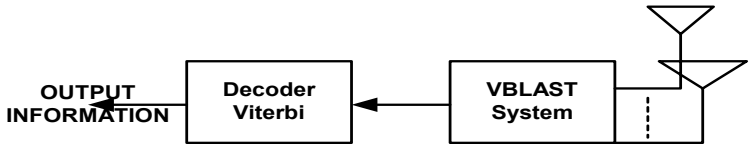


Figure 6.2: MIMO receiver with convolutional decoding.

Convolutional codes are usually described into two parameters: the code rate and the constraint length. The code rate is defined as

$$CodeRate = \frac{k}{n}\text{Code} \qquad 6.1$$

it is the ratio of the information bits at the input of the convolutional encoder (k) to the coded bits (n) at its output. The constraint length of a convolutional code, K, denotes the "length" of the convolutional encoder, i.e. how many k-bit stages are available to feed the combinatorial logic that produces the coded symbols. The other important parameter is $m=kK$, which indicates how many information bits are retained and used for producing the

n output bits. Parameter *m* can be thought of as the memory length of the encoder. In this chapter, we will use convolutional coders with code rates of ½ and 1/3 [14].

In this thesis, Viterbi decoding is used because of its simplicity. It is well suited for hardware implementation; nevertheless, its computational complexity grows exponentially with the constraint length. In practice, it is usually limited to constraint lengths smaller than $K = 9$ [12]. For a thorough discussion on Viterbi decoding, refer to [8, 9, 12, 13]. In addition, appendix B gives a brief description of its functioning.

6.3: PERFORMANCE RESULTS

In this section, performance results of the concatenated convolutional coding and VBLAST system is presented together with the increase in computational complexity that this system may require. Performance is measured in terms of BLER.

BLER performance is analyzed for MIMO systems with 2 transmit antennas and 4, 6 and 8 receive antennas; systems with 8 transmit antennas and 10, 12 and 14 receive antennas; and square arrays with 2, 4, 6 and 8 antenna pairs. Convolutional codes with code rate of ½ and 1/3, and constraint lengths of K=3, 4, 5, 6 and 7 are explored.

6.3.1 BLER PERFORMANCE

6.3.1.1 Code Rate 1/2

This section presents the BLER performance of multiple antenna arrays for the system in figure 6.1 where the convolutional code has a code rate of ½. It is well known that the generator polynomials of a convolutional code are usually selected based on the code's free distance properties. The first criterion is to select a code that does not have catastrophic error propagation and that has the maximum free distance for the given rate and constraint length. [14]. The convolutional codes used in this chapter are listed table 6.1. As can be seen, constraint length from K=3 to K=7 are used. The generator polynomials of each code is indicated in the table

Rate	Constraint Length	Free Distance	Generator Polynomial
½	3	5	111 , 101
½	4	6	1111 , 1011
½	5	7	10111 , 11001
½	6	8	101111 , 110101
½	7	10	1001111 , 1101101

Table 6.1: Optimum Short Constraint Length Convolutional Codes for Rate ½.

In the first part of the analysis, the following MIMO systems are used: 2x4, 2x6 and 2x8. Tables 6.2, 6.3 and 6.4 illustrate the performance/complexity of the V-LS systems with and without convolutional coding for a SNR_{avg} =12dB. In the case where convolutional coding

is in use, constraint lengths of K=3 up to 7 are considered. As can be seen from the tables, a performance gain is achieved when channel coding is concatenated to the V-LS system. Even the simple K=3 convolutional code achieves an important gain in BLER performance over the V-LS system. In addition, as usually expected, increasing the constraint length of the convolutional code entails an increase in the performance of the communication system. Finally, from these tables we can see that the price to pay for obtaining a performance improvement is an increase in computational load and storage requirements. This will be analyzed more deeply in the next section.

2x4					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.25E-02	6.08E-01	13.18	30.38
CC+VB	**3**	8.63E-03	1.01E-01	30.35	72.75
CC+VB	**4**	8.63E-03	8.74E-02	34.35	84.75
CC+VB	**5**	1.04E-02	8.41E-02	42.35	108.75
CC+VB	**6**	1.20E-02	8.86E-02	58.35	156.75
CC+VB	**7**	1.10E-02	8.03E-02	90.35	252.75

Table 6.2: Performance analyses for two transmit and four receive antennas.

2x6					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	5.00E-03	2.39E-01	15.73	37.48
CC+VB	**3**	7.32E-04	1.32E-02	35.45	86.95
CC+VB	**4**	7.53E-04	1.07E-02	39.45	98.95
CC+VB	**5**	7.74E-04	1.03E-02	47.45	122.95
CC+VB	**6**	7.00E-04	9.91E-03	63.45	170.95
CC+VB	**7**	6.69E-04	9.68E-03	95.45	266.95

Table 6.3: Performance analyses for two transmit and six receive antennas.

2x8					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	1.29E-03	8.08E-02	18.28	44.58
CC+VB	**3**	8.89E-05	2.02E-03	40.55	101.15
CC+VB	**4**	8.64E-05	1.99E-03	44.55	113.15
CC+VB	**5**	8.48E-05	1.87E-03	52.55	137.15
CC+VB	**6**	8.30E-05	1.72E-03	68.55	185.15
CC+VB	**7**	7.56E-05	1.65E-03	100.55	281.15

Table 6.4: Performance analyses for two transmit and eight receive antennas.

Figures 6.3, 6.4 and 6.5 show the BLER performance of the V-LS system alone and the V-LS system with convolutional coding with K=3 and 7.

From these figures we can observe two situations. First, the addition of the convolutional code in the transmission chain improves the BLER performance of the MIMO system, and increasing the constraint length of the convolutional code further improves its performance.

For instance, figure 6.3 shows that for a BLER of 10^{-2}, a gain of almost 5 dB is achieved by the K=3 convolutional code concatenated to the VBLAST system; and a further increase in 0.5 dB is attained when constraint length is increased from K=3 to K=7. Second, the performance features of the VBLAST system is preserved, i.e. the BLER performance is increased as the number of receive antennas is increased.

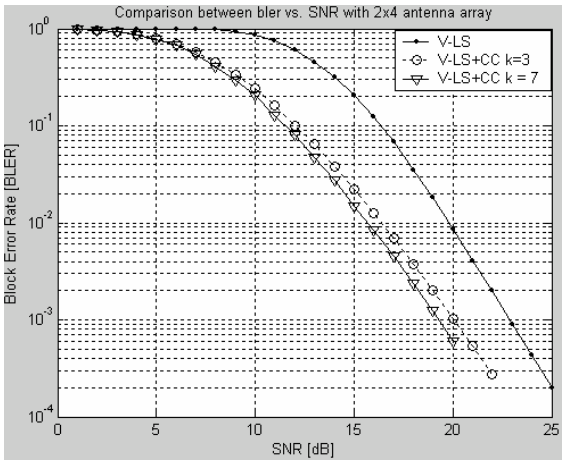


Figure 6.3: BLER vs SNR_{avg} M = 2; N = 4

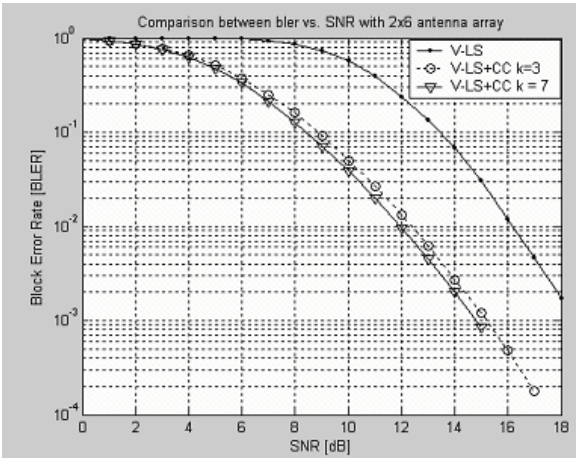


Figure 6.4: BLER vs SNR_{avg} M = 2; N = 6

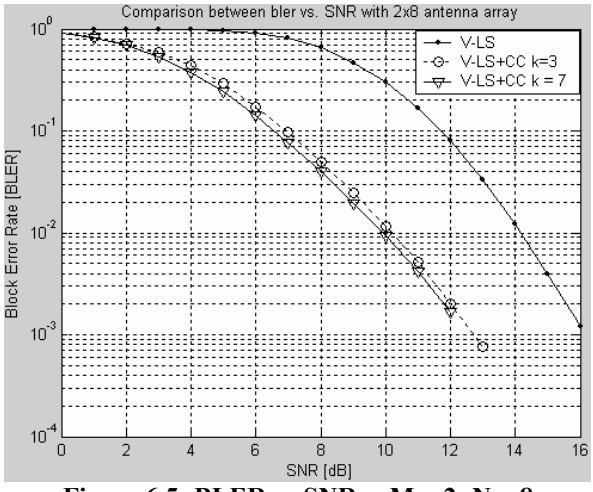


Figure 6.5: BLER vs SNR_{avg} M = 2; N = 8

The following part of the analysis is when the transmitter array is increased to eight transmit antennas, and the receive array has ten, twelve and fourteen receive antennas. In tables 6.5, 6.6 and 6.7, we will present a comparison between these systems for a SNR_{avg}

equal to 15dB. Clearly, as the constraint length is increased, the BLER performance is improved. Then, we can observe in figures 6.6, 6.7 and 6.8 that an increase in 1.5dB is obtained with a constraint length K=7. With a constraint length of five we can obtain a coding gain of almost 1 dB.

8x10					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	4.26E-02	9.75E-01	93.00	189.98
CC+VB	**3**	3.17E-02	6.46E-01	190.01	391.95
CC+VB	**4**	3.07E-02	5.82E-01	194.01	403.95
CC+VB	**5**	3.30E-02	5.40E-01	202.01	427.96
CC+VB	**6**	3.08E-02	4.82E-01	218.01	475.96
CC+VB	**7**	3.28E-02	4.49E-01	250.01	571.96

Table 6.5: Performance analyses for eight transmit and ten receive antennas.

8x12					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	1.39E-02	8.23E-01	97.05	200.68
CC+VB	**3**	6.06E-03	2.35E-01	198.10	413.35
CC+VB	**4**	5.61E-03	1.89E-01	202.10	425.35
CC+VB	**5**	5.15E-03	1.56E-01	210.10	449.35
CC+VB	**6**	4.30E-03	1.24E-01	226.10	497.35
CC+VB	**7**	4.05E-03	1.07E-01	258.10	593.35

Table 6.6: Performance analyses for eight transmit and twelve receive antennas.

8x14					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	4.74E-03	5.45E-01	101.10	211.38
CC+VB	**3**	1.23E-03	6.53E-02	206.20	434.75
CC+VB	**4**	1.07E-03	4.95E-02	210.20	446.75
CC+VB	**5**	8.74E-04	3.56E-02	218.20	470.75
CC+VB	**6**	7.50E-04	2.74E-02	234.20	518.75
CC+VB	**7**	6.07E-04	2.19E-02	266.20	614.75

Table 6.7: Performance analyses for eight transmit and fourteen receive antennas.

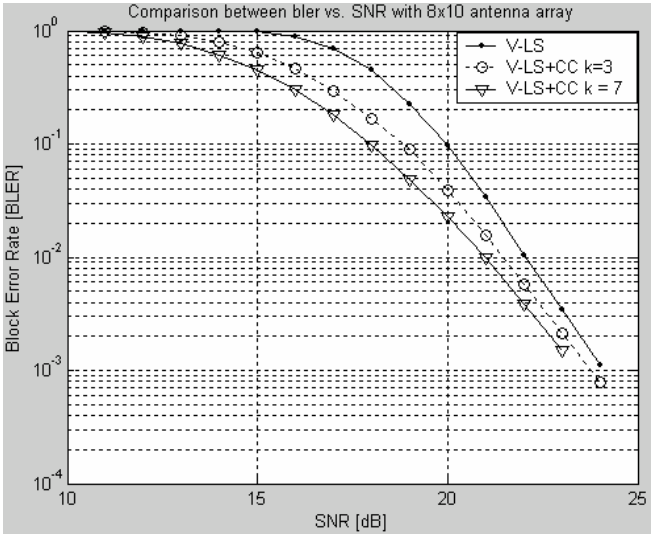


Figure 6.6: BLER vs SNR_{avg} M = 8; N = 10

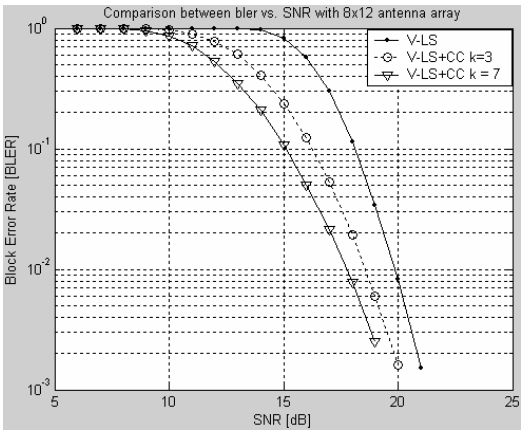


Figure 6.7: BLER vs SNR_{avg} M = 8; N = 12

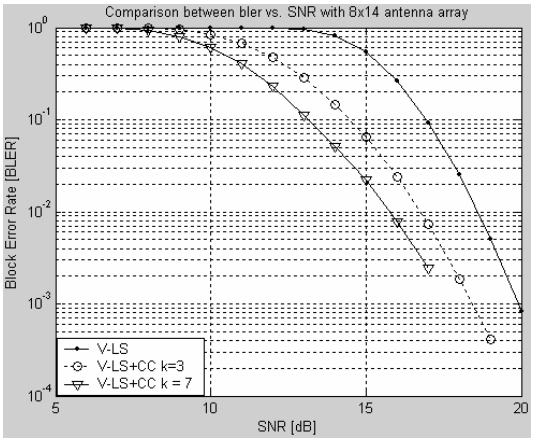


Figure 6.8: BLER vs SNR_{avg} M = 8; N = 14

Finally, the third part of the analysis includes a series of square antenna arrays. Table 6.8, 6.9, 6.10 and 6.11 shows comparison between algorithms with 2, 4, 6 and 8 pairs of antennas (both, transmit and receive) with a SNR_{avg} of 30dB.

2x2					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.19E-03	2.47E-02	10.63	23.28
CC+VB	**3**	2.00E-03	1.42E-02	25.25	58.55
CC+VB	**4**	1.99E-03	1.21E-02	29.25	70.55
CC+VB	**5**	2.31E-03	1.14E-02	37.25	94.55
CC+VB	**6**	2.50E-03	1.18E-02	53.25	142.55
CC+VB	**7**	2.62E-03	1.07E-02	85.25	238.55

Table 6.8: Performance analyses for two transmit and receive antennas.

4x4					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.14E-03	2.46E-02	26.50	55.08
CC+VB	**3**	2.15E-03	2.12E-02	57.00	122.15
CC+VB	**4**	2.08E-03	1.96E-02	61.00	134.15
CC+VB	**5**	2.14E-03	1.86E-02	69.00	158.15
CC+VB	**6**	2.24E-03	1.73E-02	85.00	206.15
CC+VB	**7**	2.49E-03	1.57E-02	117.00	302.15

Table 6.9: Performance analyses for four transmit and receive antennas.

6x6					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.31E-03	2.69E-02	51.68	104.68
CC+VB	**3**	2.38E-03	2.41E-02	107.35	221.35
CC+VB	**4**	2.46E-03	2.35E-02	111.35	233.35
CC+VB	**5**	2.30E-03	2.25E-02	119.35	257.35
CC+VB	**6**	2.33E-03	2.17E-02	135.35	305.35
CC+VB	**7**	2.56E-03	2.11E-02	167.35	401.35

Table 6.10: Performance analyses for six transmit and receive antennas.

8x8					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.17E-03	2.77E-02	88.95	179.28
CC+VB	**3**	2.36E-03	2.56E-02	181.90	370.55
CC+VB	**4**	2.36E-03	2.52E-02	185.90	382.55
CC+VB	**5**	2.40E-03	2.48E-02	193.90	406.55
CC+VB	**6**	2.31E-03	2.40E-02	209.90	454.55
CC+VB	**7**	2.39E-03	2.34E-02	241.90	550.55

Table 6.11: Performance analyses for eight transmit and receive antennas.

From these tables we can observe that for the same SNR_{avg} (30dB), we have that BLER is increased almost in 0.5dB when the constraint length in the channel coding is seven, when constraint is three, the channel gain is closer to 0.2dB for a square array system with two antennas. In figures 6.9, 6.10, 6.11 and 6.12 we can observe this phenomenon clearly. If we have a square system, we cannot obtained significant increasing in the BLER or the BER performance, basically, this phenomenon indicates that the convolutional coder will not bring us any advantage.

From figure 6.9 we can observe that the concatenated system with constraint length K=7 presents a code gain of 4dB with respect to V-LS system; however, if we increase a pair of antennas, the BLER will not increase, as compared to the non-concatenated system, i.e. no coding gain is obtained from the concatenated system.

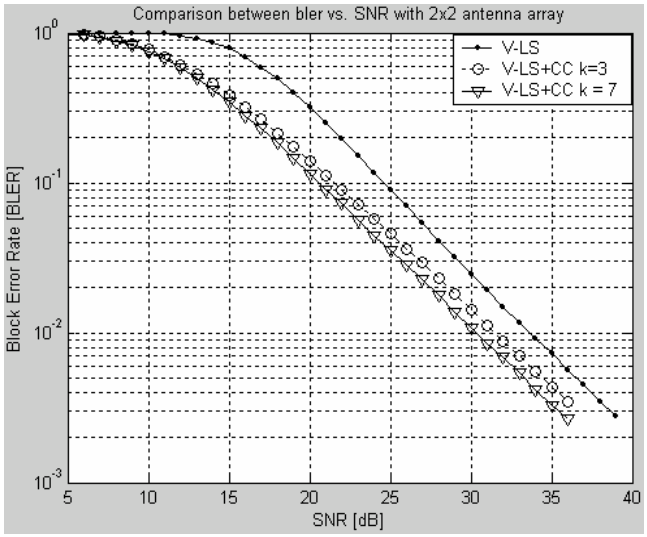


Figure 6.9: BLER vs SNR_{avg} M = 2; N = 2

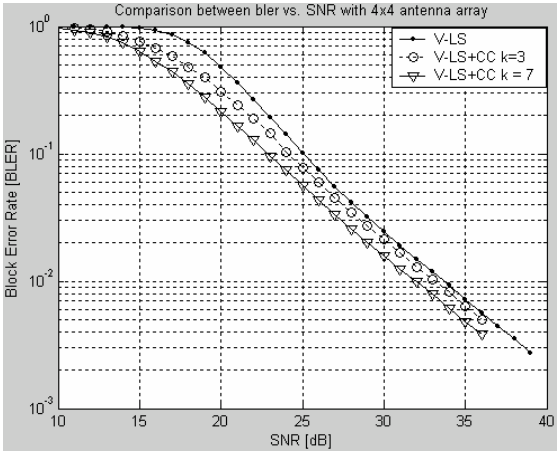


Figure 6.10: BLER vs SNR_{avg} M = 4; N = 4

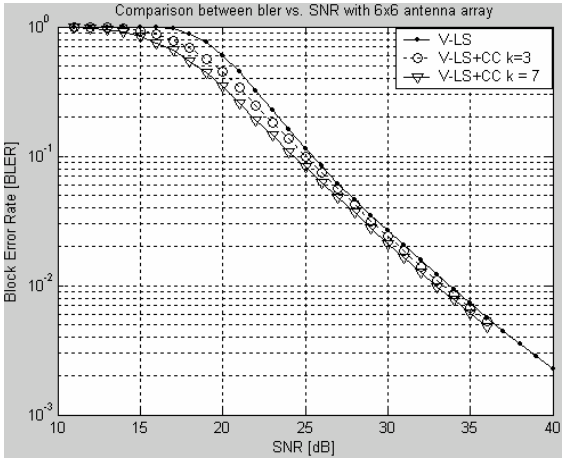


Figure 6.11: BLER vs SNR_{avg} M = 6; N = 6

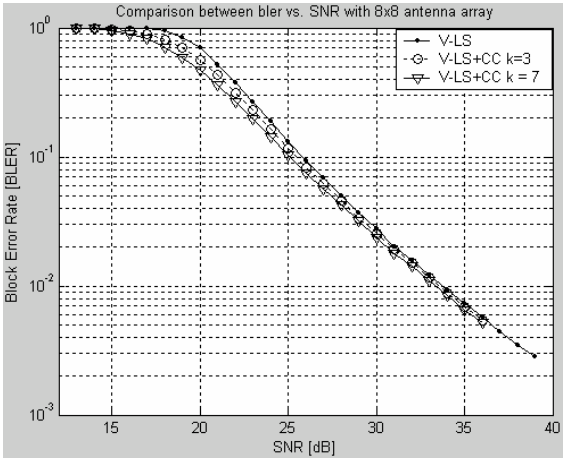


Figure 6.12: BLER vs SNR_{avg} M = 8; N = 8

6.3.1.2 Code Rate 1/3

In this section, the concatenated system consists of rate 1/3 convoltuional code and VBLAST ST coder with 2x4, 2x6 and 2x8 antenna arrays. Afterwards, the system will use

8x10, 8x12 and 8x14 antenna arrays. Table 6.12 illustrates the rate 1/3 convolutional codes used together with their generator polynomials for different constraint lengths.

Rate	Constraint Length	Free Distance	Code Vector
1/3	3	8	111 , 111, 101
1/3	4	10	1111, 1011, 1101
1/3	5	12	11111, 11011, 10101
1/3	6	13	101111, 110101, 111001
1/3	7	15	1001111, 1101101, 1010111

Table 6.12: Optimum Short Constraint Length Convolutional Codes for Rate 1/3.

Tables 6.13, 6.14 and 6.15 show us the first part of the analysis. From these tables we can observe the BLER performance for a SNR_{avg} of 7dB, and the best performance gain is given when K=7, as expected.

2x4					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	9.93E-02	9.93E-01	13.18	30.38
CC+VB	**3**	3.09E-02	2.23E-01	44.54	105.47
CC+VB	**4**	2.03E-02	1.77E-01	48.54	117.47
CC+VB	**5**	2.30E-02	1.66E-01	56.54	141.47
CC+VB	**6**	2.58E-02	1.57E-01	72.54	189.47
CC+VB	**7**	2.35E-02	1.53E-01	104.55	285.48

Table 6.13: Performance analyses for two transmit and four receive antennas.

2x6					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	5.31E-02	9.37E-01	15.73	37.48
CC+VB	**3**	4.74E-03	4.56E-02	52.39	127.31
CC+VB	**4**	2.39E-03	2.40E-02	56.39	139.31
CC+VB	**5**	2.43E-03	2.38E-02	64.39	163.31
CC+VB	**6**	2.92E-03	2.37E-02	80.39	211.31
CC+VB	**7**	2.35E-03	2.16E-02	112.39	307.31

Table 6.14: Performance analyses for two transmit and six receive antennas.

2x8					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.92E-02	8.16E-01	18.28	44.58
CC+VB	**3**	8.22E-04	9.37E-03	60.23	149.15
CC+VB	**4**	2.94E-04	3.60E-03	64.23	161.15
CC+VB	**5**	2.89E-04	6.19E-03	72.23	185.15
CC+VB	**6**	3.66E-04	3.62E-03	88.23	233.15
CC+VB	**7**	2.68E-04	3.31E-03	120.23	329.15

Table 6.15: Performance analyses for two transmit and eight receive antennas.

Figures 6.13 through 6.15 compare the V-LS system with and without convolutional coding. When convolutional coding is used, constraint length K=3 and K=7 are considered. In figure 6.13 we can see that a performance gain of 7 dB is attained with a constraint length of K=3, as compared to the V-LS system. Nevertheless, an increase in complexity by a factor of 5 has to be paid for this performance improvement. In figures 6.14 and 6.15, an improvement of 3.5 dB is attained for a complexity increase in a factor of 6

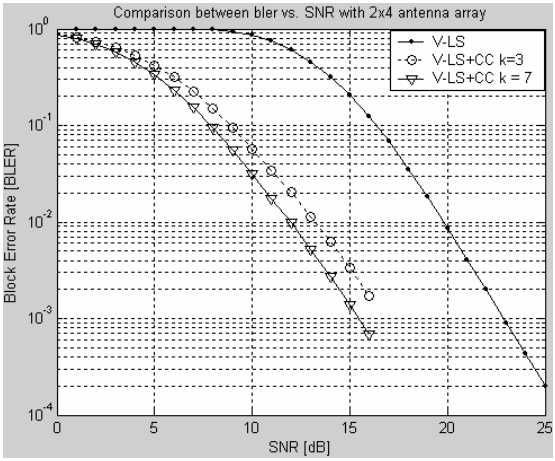


Figure 6.13: BLER vs SNR_{avg} M = 2; N = 4

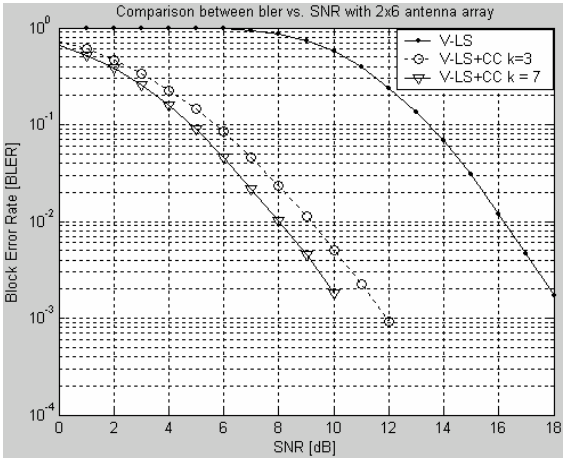


Figure 6.14: BLER vs SNR_{avg} M = 2; N = 6

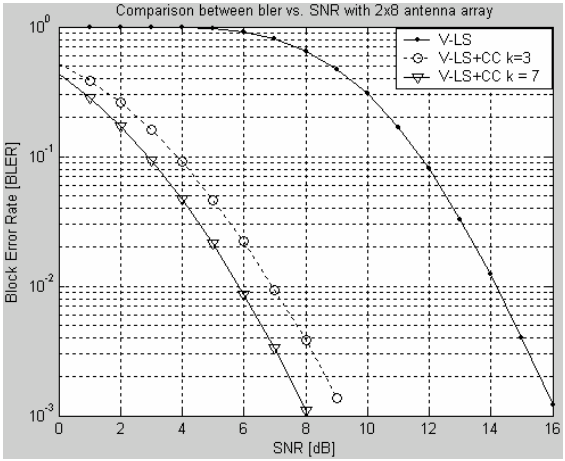


Figure 6.15: BLER vs SNR_{avg} M = 2; N = 8

Tables 6.16 through 6.18 present performance results for MIMO systems with 8x10, 8x12 and 8x14 antenna arrays. The signal to noise ratio considered is 14 dB. As expected, the BLER decreases as the constraint length of the convolutional code increases. Nevertheless, a penalty in computational complexity has to be accounted for.

8x10					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	6.29E-02	9.97E-01	93.00	189.98
CC+VB	**3**	2.03E-02	4.56E-01	284.76	585.51
CC+VB	**4**	1.40E-02	3.23E-01	288.76	597.51
CC+VB	**5**	1.17E-02	2.41E-01	296.76	621.51
CC+VB	**6**	9.32E-03	1.70E-01	312.76	669.51
CC+VB	**7**	8.33E-03	1.36E-01	344.76	765.51

Table 6.16: Performance analyses for eight transmit and ten receive antennas.

8x12					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	2.55E-02	9.63E-01	97.05	200.68
CC+VB	**3**	3.66E-03	1.37E-01	296.98	617.81
CC+VB	**4**	1.99E-03	7.22E-02	300.98	629.81
CC+VB	**5**	1.25E-03	4.24E-02	308.98	653.81
CC+VB	**6**	7.25E-04	2.21E-02	324.98	701.81
CC+VB	**7**	4.98E-04	1.29E-02	356.98	797.81

Table 6.17: Performance analyses for eight transmit and twelve receive antennas.

8x14					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-LS	**-**	1.04E-02	8.15E-01	101.10	211.38
CC+VB	**3**	6.60E-04	3.11E-02	309.21	650.11
CC+VB	**4**	3.02E-04	1.38E-02	313.21	662.11
CC+VB	**5**	1.47E-04	6.23E-03	321.21	686.11
CC+VB	**6**	6.55E-05	2.59E-03	337.21	734.11
CC+VB	**7**	3.45E-05	1.18E-03	369.21	830.11

Table 6.18: Performance analyses for eight transmit and fourteen receive antennas.

Figures 6.16, 6.17 and 6.18 illustrate the performance in terms of BLER of the V-LS system and the proposed system as a function of the received SNR. We can see in figure 6.16 that for a BLER of 10^{-2}, a gain of almost 2 dB is achieved with the K=3 V-LS+CC and 4 dB with K=7. Nevertheless, a complexity is increased of a factor of three for a constraint length of 3.

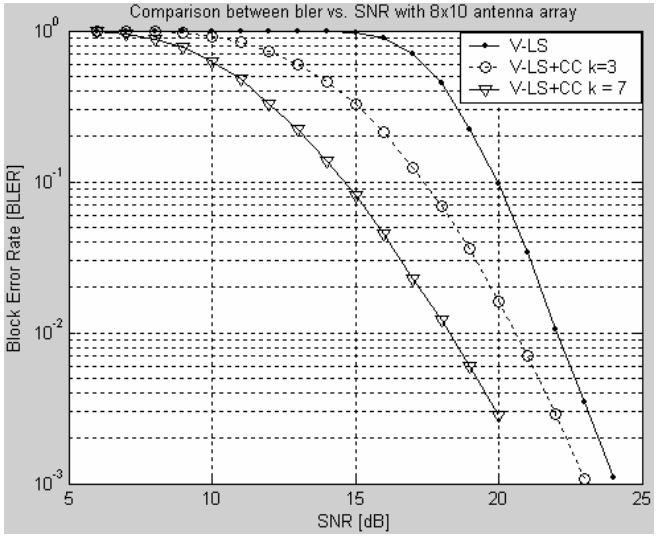


Figure 6.16: BLER vs SNR_{avg} M = 8; N = 10

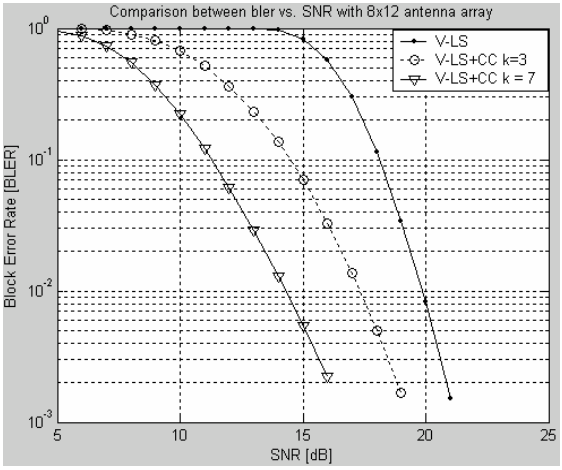


Figure 6.17: BLER vs SNR_{avg} M = 8; N = 12

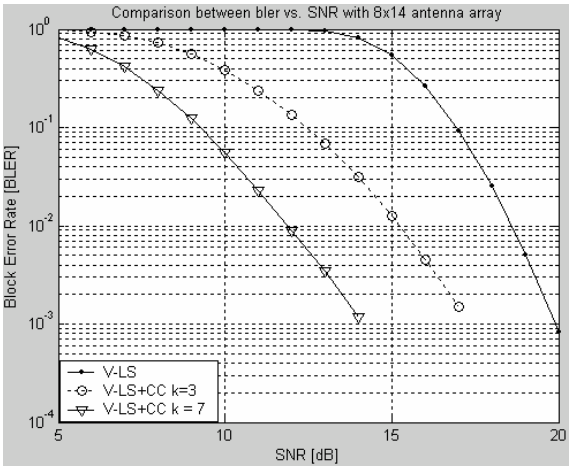


Figure 6.18: BLER vs SNR_{avg} M = 8; N = 14

Finally, a comparison of MIMO system for a 8x14 array reveals that a coding gain of 6 dB can be achieved with the V-LS+CC system with K=7 but with a cost size of complexity.

6.3.2 COMPUTATIONAL COMPLEXITY

Additional computational complexity is introduced when a channel coding (convolutional) is incorporated to the LS-VBLAST algorithm. As we showed, BLER performance can obtain better advantages, but the real question is how much complexity are we willing to accept.

Independent to the number of transmit and receive antennas, computational complexity of the Viterbi algorithm is dependent on the constraint length of the code. However, the overall complexity depends on both the number of transmit and receive antennas and the complexity of the Viterbi decoder.

As an example, we will take a system consisting of two transmit antennas and four receive antennas. The computational complexity is described below [16].

For Viterbi decoder, the storage requirements are mostly due to the survivor memory management. A rough estimation of the storage requirement is

$$Storage = L * 2^{K-1} \qquad 6.2$$

This is the storage requirement for register-exchange-like decoders [16]. Parameter L represents the number of bits that have to be stored in order to be able to make a correct decision on one decoded bit. Parameter K is the constraint length of the code.

In previous section it was shown that the best performance was obtained for K=7 coders. For this constraint length, 2^7 addition operations, 2^6 comparisons 2^7 memory addresses are needed. In addition, 2^K operations are needed for the branch metric calculation. This way, the Viterbi algorithm needs 384 additions per processing cycle to update the path metrics.

Table 6.19 and figure 6.19 shows how convolutional channel coding affects the computational complexity of the V-LS algorithm. For a constraint length of three and seven and for a code rate equal to 1/2, we have the lowest and the highest computational complexities, respectively.

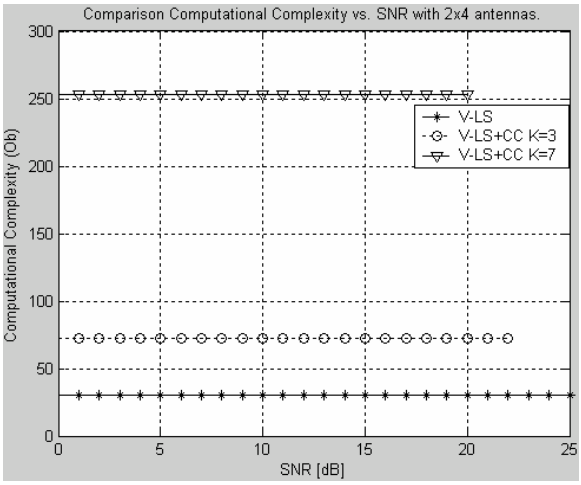


Figure 6.19: Computational Complexity vs SNR_{avg} M = 2; N = 4

2x4							
Algorithm	**K**	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-LS	**-**	13.18	10.95	9.30	0.34	0.04	30.38
CC+VB	**3**	30.35	33.90	18.60	0.68	0.08	72.75
CC+VB	**4**	34.35	45.90	18.60	0.68	0.08	84.75
CC+VB	**5**	42.35	69.90	18.60	0.68	0.08	108.75
CC+VB	**6**	58.35	117.90	18.60	0.68	0.08	156.75
CC+VB	**7**	90.35	213.90	18.60	0.68	0.08	252.75

Table 6.19: Performance analyses for two transmit and four receive antennas.

Figure 6.19 presents the computational complexity (arithmetic operations) as a function of the SNR_{avg}, for a 2x4 array. As we can see, computational complexity is strongly related to the constraint length of the convolutional code. In general, the computational complexity

depends on the number of transmit and receive antennas, on the number of symbols per block and on number of trellis states in the Viterbi algorithm until table is the complexity and storage requirements for a one bit.

6.4 COMPARISON WITH CP ALGORITHM

From past chapter we know that the best BLER performance was obtained with the CP algorithm; however, computational complexity was its main drawback when thinking of a hardware implementation. In this section, the proposed system (V-LS+CC) is compared to the CP algorithm from both performance and complexity point of views.

2x4					
Algorithm	K	BER	BLER	Storage	Ob
V-CP		5.22E-03	1.95E-01	61.77	63.71
V-LS	-	2.25E-02	6.08E-01	13.18	30.38
CC+VB	3	8.63E-03	1.01E-01	30.35	72.75
CC+VB	7	1.10E-02	8.03E-02	90.35	252.75
CC+VB	3	2.18E-03	2.02E-02	44.54	105.46
CC+VB	7	1.29E-03	9.95E-03	104.54	285.46

Table 6.20: Performance comparison: CP algorithm and new system for a 2x4 array.

2x4							
Algorithm	K	Storage	Add	Mul	Div	Srt	Ob
V-CP		61.77	26.58	15.74	2.63	0.10	63.71
V-LS	-	13.18	10.95	9.30	0.34	0.04	30.38
CC+VB	3	30.35	33.90	18.60	0.68	0.08	72.75
CC+VB	7	90.35	213.90	18.60	0.68	0.08	252.75
CC+VB	3	44.54	45.69	28.62	1.04	0.12	105.46
CC+VB	7	104.54	225.69	28.62	1.04	0.12	285.46

Table 6.21: Computational Complexity comparison: CP algorithm and new system for a 2x4 array.

Table 6.20 and 6.21 present the BLER performance and the computational complexity for a 2x4 array of antennas with SNR equal to 15dB and 14dB respectively. From tables we can observe that storage requirements are similar for CP and V-LS+CC algorithms with constraint equal K=7 (R=1/2) and K=3 (R=1/3), but the BLER performance obtained by the proposed system is better. Table 6.21 shows that the storage requirements and the computational complexity increases considerably for constraint lengths K=7 (either R=1/2 or R=1/3). In these cases, the computational complexity (arithmetic) of the proposed system is 2 and 4 times that of the CP algorithm.

Figures 6.20 and 6.21 present the results obtained by the comparison between the two algorithms (CP and V-LS+CC). The V-LS+CC system is shown with code rate of ½ and a constraint length of 3 and with a code rate of 1/3 and a constraint length of seven, i.e., the worst performance obtained by the new system and the best performance. This is done in order to visualize the BLER performance but also the increase in computational complexity.

From figures 6.20 and 6.21 we can observe that BLER performance is greatly improved for an acceptable increase in computational complexity. As a result, the proposed system can

be seen as a good tradeoff between performance and complexity as compared to the CP algorithm considering that the increase with respect a CP is more less only 10 percent.

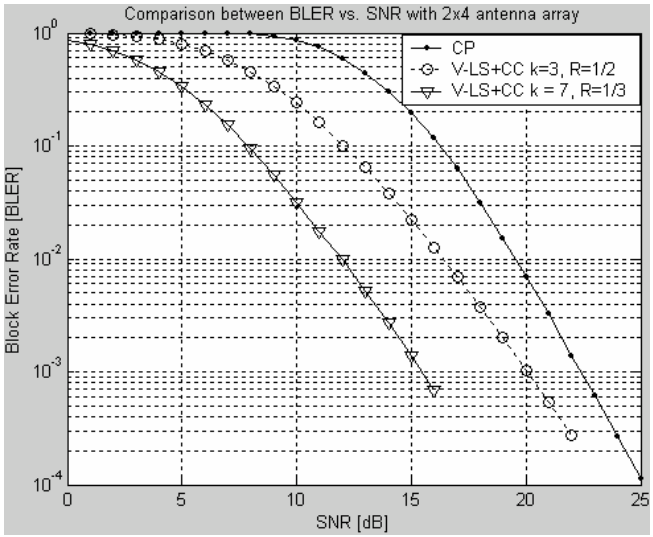


Figure 6.20: BLER vs SNR_{avg} Comparison: CP Algorithm and new system for M = 2; N = 4

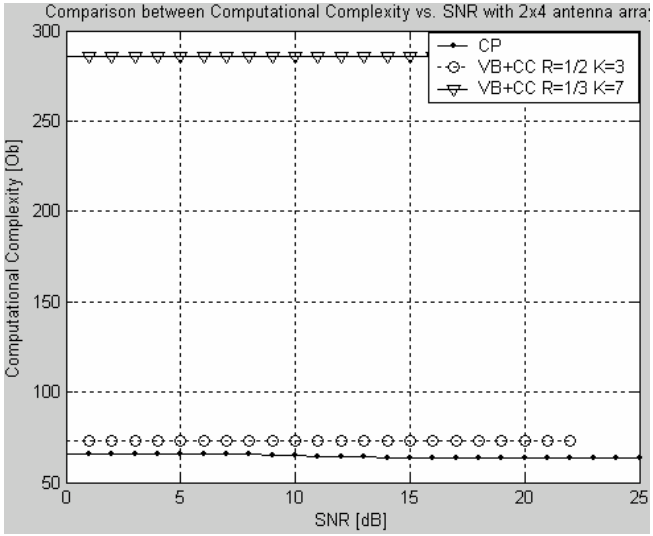


Figure 6.21: Computational Complexity vs SNR_{avg}: CP Algorithm and new system for M = 2; N = 4

In order to have a better idea of this comparison, we will analyze a 8x14 MIMO system. Tables 6.22 and 6.23 compare the proposed system and the systems analyzed in chapter 5 with a SNR of 14dB.

8x14					
Algorithm	**K**	**BER**	**BLER**	**Storage**	**Ob**
V-CP		1.28E-04	3.37E-02	414.21	476.45
V-LS	**-**	4.74E-03	5.45E-01	93.00	189.98
CC+VB	**3**	1.23E-03	6.53E-02	206.20	434.75
CC+VB	**7**	6.07E-04	2.19E-02	266.20	614.75
CC+VB	**3**	6.60E-04	3.11E-02	309.21	650.11
CC+VB	**7**	3.45E-05	1.18E-03	369.21	830.11

Table 6.22: Performance comparison: CP algorithm and new system for a 8x14 array.

8x14							
Algorithm	**K**	**Storage**	**Add**	**Mul**	**Div**	**Srt**	**Ob**
V-CP		414.21	169.84	143.78	9.33	0.10	476.45
V-LS	**-**	93.00	63.40	59.65	3.41	0.11	189.98
CC+VB	**3**	206.20	152.80	133.30	7.23	0.23	434.75
CC+VB	**7**	266.20	332.80	133.30	7.23	0.23	614.75
CC+VB	**3**	309.21	224.53	201.21	10.91	0.34	650.11
CC+VB	**7**	369.21	404.53	201.21	10.91	0.34	830.11

Table 6.23: Computational Complexity comparison: CP algorithm and new system for a 8x14 array.

We can see from figures 6.22 and 6.23 that either BLER performance is better with any of the V-LS with the Viterbi algorithm. However, the computational complexity is very similar when the code rate is ½, but is larger when the code rate is increased, no matter the constraint length. Do not forget that storage requirements and arithmetic operations needed for the CP algorithm decrease with increasing the SNR, but in the proposed system, the number of operations and storage requirements are independent of the SNR.

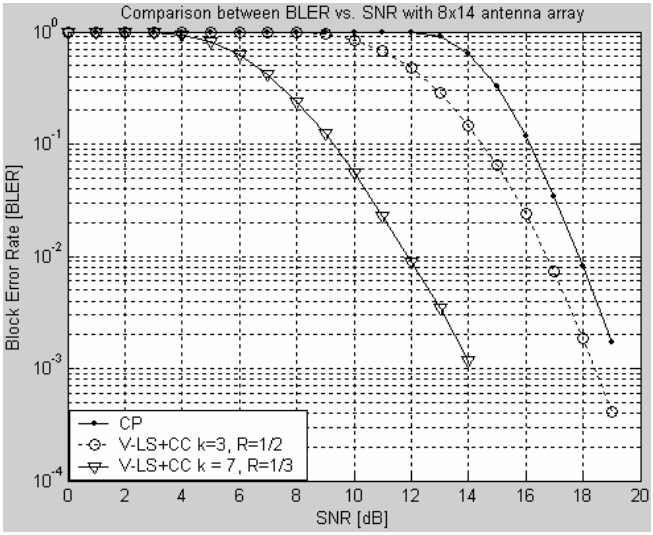


Figure 6.22: BLER vs SNR_{avg} : CP Algorithm and new system for M = 8; N = 14

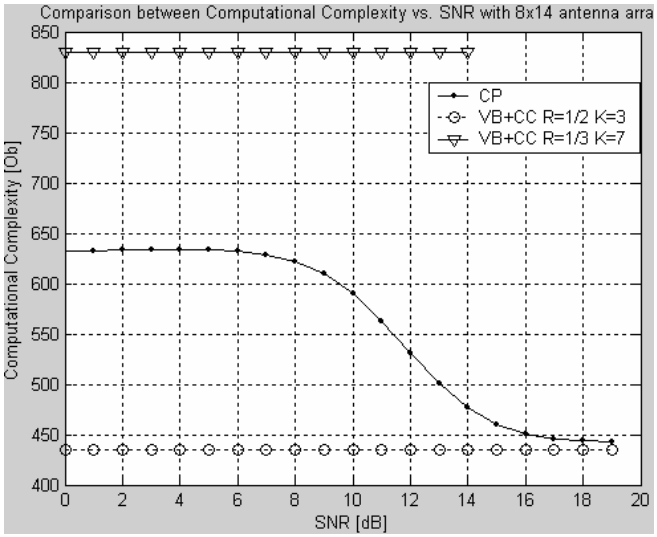


Figure 6.23: Computational Complexity vs SNR_{avg}: CP Algorithm and new system for M = 8; N = 14

6.5 CONCLUSIONS

In this chapter, convolutional channel coding is introduced to the LS-VBLAST algorithm. The performance and complexity have been analyzed and compared with different antenna arrangements in order to determine the algorithm best suited for a hardware implementation. From results we can observe several things. First, we can obtain improvements in the BLER performance when the constraint length of the convolutional code is increasing; however, the BER performance does not present the same pattern. An example of this situation can be seen in table 6.2, where we can observe that encoder with constraint lengths of three and four present better performance than encoders with constraint lengths equal to five, six and seven. We believe that this is due to the fact that no interleaving is used prior to demultiplexing the information stream coming out of the convolutional encoder. That may be the reason why small constraint length encoders exhibit better BER performance.

Another observation is the behavior of the system with respect to the constrain length. From theory, increasing the constraint length of the code in one increases the performance in ½ dB [17]. However, simulations results indicate that the unique systems which respects this is the code rate equal to ½ and eight transmit antennas systems.

The two previous statements take us to conclude that we need the introduction of an interleaver in the system. It is very probably that the independence needed by the VBLAST system is not complete only with the transmit matrix definition. Further research is to be done regarding this observation.

In addition, this chapter has been shown that the best BLER performance is obtained with a constraint length equal to seven, however, computational complexity grows exponentially with the number of states of the trellis.

Another results highlighted in this chapter is a detailed comparison between the CP algorithm and the conventional channel coding with VBLAST. We observe that BLER performance is always increased in the new system as compared to the CP algorithm; however, the complexity of the former grows exponentially with respect of the trellis states. Nevertheless, simulation results showed that computational complexity of the new system with a code rate of ½ essentially the same as the CP algorithm, for the same number for antenna arrangements, but the proposed system outperforms the CP algorithm in BLER performance.

REFERENCES

[1] G. J. Foschini, G. D. Golden, R. A. Valenzuela, P. W. Wolniansky, “Simplified Processing for High Spectral Efficiency Wireless Communications Employing Multielements Arrays”, IEEE J. Select. Areas Commun., Vol. 17, pp. 1847-1852, Nov. 1999.

[2] C. Langton, "Coding and decoding with Convolutional Codes", http://www.complextoreal.com/chapters/convo.pdf
[3] B. Aazhang, "Convolutional Codes", http://cnx.rice.edu/content/m10181/latest/
[4] C. Langton, "Coding Concepts and Block Coding", http://www.complextoreal.com/chapters/block.pdf
[5] R. Gaspa, J. R. Fonollosa, "Space-Time Coding for UMTS, Performance Evaluation in Combination with Convolutional and Turbo Coding", IST-1999-11729 METRA. 0-7803-6507-0/00 2000 IEEE.
[6] P. W. Wolniansky, G. J. Foschini, G. D. Golden, R. A. Valenzuela, "V-BLAST: An architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel", invited paper, Proc. ISSSE-98, Pisa, Italy, Sept. 29, 1998.
[7] K. Sosa, L. F. González, M. Bazdresch, J. R. Guisantes, "Performance Study of Space-time Communications Systems Base on the VBLAST algorithm", XV International conference on Electronics, Communications and Computers, CONIELECOM 2005.
[8] G. D. Forney Jr., "The Viterbi Algorithm", Proceedings of the IEEE, Vol. 61, pp. 268-278, March 1973.
[9] G. Forney, "Convolutional codes: I. Algebraic structure", IEEE Transactions on Information Theory, Vol. 16, pp. 720-738, November 1970.
[10] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", IEEE Transactions on Information Theory, Vol. IT-13, pp. 260-269. April 1967.
[11] C. E. Shannon, "Mathematical Theory of Communications", University of Illinois Press, 1963.
[12] K. J. Lansen, "Short Convolutional Codes with Maximal Free Distance for Rates ½, 1/3 and ¼", IEEE Transactions on Information Theory, Vol. IT-19, May 1973, pp. 371-372.
[13] S. Lin, D. J. Costello, Jr., "Error Control Coding: Fundamentals and Applications", Prentice Hall: Englewood Cliffs, NJ, 1983.
[14] B. Sklar, "Digital Communications: Fundamentals and Applications", Second Edition. PRT, 2001.
[15] D. S. Hirscheberg, "Fast Parallel Sorting Algorithms", Communications of the ACM, Vol. 21, No. 8, pp. 657-661, August 1978.
[16] L. González, "Thèse: Architectures VLSI pour le codage conjoint source canal en treillis", ENTS 2000.
[17] J. A. Heller, I. M. Jacobs, "Viterbi Decoding for Satellite and Space Communications". IEEE Trans. Commun. Technol., Vol. COM19, No. 5, October 1971, Fig. 7, p. 84.

7

CONCLUSIONS AND PERSPECTIVES

CONCLUSIONS

In this thesis, the potentials of MIMO systems in the context of high data rate wireless communications systems have been investigated. We saw that MIMO systems have the potential of increasing drastically the channel capacity and the data rates, as opposed to conventional SISO systems. This explains why space-time coding has been considered for standardization in the third generation mobile communications systems (3G) where the demand of high rate multimedia services is continuously increasing.

An important drawback of MIMO systems has to do with its computational complexity, which makes these systems very difficult to implement in hardware. In other words, implementing an algorithm not only has to do with its overall performance but also with its computational burden.

Of special interest, the Vertical Bell-Labs Space-Time Coding Architecture has proven its excellent complexity-performance tradeoff. That is, among the MIMO decoding procedures that exist, VBLAST is the algorithm that shortens the gap between theory and practice, i.e., the hardware implementation of these promising systems.

The VBLAST architecture has been analyzed with three different approaches which are related to the way the Moore Penrose pseudo inverse of the channel matrix **H** is performed. These approaches are Singular Value Decomposition VBLAST, Sorted QR Decomposition VBLAST and Least-Squares VBLAST. The analysis has been made from a performance and computational complexity point of view. In addition, these systems were compared to the closest point-based detection algorithm proposed by Agrell. It was shown that even though the CP algorithm presents the best performance in terms of block error rate, its complexity is considerably high as to be considered for a hardware implementation. From all the considered systems, the Least-Squares approach presents the best tradeoff. That is, this approach implies a performance degradation as compared to the CP algorithm; nevertheless, the computational complexity of the latter is much more important than the former.

In the last part of this work, and once the approach with the best tradeoff has been identified, a new MIMO system consisting in the concatenation of a conventional channel coder and this MIMO system (LS-VBLAST) is proposed. Using an off-the-shelf convolutional code with Viterbi decoding and with acceptable complexity (constraint lengths between K=3 and K=7), performance improvements of up to 7 dB were achieved. Hence, conventional channel coding techniques and V-LS algorithm may lead to the design of MIMO system with a very good tradeoff performance-complexity load.

In a more specific way, we studied the VBLAST algorithm with three variants; one of them, the V-LS presents the following characteristics:

- A BLER performance closer to the quasi-optimal CP algorithm with lower computational complexity for specific scenarios
- Reduction in the number of complex mathematical operations and a lower increase in the addition operations, which basically makes the algorithm easier to implement
- This algorithm allows a greater number of transmit and receive antennas
- Additional convolutional coding to the systems leads performance gains of 2dB with small antenna arrays and channel codes with R= ½ and up to 7dB for a R=1/3 code rate
- Computational complexity with the additional channel coding is still acceptable, specially for current technology target such as FPGA and/or DSPs

FUTURE WORK

There are a number of areas considered in this thesis where further work could be pursued:

- Hardware Implementation

    1. Obviously, a perspective for thesis is the hardware implementation in DSP (Digital Signal Processing) or Programmable Logic Device (PLD) platforms of the algorithm used here.
    2. A good idea for this thesis was to introduce another VBLAST variant in order to makes major number of complex mathematical computations into additions, which makes algorithm better for implementations.

- Improvement of the overall performance

    1. The improvement of the overall performance by introducing a channel space-time coding techniques, either trellis [1] of block [2] codes to the MIMO systems.
    2. Also, it can be introduced a coded-modulation technique [3] directly to the LS-VBLAST algorithm
    3. Adding the turbo principle to the detection process of the VBLAST algorithm

- Wireless standardization

    1. Introduce the VBLAST system to the wireless standardization IEEE802.16 or IEEE802.11n.
    2. Introduce to the CDMA systems as a fast search technique.

REFERENCE

[1] S. Baro, G. B. A. Hansmann, "Improved Codes for Space-Time Trellis Coded Modulation", IEEE Commun., Letters, Vol. 1, pp. 20-22, January 2000.

[2] S. H. Nam, C. Hwang, J. Chung, V. Tarokh, "Differential Space-Time Block Codes using QAM for Four Transmit Antennas", IEEE Communications Society, pp. 952-956, 2002.

[3] Z. Chen, B. Vucetic, J. Yuan, "Space-Time Trellis Codes with Antenna Selection", Electronics Letters, Vol. 39, No. 11, 29th May 2003.

APPENDIX A

OTHER MIMO CHANNEL CHARACTERISTICS

A.1 PERFORMANCE LIMITS OF MIMO SYSTEMS

A.1.1 CAPACITY MIMO FAST AND BLOCK RAYLEIGH FADING CHANNEL

In order to get an expression for capacity in MIMO fast and block Rayleigh fading channels [1], we have to define the coefficient $|h|^2$ as squared distributed random variable, with two degrees of freedom, denoted by x_2^2. This variable can be expressed as $y = x_2^2 = z_1^2 + z_2^2$, which was defined in past section. Therefore, the PDF is given by

$$p(y) = \frac{1}{2\sigma_r^2} e^{\frac{-y}{2\sigma_r^2}}, y \geq 0 \qquad \text{A.1}$$

The channel capacity can be obtained by estimating the mean value as follow

$$C = E\left[W \log_2\left(1 + x_2^2 \frac{P}{\sigma^2} \right) \right] \qquad \text{A.2}$$

Where E[.] denotes the expectation with respect to the random variable x.

To approximate capacity in MIMO with singular value decomposition, with the channel matrix H, we can represented by an equivalent channel, which consist in r ≤ min(N,M) decoupled parallel sub channels, where r is the rank of H. So, capacity is giving for the overall capacity

$$C = E\left[W \sum_{i=1}^{r} \log_2\left(1 + \lambda_i \frac{P}{N\sigma^2} \right) \right] \qquad \text{A.3}$$

where $\sqrt{\lambda_i}$ are the singular values of the channel matrix. Alternatively, by using an approximation, we can write for the mean MIMO capacity on fast fading channels

$$C = E\left[W \log_2\left[\det\left(I_r + \frac{P}{N\sigma^2} Q \right) \right] \right] \qquad \text{A.4}$$

Where Q is defined as

$$Q = \begin{cases} HH^H, N < M \\ H^H H, N \geq M \end{cases} \quad \text{A.5}$$

The channel capacity curve for receive diversity with maximum ratio combining (equation A.2) is shown in figure A.1.

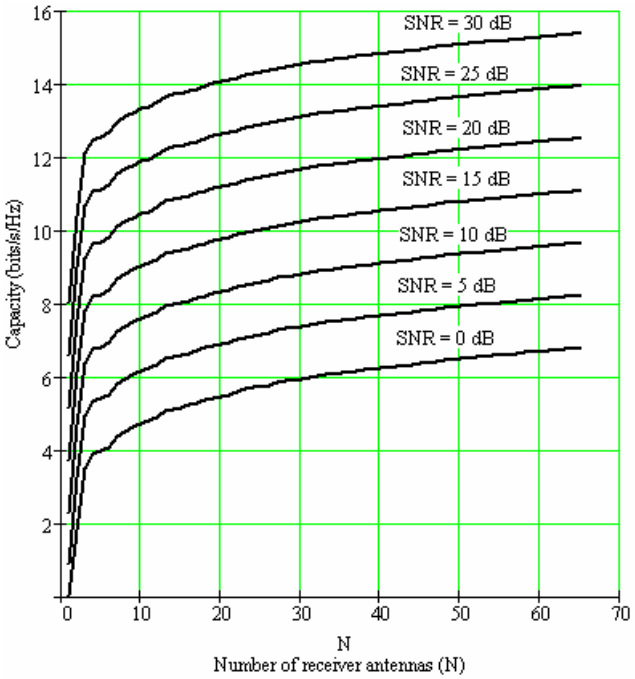


Figure A.1: Channel capacity curve for receive diversity on a fast and block Rayleigh fading channel with maximum ratio diversity combining.

Notice that, the diversity in the transmitter can remove the effect of fading for a large number of antennas, which is very important and the main objective of MIMO channels. It is important to remark that the capacity showed is for coordinated transmission, which means that all transmitter signals are the same and synchronous, wherever, for uncoordinated transmission, we can define the channel capacity by

$$C = E\left[W \log_2\left(1 + \frac{P}{N\sigma^2} x_{2M}^2\right)\right] \quad \text{A.6}$$

A.1.2 Capacity of MIMO Slow Rayleigh Fading channels

According to the channel, we will have Rayleigh distribution at the beginning of the transmission and held constant for a transmission block. This channel is the representation of LAN systems with high rates and low fade rates. The capacity estimated is random variable. For transmit diversity, the capacity has been defined by equation A.6, while receive diversity is given by

$$C = E\left[W \log_2\left(1 + \frac{P}{\sigma^2} x_{2N}^2\right)\right] \quad \text{A.7}$$

In a system with M transmitter and N receiver antennas, assuming that M ≥ N, we have that capacity can found with the lower and upper bound of the capacity, respectively, as follow

$$C > W \sum_{i=M-(N-1)}^{M} \log_2\left(1 + \frac{P}{M\sigma^2}\left(x_2^2\right)_i\right) \quad \text{A.8}$$

$$C < W \sum_{i=1}^{M} \log_2\left(1 + \frac{P}{M\sigma^2}\left(x_{2M}^2\right)_i\right) \quad \text{A.9}$$

where x is the squared random variable with 2N degrees of freedom. This system corresponds to an uncoupled parallel transmission system, which each of M antennas is received by a separated set of N antennas. When we have a squared system (means that receiver antennas is the same than transmitter), we can found capacity as lower bound

$$\frac{C}{WN} > \left(1 + \frac{\sigma^2}{P}\right)\log_2\left(1 + \frac{P}{\sigma^2}\right) - \log_2 e + \varepsilon_N \quad \text{A.10}$$

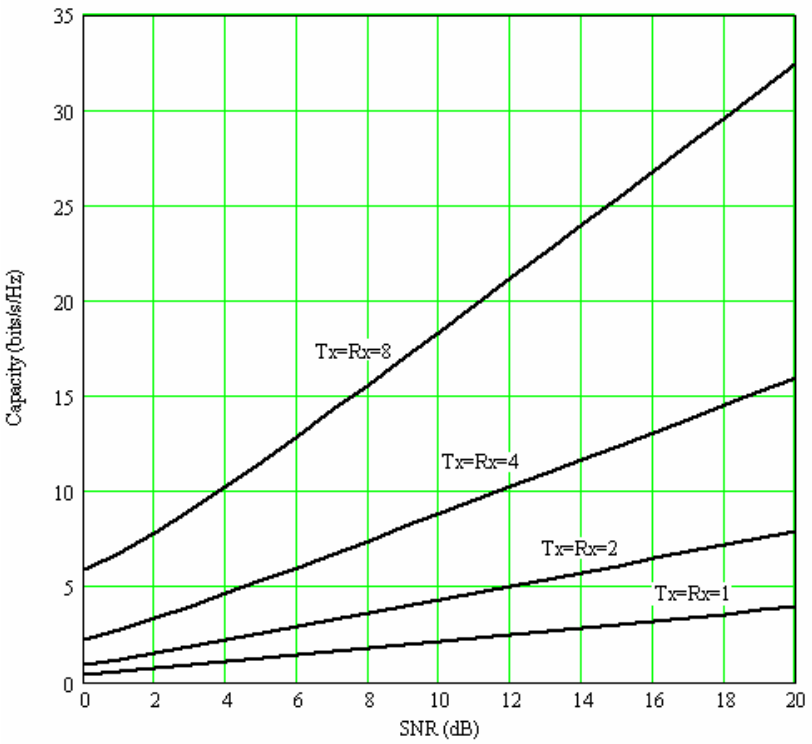


Figure A.2: Achievable capacity for a MIMO slow Rayleigh fading channel versus SNR for a variable number of transmit/receive antennas.

Where the random variable has a Gaussian distribution with mean

$$E[\varepsilon_N] = \frac{1}{N}\log_2\left(1+\frac{P}{\sigma^2}\right)^{-1/2} \quad \text{A.11}$$

In Figure A.2 is plotted for various numbers of antennas and a constant SNR. We can demonstrate that capacity achieves a given level improves markedly when the number of antennas increased. It shows that even with relative small number of antennas, large capacities can be available. As an example, at an SNR of 20dB and with 8 number of transmit/receive antennas, about 34 bits/s/Hz could be achieved while with only one antenna each side could achieve 4 bits/s/Hz.

A.2 EFFECT OF SYSTEM PARAMETERS AND ANTENNA CORRELATION

MIMO channels have been calculated its capacity under the idealistic assumption that the channel matrix entries are independent complex Gaussian variables, but in reality there is no true, so gain channel is diminished when the signals arriving are correlated. Correlated means that receiver antennas have a relationship between them, and it can be reduced by separating antennas spatially. In real channels, another degenerate propagation condition is the “keyholes”, which decrease channel capacity. The keyholes reduce the rank of the channel matrix and thus, lower the capacity.

In this section, we will introduce the correlation matrix models in MIMO systems in order to make them real. First able to define the correlation coefficients and introduce the correlation channels with a line of sight (LOS) path and in the absence of scattering. It is followed by a model for Rician fading channel. Subsequently, we will discuss about the keyhole effect and its influence on the channel matrix and capacity. Then, a quickly view in channel model for an outdoor channel with scattering, described by the system parameter such as the angular spread, scattering radio and the distance between the transmitter and receiver.

As we know, channel matrix can be represented by a vector h as follow

$$H = [h_1, h_2, ..., h_i, ..., h_N]^T \quad \text{A.12}$$

where each one of the h_x elements are vector from the receiver antenna. From here, we can define an NMxNM correlation matrix Θ given by

$$\Theta = E[h^H h] \quad \text{A.13}$$

where h^H denotes the Hermitian of h[3].

When entries of matrix H are independent identically distributed variables, then correlation matrix can generates a maximum capacity. From here, we assume that the correlation between receiver antenna elements do not depend on the transmitted antennas and vice versa. Which is real if we consider that immediate antenna surroundings causes the

correlation between array elements and have no impact on correlation observed between the elements of the array at the other end.

A correlated MIMO channel matrix can be resented as

$$Hc = K_R H K_T \tag{A.14}$$

where H is the matrix with uncorrelated complex Gaussian entries, K_R is an NxN matrix and K_T is an MxM matrix, given by their correlation matrices respectively which can be obtained by Cholesky Decomposition [2].

$$\begin{aligned} \Theta_R &= K_R K_R^H \\ \Theta_T &= K_T K_T^H \end{aligned} \tag{A.15}$$

A.2.1 Correlation Model for a Rayleigh MIMO Fading channel

Considering a linear array of M receiver antennas, spaced at a distance d_r, surrounded by clutter, similar to figure C.3. In figure, transmit antennas radiating signals which are reflected by the scatterers surrounding the receiver. The plane wave directions of arrival signals from the scatterers is α. The correlation coefficients can be calculated by the equation C.16

$$\theta_{mk} = \begin{cases} \int_{-\alpha_r/2}^{\alpha_r/2} e^{-j2\pi \frac{d_{mk}}{\lambda} \sin(\alpha)} p(\alpha) d\alpha, m \neq k \\ 1, m = k \end{cases} \tag{A.16}$$

where $p(\alpha)$ is the probability of the direction of arrival and α_r is the receiver antenna angular spread. For small values of the angle, we can obtain spread very large antenna separations to obtain low correlation. On the other hand, if the angle spread is reasonably large, low correlation can be obtained for antenna not higher than two wavelengths. For low element separation, the correlation coefficient is high even for large angle spreads. If there are pdf for the zero mean Gaussian distributed direction of arrival, then, we can denote this probability as

$$p(\alpha) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{\alpha^2}{2\sigma^2}}, -\frac{\alpha_r}{2} \leq \alpha \leq \frac{\alpha_r}{2} \\ 0, |\alpha| > \frac{\alpha_r}{2} \end{cases} \tag{A.17}$$

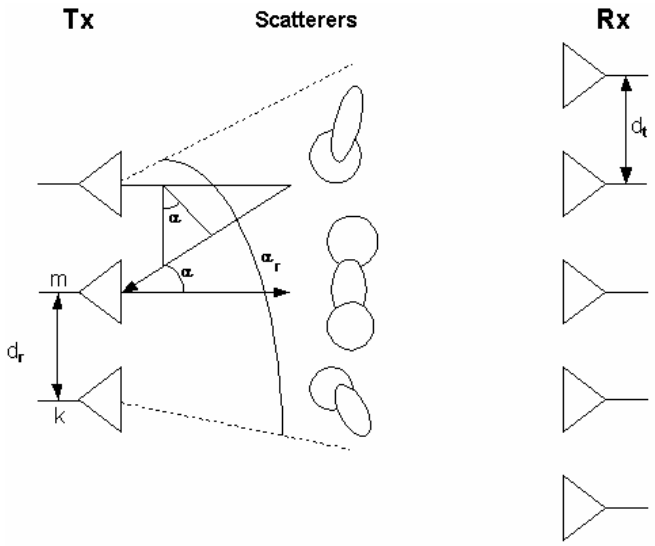


Figure A.3: Propagation model for a MIMO fading channel.

The standard deviation for the Gaussian distributed direction of arrival is calculated for a given angle spread α_r, and the expression is $\sigma = \alpha_r k$, where $k = 1/2\sqrt{3}$.

The capacity for a MIMO channel with this kind of correlation is given in equation C.18, where the correlation coefficients can be calculated by expression A.17, assuming a uniform distribution of the direction of arrival.

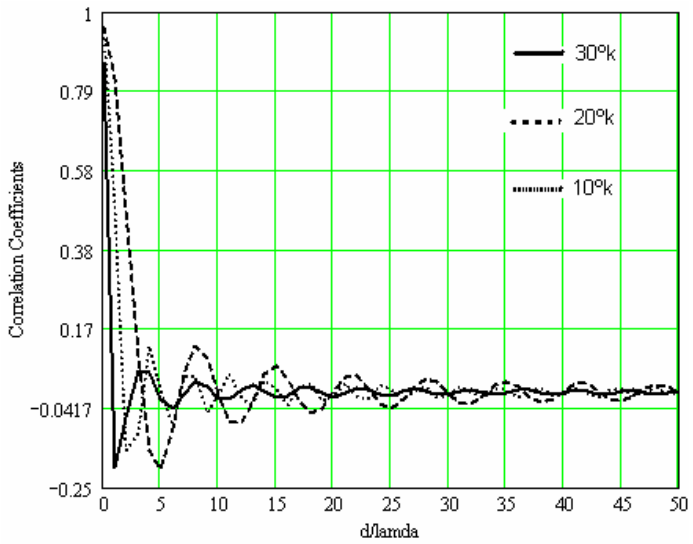


Figure A.4: Correlation coefficients in a fading MIMO channel with a uniformly distributed direction of arrival α.

Assuming that N = M, the capacity of a correlated MIMO fading channel can be expressed as

$$C = \log_2\left[\det\left(I_n + \frac{P}{N\sigma^2}\Theta_R^H H \Theta_T H^H\right)\right] \qquad \text{A.18}$$

From expression we can say that the number of transmit and receive antennas must be the same, which means that equation only can be applying in square systems. The correlation coefficients for a Gaussian distribution of the direction of arrival for the same spread as for the uniform distribution, denoted by expression A.16, is given in figure A.4.

A.2.2 Correlation Model for a Rician MIMO Channel

In order to understand the meaning of the expression, we must define the Rician model of channel as a system with LOS propagation and scattering. The model is characterized by the Rician factor, k, and it is the ratio of the line of sight and the scatter power component. The pdf for a Rician random variable *x* is given by

$$p(x) = 2x(1+k)e^{-k-(1+k)x^2} I_0\left(2x\sqrt{k(k+1)}\right), x \geq 0 \qquad \text{A.19}$$

where $k = \frac{D^2}{2\sigma_r^2}$, which superior and inferior level of equation are the powers of the LOS and scatterers components, respectively.

The matrix of the channel can be decomposed as equation A.20, where H_{LOS} is the channel matrix for the LOS propagation with no scattering and H_{RAYl} is the channel matrix for the scattering only.

$$H = DH_{LOS} + \sqrt{2}\sigma_r H_{RAYl} \qquad \text{A.20}$$

The equation can be interpreted as follow. For the Rician factor of zero the capacity is equal to the Rayleigh fading channel. As the Rician factor increase, the capacity reaches with the logarithmic expression. In LOS extreme case, the capacity is equal to the capacity of an uncorrelated Rayleigh fading channel. As Rician increase, then the capacity approaches the linear expression and increase.

A.3 SPACE-TIME CODING PERFORMANCE ANALYSIS AND CODE DESIGN

A.3.1 STATISTICAL MODELS FOR FADING CHANNELS

Statistical techniques are used to describe signal variations in a cellular mobile environment. It is important to say that all spectral components of the transmitted signal are subject to the same fading attenuation and occupy bandwidth smaller that the channel's coherent bandwidth, is called as frequency nonselective or frequency flat. On the other hand, if bandwidth is grater than the channel coherence, the spectral components of signal with a frequency separation larger that the coherence bandwidth are faded independently. The received signal becomes distorted, this phenomenon is known as frequency selective fading.

In this section, Rayleigh and Rician fading models are introduced to describe signal variations in narrowband multipath environment.

For a fast Rayleigh fading channel we consider a transmission of a single tone with constant amplitude, and assuming that direct wave is obstructed and the mobile unit received only reflected waves. When the number of waves is large, two quadrature components of the received signal are uncorrelated Gaussian random processes with a zero mean and variance σ_s^2. As a result, the envelope of the received signal at any time has a Rayleigh probability distribution and the phase has a uniform distribution between -π and π. The probability density function (pdf) of the Rayleigh distribution is given by

$$p(a) = \begin{cases} \frac{a}{\sigma^2} e^{-a^2/2\sigma_s^2}, a \geq 0 \\ 0, a < 0 \end{cases} \qquad \text{A.21}$$

The mean value, denoted by m_a, and the variance, denoted by σ_a^2, of the Rayleigh distributed random variable are given by

$$\begin{aligned} m_a &= \sqrt{\frac{\pi}{2}}\sigma_s = 1.25533\sigma_s \\ \sigma_a^2 &= \left(2 - \frac{\pi}{2}\right)\sigma_s^2 = 0.4292\sigma_s^2 \end{aligned} \qquad \text{A.22}$$

If the probability density function in A.21 is normalized so that the average signal power $\left(E\left[a^2\right]\right)$ is unity, then the normalized Rayleigh distribution becomes

$$p(a) = \begin{cases} 2ae^{-a^2}, a \geq 0 \\ 0, a < 0 \end{cases} \qquad \text{A.23}$$

The mean value and the variance are $m_a = 0.8862$ and variance $\sigma_s^2 = 0.2146$. The pdf for a normalized Rayleigh distribution is shown in figure A.5

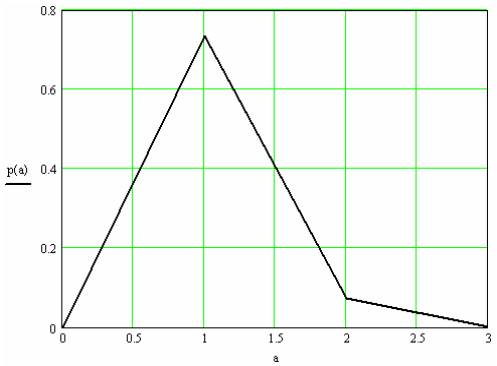


Figure A.5: The pdf of Rayleigh distribution normalized.

A.3.1.1 Rician Fading

In some scenarios, there are no obstacles on the line of sigh, but the received signal consists of a direct wave and a number of reflected waves. The direct one is a stationary nonfading signal with constant amplitude. The others one reflected are independent random signals. Their sum is called the scattered component of the received signal.

When number of reflected signal is large, the quadrature components of the scattered signal can be characterized as a Gaussian random process with a zero mean and variance σ_a^2. The envelope of the scattered component has a Rayleigh probably distribution.

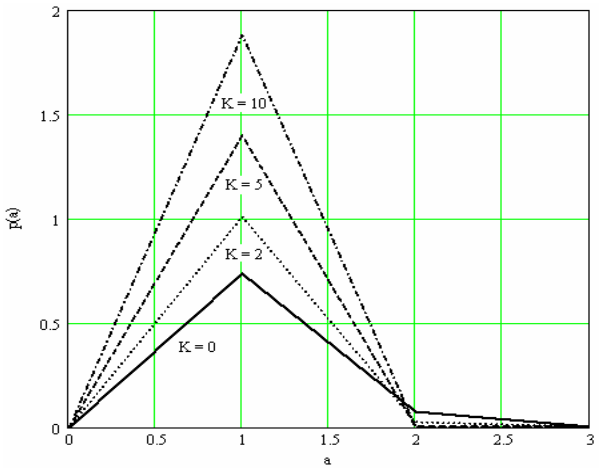


Figure A.6: The pdf of Rician normalized distributions with various K.

The sum of the constant amplitude direct signal and a Rayleigh distributed scattered signal results in a signal with a Rician envelope distribution. The pdf of the Rician distribution is given by

$$p(a) = \begin{cases} \dfrac{a}{\sigma_s^2} e^{\frac{-(a^2+D^2)}{2\sigma_s^2}} I_0\left(\dfrac{aD}{\sigma_s^2}\right), a \geq 0 \\ 0, a < 0 \end{cases} \qquad \text{A.24}$$

where D is the direct signal power and $I_0(\ )$ is the modified Bessel function of the first kind and zero-order. Assuming that the average power ratio is normalized to unity, the pdf becomes

$$p(a) = \begin{cases} 2a(1+K)e^{-K-(1+K)a^2} I_0\left(2a\sqrt{K(K+1)}\right), a \geq 0 \\ 0, a < 0 \end{cases} \qquad \text{A.25}$$

where K is the Rician factor, denoting the power ratio of the direct and scattered signal components. The Rician factor, the mean and the variance of the Rician distributed random variable are given by A.24 and represented in figure A.6

$$K = \frac{D^2}{2\sigma_s^2}$$
$$m_a = \frac{1}{2}\sqrt{\frac{\pi}{1+K}}\, e^{-\frac{K}{2}\left[(1+K)I_0\left(\frac{K}{2}\right)+KI_1\left(\frac{K}{2}\right)\right]} \qquad \text{A.26}$$
$$\sigma_s^2 = 1 - m_a^2$$

where $I_1(\,)$ is the first order modified Bessel function of the first kind.

These two models can be applied to describe received signal amplitude when the signal bandwidth is much smaller than the coherence bandwidth.

A.3.2 PERFORMANCE IN FADING CHANNELS

In order to illustrate the effects of multipath fading channels on error performance, we consider an uncoded system BPSK with and without multipath fading. The error probability of BPSK on AWGN channels with coherent detection is given by [3]

$$P_b(e) = Q\left(\sqrt{2\frac{E_b}{N_0}}\right) \qquad \text{A.27}$$

where E_b/N_0 is the ratio of the bit energy to the noise power spectral density. In fading channels, for a given fading attenuation *a*, which we consider that is constant each signal interval so that coherent detection can be achieved, the conditional bit error probability is given by

$$P_b(e) = \frac{1}{2}\left(1 - \sqrt{\frac{y_b}{1+y_b}}\right) \qquad \text{A.28}$$

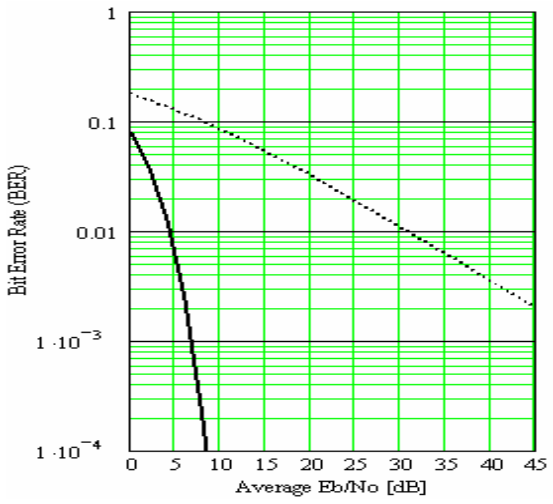


Figure A.7: BER performance comparison of coherent BPSK on AWGN and Rayleigh fading channels.

where $y_b = a^2 \frac{E_b}{N_0}$ is the received SNR per bit.

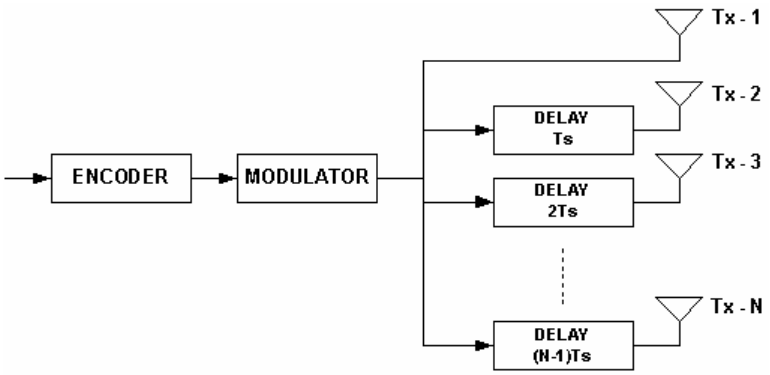
In order to compare the performance of the BPSK signaling, let plot the error probability of both (figure A.7). From figure we can observe that error rate decreases exponentially with the increasing SNR for a non-fading channel. However, for a fading channel, the error rate decreases inversely with the SNR. It is important to remark that, in order to appreciate the performance of the diversity techniques in combating the multipath fading, we consider a uncoded BPSK system with receive diversity.

A.3.3 TRANSMIT DIVERSITY

In contrast to receiver diversity, transmit diversity has received little attention, and the consequence is more difficult to exploit. The difficulties include that transmit signals from multiple antennas are mixed spatially before they arrive at the receiver, some additional signal processing is required at both, the transmitter and receiver, in order to separate the received signal and exploit diversity. And additional, transmitter does not posses any channel information.

However, transmit diversity can increase the channel capacity considerably. There are two transmit diversity schemes: with and without feedback. The principal difference between them is that the former relies on the channel information at the transmitter, which is obtained via feedback channels, while the latter does not require any channel information.

For transmit diversity with feedback, modulated signals are transmitted from multiple transmit antennas with different weighting factors, this factor are chosen adaptively so that the received signal power is maximized. In practical cellular mobile systems, mobility and environment changes quickly, making channel estimation and tracking difficult. The imperfect channel variations and mismatch between the channel state and current condition will decrease the received signal SNR and affect the system performance.



FigureA.8: Delay transmit diversity scheme.

For transmit diversity without feedback, messages are usually processed and then sent from multiples antennas. The signal processing is designed to enable the receiver exploiting the diversity from the received signals. At the receiver, the messages are recovered by using

signal detection technique. An example of this system is shown in figure A.8, which information is transmitted by multiple antennas. At the receiver side, the delays of the second up to the N transmit antennas introduce a multipath distortion for the signal transmitted from the first antenna. The distortion can be resolved at the receiver using maximum likelihood (ML) to obtain a diversity gain.

In order to show the performance, we consider a N transmit diversity system with a single receive antenna and no feedback. The average bit error probability of this scheme with BPSK modulation on Rayleigh fading channel is given by

$$P_b(e) = \left[\frac{1}{2}\left(1-\sqrt{\frac{y}{1+y}}\right)\right]^N \sum_{k=0}^{N-1}\binom{N-1+k}{k}\left[\frac{1}{2}\left(1+\sqrt{\frac{y}{1+y}}\right)\right]^k \qquad \text{A.29}$$

where $y = \frac{1}{N}\cdot\frac{E_b}{N_0}$.

In figure A.9, bit error rate performance of the scheme against noise ratio for various numbers of antennas. We can observe that when transmit diversity order is increased from one to two, the performance curve suggest that transmit diversity can improve only by less than 1 dB. For a large number of diversity branches, fading channel converges towards an AWGN channel, and curve almost approaches the one for the AWGN channel.

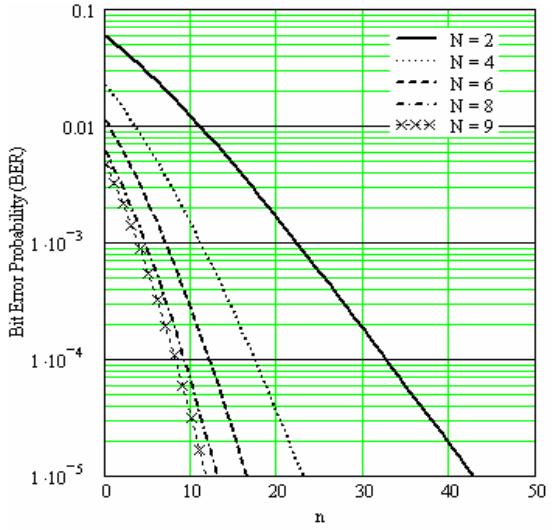


Figure A.9: BER performance of BPSK on Rayleigh fading channels with transmit diversity and various numbers of antennas.

A.3.4 ERROR PROBABILITY ON SLOW FADING CHANNELS

On slow fading channels, fading coefficients are constant. Let us define a codeword difference as the difference between one codeword to other. We can construct an NxN codeword distance matrix as

$$A(X,Y) = B(X,Y) \cdot B^H(X,Y) \quad \text{A.30}$$

where H denotes the Hermitian (transpose conjugate) of a matrix. It is clear that the matrix A is a non-negative real numbers [4]. Therefore, there existing a unitary matrix V and a real diagonal matrix Δ such that

$$VA(X,Y)V^H = \Delta \quad \text{A.31}$$

The rows of V are the eigenvectors of A, forming a complete orthonormal basis of a N vector space. The diagonal elements of matrix Δ are the eigenvalues of A, so the matrix can be represented by equation A.32, remembering that for simplicity, all arguments in diagonal matrix are major than zero.

$$\Delta = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix} \quad \text{A.32}$$

So, equation A.31 can be rewritten as

$$d_h^2(X,Y) = \sum_{j=1}^{N} h_j A(X,Y) h_j^H = \sum_{j=1}^{N}\sum_{i=1}^{N} \lambda_i \left|\beta_{j,i}\right|^2 \quad \text{A.33}$$

where $\beta_{j,i} = h_j \cdot v_i$ and . denotes the inner product of complex vectors. We can define upper bound on the conditional pairwise error probability expressed as a function of $\left|\beta_{j,i}\right|$.

$$P(X,Y \mid H) \le \frac{1}{2}\exp\left(-\sum_{j=1}^{N}\sum_{i=1}^{N} \lambda_i \left|\beta_{j,i}\right|^2 \frac{E_s}{4N_0}\right) \quad \text{A.34}$$

REFERENCES

[1] G. D. Durgin, ”Space-Time Wireless Channels”, Prentice Hall PTR. 2003, pp. 234 – 255.

[2] B. Vucetic, J. Yuan, “Space-Time Coding”. WILEY. 2004, pp. 1 – 90.

[3] V. Tarokh, N. Seshadri and A.R. Calderbank, “Space-time codes for high data rate wireless communications: performance criterion and code construction”. IEEE Trans. Inform. Theory, vol. 44, no. 2, pp. 744-765. Mar. 1998.

[4] J. G. Proakis, “Digital Communications”, 4th Ed., Mc. Graw-Hill, New York, 2001.

[5] www.mathcad.com

APPENDIXE B

DIVERSITY

B.1 Diversity Techniques

Diversity is used to reduce the multipath effects and improve the reliability of transmission without increasing the transmitted power or sacrificing the bandwidth. This technique requires multiples replicas of the signal at the receiver, same information but with small correlation. The basic idea is that two or more independent samples of a signal are taken, these will fade in a uncorrelated manner, in the way that some samples are severely faded while other are less attenuated, which will make that the probability of all the samples being simultaneously below a given level is lower than the probability of any individual sample. Thus, combinations of the various samples results in greatly reduced severity of fading and improved reliability of transmission.

Diversity techniques are used in mobile communications with a diversity of methods in order to get better performance. According to the domain where diversity is introduced, the techniques are classified into time, frequency and space.

B.1.1 TIME DIVERSITY

Time diversity can be achieved by transmitting identical messages in different time slots, which results in uncorrelated fading signals at the receiver. The time separation required is at least the coherence time of the channel, which is the same that the reciprocal of the fade rate $1/f_d = c/vf_c$. Definition of coherence time is the statistical measure of the period of time over which the channel fading if correlated. It is important to remark that in wireless communications, time diversity can be achieved by interleaving techniques, which means that the replicas of the transmitted signals are provided to the receiver in the form of redundancy in the time domain, the time separation between replicas is provided by the interleaving to obtain independent fades at the input of the receiver. This technique is very useful in fast fading environments where coherence time is small.

B.1.2 FREQUENCY DIVERSITY

A number of different frequencies are used to transmit the same message in frequency diversity. The separation must be enough to ensure independence fading associated with each frequency, this separation will guarantee that the fading statistics for different frequencies are essentially uncorrelated. The coherence bandwidth is different for different propagation environments. In mobile communications, the replicas of the transmitted

signals are usually provided to the receiver in form of redundancy in the frequency domain introduced by spread spectrum. Like time diversity, frequency diversity introduces a loss of bandwidth efficiency due to a redundancy introduced in the frequency domain.

B.1.3 SPACE DIVERSITY

This technique is also called as antenna diversity. It is implemented using multiple antennas arrays in both, transmitter and receiver. The multiple antennas are separated physically by a proper distance so that the individual signals are uncorrelated. The separations requirements vary with antenna height, propagation environment and frequency. The separation must be enough with a few wavelengths to obtain uncorrelated signals. Space diversity does not introduce any loss in bandwidth efficiency, which makes them very attractive for wireless communications. In this work, we will not make any difference with the physical implementations, because all study will be oriented to the channel coding and the performance of this technique. However, polarization and angle diversity are two examples of space diversity. In polarization diversity signals are transmitted by two different polarized antennas and received by two different polarized antennas, which ensure that the two signals are uncorrelated without having to place the two antennas far apart. Angle diversity is applied for transmission with carrier frequency larger than 10GHz. The receiver signals from different directions are independent to each other.

We can classify multiple antennas in two categories: receiver diversity and transmit diversity. First one is used at the receiver site to pick up independent copies of the transmit signal, which is used to obtain the replicas properly combined in order to increase the overall received SNR and mitigate multipath fading. In transmit diversity, messages are processed at the transmitter and then spread across multiples antennas.

B.2 DIVERSITY COMBINING METHODS

In general, the diversity techniques depends of how multiples signals are combined at the receiver, therefore, diversity schemes can also be classified into combined methods. According to the implementation complexity and the channel state information required, there are four main types of combining techniques: selection combining, switched combining, equal-gain combining (EGC) and maximal ratio combining (MRC), which will be explain as follow.

B.2.1 SELECTION COMBINING

It is a simple technique, which consider a receiver diversity system with N receive antennas. The signal with the largest instantaneous signal to noise ratio (SNR) at every symbol interval is selected as the output, so that the output SNR is equal to best incoming signal (figure B.1). In practice, the signal with the highest sum of the signal and noise power is usually used.

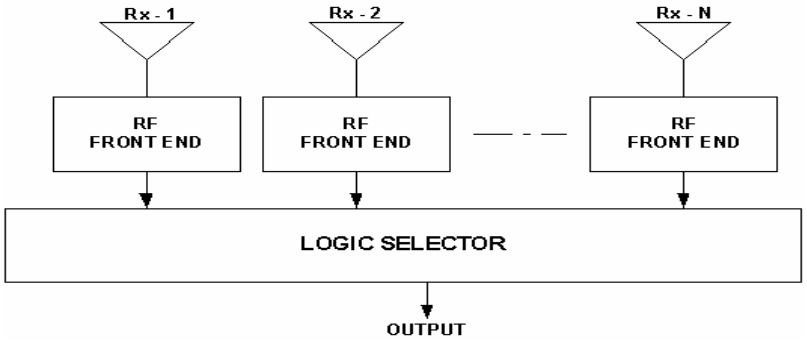


Figure B.1: Selection Combining Method.

B.2.2 SWITCHED COMBINING

The receiver scans all the diversity branches and selects a particular one with the SNR above a certain predetermined threshold. The signal is selected as the output, until SNR drops below the threshold. When this happens, the receiver stars scanning again and switches to another branch. It is also called scanning diversity.

Compared to selection diversity, this method is inferior because is not continually pick up the best instantaneous signal. Figure B.2 shows diagram of switched combining diversity.

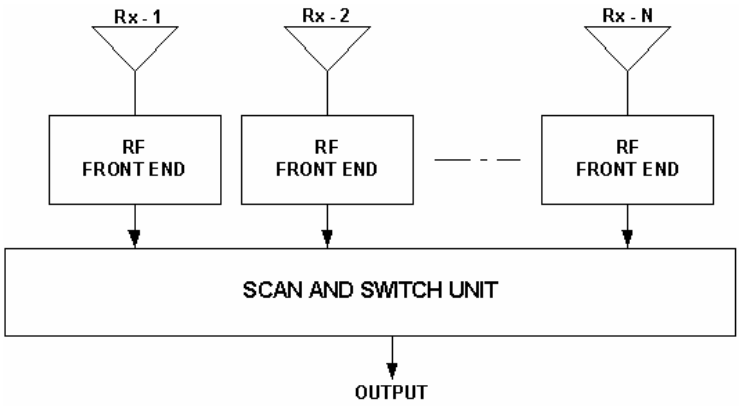


Figure B.2: Switched Combining Method.

B.2.3 Maximal Ratio Combining (MCR)

This method is linear, which means that various signal inputs are individually weighted and added together in order to get an output signal. The weighting factors can be chosen in different ways. The figure D.3 shows us a diagram of this method. The output signal is a linear (add) of a weighted replica of all the received signals. The output is given by equation D.1

$$r = \sum_{i=1}^{N} \alpha_i \cdot r_i \qquad \text{B.1}$$

where r_i is the received signal at antenna I, and the other element, α_I is the weight factor for the antenna. In maximal ratio combining, the weighting factor is chosen in proportion to its own signal voltage to noise power ratio (SNR). The weighting factor when each antenna has same average to noise power can be described in equation D.2

$$\alpha_i = A_i e^{-j\phi_i} \qquad \text{B.2}$$

where A_i and ϕ_i are the amplitude and phase of the received signal respectively. The method is called optimum combining since it can maximize the output SNR. In this scheme, each individual signal must be co-phased, weighted with its corresponding amplitude and then summed. Of course, the method needs to know channel's fading amplitude and signal phases.

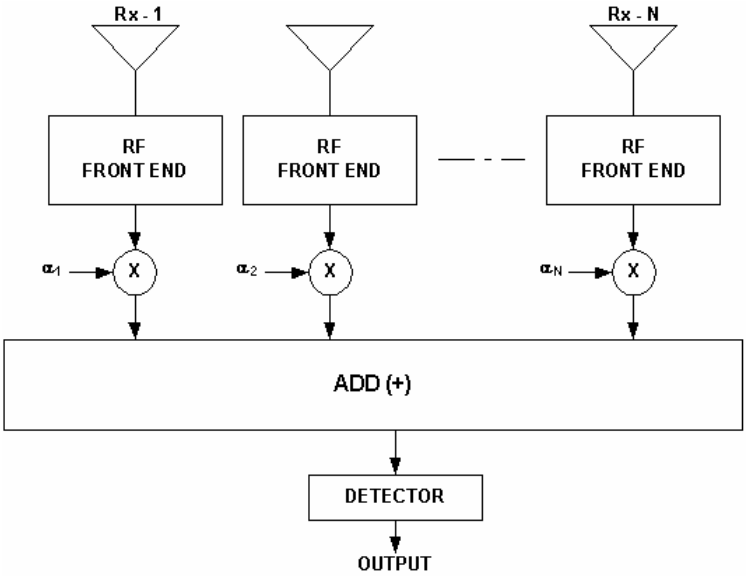


Figure B.3: Maximum Ratio Combining Method.

B.2.4 Equal Gain Combining

Equal gain combining is a suboptimal linear method, which does not require estimation of the fading amplitude for each individual branch. Instead, the receiver can calculate the weighting factor to be unity as follow

$$\alpha_i = e^{-j\phi_i} \qquad \text{B.3}$$

All the receiver signals are co-phased and added together with equal gain. The performance of this method is marginally inferior to maximum ratio combining. Its implementation is significantly less than any other method.

APPENDIX C

INTRODUCTION TO MATRIX COMPUTATIONS.

All the analysis done in this and next chapter are based in array processing area utilizes vectors and matrices extensively. In this section, we have summarized most important definitions and properties that will be helpful to the next analysis [31]. In general, all vectors and matrix are assumed to be complex, because in communications systems, all systems which provided of channel are complex. The labeling of the matrices: *A, B, C, ...*, and vectors *a, b, c, ...,* and there is not any physical significance unless specifically indicated.

C.1 Basic Definitions

We define an NxM matrix A by defining its elements as $a_{i,k}$, i=1, 2, …, N, k=1, 2, …, M. We write it in matrix form with N rows and M columns, but we can refer to matrix as $[A]_{i,j}$ for the *ij-th* element.

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,M} \\ a_{2,1} & a_{2,2} & \dots & a_{2,M} \\ \dots & \dots & \dots & \dots \\ a_{N,1} & a_{N,2} & \dots & a_{N,M} \end{bmatrix} \qquad \text{C.1}$$

The notation used to define a vector is given by equation C.2, and is an Nx1 matrix. We can also refer to it as a column vector.

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \dots \\ a_N \end{bmatrix} \qquad \text{C.2}$$

The product of two matrices, A and B, which need dimensions NxM and MxL respectively, give as a result a new matrix C with dimensions NxL, whose elements are given by

$$c_{i,j} = \sum_{k=1}^{M} a_{i,k} b_{k,j} \qquad \text{C.3}$$

However, it is important to remark that matrix product is only defined if and only if the number of columns in the first matrix is equal to the number of rows in the second.

Transpose of a matrix A is denoted by the superscript T, and consist in reordering rows to columns as follow

$$\left[A^T\right]_{i,j} = \left[A\right]_{j,i} = a_{j,i} \tag{C.4}$$

The transpose has the property of $(AB)^T = B^T A^T$.

Hermitian matrix is a square matrix A with elements that have complex conjugate symmetry $A^H = A$ which implies that $a_{ij} = a_{ji}^*$. The diagonal elements are real. Four elements matrix example is given by

$$A^H = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12}^* & a_{22} & a_{23} & a_{24} \\ a_{13}^* & a_{23}^* & a_{33} & a_{34} \\ a_{14}^* & a_{24}^* & a_{34}^* & a_{44} \end{bmatrix} \tag{C.5}$$

The inner product of two vectors is defined by equation $\alpha = x^H y$, when $x = y$, α corresponds to the square of the Euclidean norm of the vector $\alpha = x^H x$.

The outer product of complex vector x and a complex vector y is a matrix A with dimension NxM, $A = xy^H$. When $x = y$, A is a square matrix with is call the Hermitian of arbitrary vector x, $A = xx^H$.

The trace of a square matrix is denoted by tr(A) and is the sum of the diagonal elements, denoted by equation A.6, and has the distributive and associative properties.

$$tr(A) = \sum_{i=1}^{N} a_{ii} \tag{C.6}$$

The rank of a matrix is the number of linearly independent columns of rows. The rank of a matrix has the following properties

$$\begin{aligned} &rank(A+B) \le rank(A) + rank(B) \\ &rank(AB) \le \min(rank(A), rank(B)) \end{aligned} \tag{C.7}$$

C.2 Toeplitz Matrices

A Toeplitz matrix has the property that all the elements along each diagonal are identical, an example of this type of matrix is given in equation A.8. If A is a square matrix, then it is a special case of the persymmetric matrix. If A is also Hermitian, then is centrohermitian.

The inverse of this matrix is persymmetric. Persymmetric, centrohermitian and inverse of a matrix will not be described here, but it can be found in [31].

$$A = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & a_{-3} \\ a_1 & a_0 & a_{-1} & a_{-2} \\ a_2 & a_1 & a_0 & a_{-1} \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \quad \text{C.8}$$

C.3 Triangular Matrices

A lower triangular matrix is defined as a square matrix whose elements above the main diagonal are zero, and the inverse is also lower triangular matrix. The determinant of the matrix is $\det(L) = \prod_{i=1}^{N} l_{ii}$ and is given by

$$L = \begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{N1} & l_{N2} & \dots & l_{NN} \end{bmatrix} \quad \text{C.9}$$

Upper triangular matrix is called when a matrix whose elements below the main diagonal are zero. As below, the inverse of matrix must be an upper triangular matrix. The determinant is $\det(U) = \prod_{i=1}^{N} u_{ii}$

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1N} \\ 0 & u_{22} & \dots & u_{2N} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{NN} \end{bmatrix} \quad \text{C.10}$$

We can define that L^T and L^H are upper triangular matrix and U^T and U^H are lower triangular matrix. If A is a square Hermitian positive definite matrix, there are two factorizations of interest. The first factorization is called the LDL^H factorization, where D is a diagonal matrix with positive entries [14].

The second factorization is called Cholesky decomposition $A=GG^H$, where G is a lower triangular matrix

$$G = L \cdot diag\left[\sqrt{d_1}, \sqrt{d_2}, ..., \sqrt{d_N}\right] \quad \text{C.11}$$

C.4 QR Decomposition

The key of the decomposition of the KxN matrix A_m into a KxN matrix that is partitioned into a NxN upper triangular matrix R and a (K-N)xN null matrix. There exist a KxK unitary matrix Q such that

$$QA_m = \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{C.12}$$

where R has the form upper triangular matrix. With equation C.12 we suppressed the K dependence and 0 is a (K-N)xN matrix with zero elements. Note that in equation C.13, the R matrix element is the Cholesky factor of the weighted sample covariance matrix.

$$\Phi = A_m^H A_m = A_m^H Q^H Q A_m = R^H R \tag{C.13}$$

The decomposition has other properties:

a. Each row of the triangular matrix R is unique up to a complex scale factor with unit magnitude. Hence if R_1 and R_2 are two different R-factors in a QR decomposition of the A_m, we can always find a complex-valued diagonal matrix β whose complex valued diagonal elements have unit magnitude, so that $R_1=\beta R_2$.
b. For any matrix A_m there exists a unique R whose diagonal elements are all real and non-negative.
c. If K > N, the unitary matrix Q in a QR decomposition of A_m is not unique for the same R-factor.
d. Since the condition number of the unitary matrix Q is unity, it follows that the condition number of R is equal to the condition number of X_m.

In order to solve equation A.13, we substitute equation number A.12, and obtain

$$\begin{bmatrix} R \\ 0 \end{bmatrix}^H Q Q^H \begin{bmatrix} R \\ 0 \end{bmatrix} w^* = \begin{bmatrix} R \\ 0 \end{bmatrix}^H Q d_m \tag{C.14}$$

Now partition d_m as equation A.15, where p is a vector with dimension Nx1.

$$Qd_m = \begin{bmatrix} p \\ v \end{bmatrix} \tag{C.15}$$

Using the equation A.15 in A.14, we obtain two relations below; where second one can be solved due R is a upper triangular matrix

$$\begin{aligned} R^H R w^* &= R^H p \\ R w^* &= p \end{aligned} \tag{C.16}$$

From upper, we can find two important things

a. We do not need an explicit expression for Q.
b. The condition number of R is the square root of the condition number of Φ.

C.5 Givens Rotation

The givens rotation is a method to implement the QR decomposition by a sequence of plane rotations. To introduce the technique, we consider a complex vector v,

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \tag{C.17}$$

The givens matrix G is a 2x2 elements with unitary matrix

$$G = \begin{bmatrix} c* & s \\ -s* & c \end{bmatrix} \tag{C.18}$$

We select the elements of G such that

$$Gv = \begin{bmatrix} c*v_1 & sv_2 \\ -s*v_1 & cv_2 \end{bmatrix} \cong \begin{bmatrix} v_1' \\ 0 \end{bmatrix} = v' \tag{C.19}$$

The zero mean located in A.19 requires $sv_1 = cv_2$ and the unitary condition requires $|c|^2 + |s|^2 = 1$. Therefore, we can say that the final values can be described by

$$\begin{aligned} c &= \frac{v_1}{\sqrt{|v_1|^2 + |v_2|^2}} \\ a &= \frac{v_2^*}{\sqrt{|v_1|^2 + |v_2|^2}} \\ v_1' &= \sqrt{|v_1|^2 + |v_2|^2} \end{aligned} \tag{C.20}$$

Thus, *v'* has the same length as *v*, but difference between them is the only one non-zero component in *v*. Similarly, if we have a row vector $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$, we can write $uG^H = \begin{bmatrix} u_1' & 0 \end{bmatrix}$, where G is given by equation A.20, with v_1 and v_2 replaced by u_1 and u_2. This result become directly from equation A.19 by letting $u = v^H$. We must remember that the operation described by equation A.18 is called a plane rotation because, if v_1 and v_2 are real, $c = \cos\phi$ and $s = sen\phi$ where the angle is distance *v* is rotated.

Now, we consider a complex vector with dimensions Kx1 $v = [v_2, \ ..., \ v_m, \ ..., \ v_n, \ ..., \ v_k]$. Operating on v with Givens rotations G and leave all the elements of v unchanged except for v_m and v_n. Then, we can create the unitary matrix of the form

$$G(m,n) = \begin{bmatrix} I & & & & & & 0 \\ & \ddots & & & & \ddots & \\ & & c_{mm} & & s_{mn} & & \\ & & & \ddots \quad \ddots & & & \\ & & & & I & & \\ & & & \ddots \quad \ddots & & & \\ & & s_{nm} & & c_{nn} & & \\ & \ddots & & & & \ddots & \\ 0 & & & & & & I \end{bmatrix} \qquad \text{C.21}$$

G(m,n) is a identity matrix except for four elements. The two c elements are on the diagonal

$$c = c^*_{mm} = c_{nn} = \frac{v_m}{\sqrt{|v_m|^2 + |v_n|^2}} \qquad \text{C.22}$$

and the two *a* elements off the diagonal, with

$$s_{nm} = s^*_{mn} = -s^* \qquad \text{C.23}$$

$$c = c^*_{mm} = c_{nn} = \frac{v_m}{\sqrt{|v_m|^2 + |v_n|^2}} \qquad \text{C.24}$$

Then, we have made the n-th element zero and rotated into the m-th element.

$$G(m,n) = \left[v_1, \quad ..., \quad \sqrt{|v_m|^2 + |v_n|^2} \quad ..., \quad 0 \quad ..., \quad v_k \right]^T \qquad \text{C.25}$$

It have been demonstrated that applying a sequence of Givens rotations to accomplish QR decomposition of A_m, performance is much better and time of execution too.

C.6 Householder Transformation

Givens rotation technique obtains a triangular matrix R by eliminating one element with each rotation. With Householder transformation, elements will be eliminates except one element in a column in each step. The transformation can be written as

$$H = I - \frac{2uu^H}{\|u\|^2} = I - 2u\left[u^H u\right]^{-1} u^H \qquad \text{C.26}$$

where u is a vector whose norm is $\|u\|$. The transformation can be written as

$$H = I - 2P_u \qquad \text{C.27}$$

where $P_u = u\left[u^H u\right]^{-1} u^H$ and is the projection matrix into subspace u. If we pre-multiply the complex vector v by H, we have $Hv = \left(I - 2P_u\right)v = v - 2P_u v$. From equation C.26, we observe that H is a Hermitian unitary matrix, so that the transformation preserves length $\|Hv\| = \|v\|$. Now, considering the same vector that equation $v = \left[v_1, \quad v_2 \quad ... \quad v_i \quad ... \quad v_N\right]^T$. It is important to say that H matrix is used to eliminate all of the elements except v_i. So, defining

$$\begin{aligned} v_{in} &= \frac{v_i}{|v_i|} \\ \|v\|^2 &= v^H v \end{aligned} \qquad \text{C.28}$$

Using equation C.26 we can obtain

$$u = v + v_{in}\|v\|e_i \qquad \text{C.29}$$

where

$$e_i^T = \left[0 \quad ... \quad 0 \quad 1 \quad 0 \quad ... \quad 1\right]^T \qquad \text{C.30}$$

where vector has only one element non-zero. Then, if we use equation C.28 though C.30, we find

$$\|u\|^2 = u^H u = 2\|v\|\left(\|v\| + |v_i|\right) \qquad \text{C.31}$$

Then, we can observe that

$$u^H v = \left(v^H + v_{in}^* \|v\| e_i^T\right)v = \|v\|^2 + |v_i|\|v\| \qquad \text{C.32}$$

Therefore, the desired result is shown in equation C.33. The procedure to do a QR decomposition using the Household transformation is analogous to the givens rotation technique.

$$Hv = \left(I - \frac{2uu^H}{\|u\|^2} \right) v$$
$$Hv = v - u \qquad \text{C.33}$$
$$Hv = -v_{in}\|v\|e_i$$

C.7 Pseudo-Inverse (Moore-Penrose Algorithm)

The pseudo-inverse is represented by A^+, where elements matrix are real, and the definition is given by

$$A^+ = V\Sigma^+ U^T \qquad \text{C.34}$$

where $\Sigma^+ = diag\left(\frac{1}{\sigma_1} \quad ... \quad \frac{1}{\sigma_r} \quad 0 \quad ... \quad 0 \right)$. A^+ is referred to as the pseudo-inverse of A. It is the unique minimal F-form solution to the problem

$$\min_{X \in \Re^{nxm}} \|AX - I_m\|_F \qquad \text{C.35}$$

If rank(A) = n , then $A^+ = (A^T A)^{-1} A^T$, while $m = n = rank(A)$. Then, pseudo-inverse of A is the inverse of A. Typically, A^+ is defined to be the unique matrix $X \in \Re^{nxm}$, that satisfies the four Moore-Penrose conditions as follow

a. $AXA = A$
b. $XAX = X$
c. $(AX)^T = AX$
d. $(XA)^T = XA$

Past conditions amount to the requirement that AA^+ and A^+A be orthogonal projections onto range(A) and range(A^+), respectively.

APPENDIX D

INTRODUCTION TO CONVOLUTIONAL CODING

Steps involved in simulating a communication channel using convolutional encoding and Viterbi decoding are basically the generation of data information, the convolutionally encode, modulations (which include the map channel symbols), additional noise, quantizes and the perform Viterbi decoding. However, as a tutorial, this appendix is dedicated to give us a brief explanation of this code [48], also the Viterbi algorithm principle [40].

D.1 CONVOLUTIONALLY ENCODING THE DATA

Convolutionally encoding the data is accomplished using a shift register and associated combinatorial logic that performs modulo-two addition. (A shift register is merely a chain of flip-flops wherein the output of the *n*-th flip-flop is tied to the input of the (*n*+1)th flip-flop. Every time the active edge of the clock occurs, the input to the flip-flop is clocked through to the output, and thus the data are shifted over one stage.) The combinatorial logic is often in the form of cascaded exclusive-or gates. As a reminder, exclusive-or gates are two-input, one-output gates often represented by the logic symbol shown below (figure D.1), that implement the following truth-table (Table D.1):

Figure D.1: Exclusive OR Gate Symbol.

Input A	Input B	Output (A xor B)
0	0	0
0	1	1
1	0	1
1	1	0

Table D.1: Truth Table Exclusive OR gate.

The exclusive-or gate performs modulo-two addition of its inputs. When you cascade q two-input exclusive-or gates, with the output of the first one feeding one of the inputs of the second one, the output of the second one feeding one of the inputs of the third one, etc., the output of the last one in the chain is the modulo-two sum of the $q + 1$ inputs.

Another way to illustrate the modulo-two adder, and the way that is most commonly used in textbooks, is as a circle with a + symbol inside, thus:

Figure D.2: Adder modulo symbol.

Now that we have the two basic components of the convolutional encoder (flip-flops comprising the shift register and exclusive-or gates comprising the associated modulo-two adders) defined, let's look at a picture of a convolutional encoder for a rate 1/2, $K = 3$, $m = 2$ code:

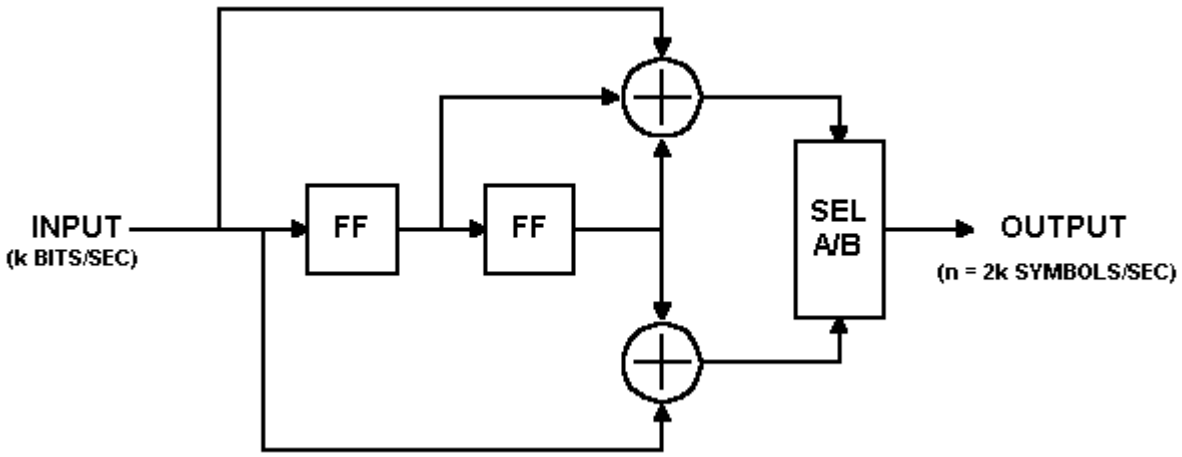


Figure D.3: Convolutional Encoder Diagram.

In this encoder, data bits are provided at a rate of k bits per second. Channel symbols are output at a rate of $n = 2k$ symbols per second. The input bit is stable during the encoder cycle. The encoder cycle starts when an input clock edge occurs. When the input clock edge occurs, the output of the left-hand flip-flop is clocked into the right-hand flip-flop, the previous input bit is clocked into the left-hand flip-flop, and a new input bit becomes available. Then the outputs of the upper and lower modulo-two adders become stable. The output selector (SEL A/B block) cycles through two states-in the first state, it selects and outputs the output of the upper modulo-two adder; in the second state, it selects and outputs the output of the lower modulo-two adder.

The encoder shown in figure D.3 encodes the $K = 3$, (7, 5) convolutional code. The octal numbers 7 and 5 represent the code generator polynomials, which when read in binary (111_2 and 101_2) correspond to the shift register connections to the upper and lower modulo-two adders, respectively. This code has been determined to be the "best" code for rate 1/2, $K = 3$.

Let's look at an example input data stream, and the corresponding output data stream. Let the input sequence be 0101110010100012. Assume that the outputs of both of the flip-flops in the shift register are initially cleared, i.e. their outputs are zeroes. The first clock cycle makes the first input bit, a zero, available to the encoder. The flip-flop outputs are both zeroes. The inputs to the modulo-two adders are all zeroes, so the output of the encoder is 00_2.

The second clock cycle makes the second input bit available to the encoder. The left-hand flip-flop clocks in the previous bit, which was a zero, and the right-hand flip-flop clocks in the zero output by the left-hand flip-flop. The inputs to the top modulo-two adder are 100_2,

so the output is a one. The inputs to the bottom modulo-two adder are 10_2, so the output is also a one. So the encoder outputs 11_2 for the channel symbols.

The third clock cycle makes the third input bit, a zero, available to the encoder. The left-hand flip-flop clocks in the previous bit, which was a one, and the right-hand flip-flop clocks in the zero from two bit-times ago. The inputs to the top modulo-two adder are 010_2, so the output is a one. The inputs to the bottom modulo-two adder are 00_2, so the output is zero. So the encoder outputs 10_2 for the channel symbols.

And so on. The timing diagram shown below illustrates the process:

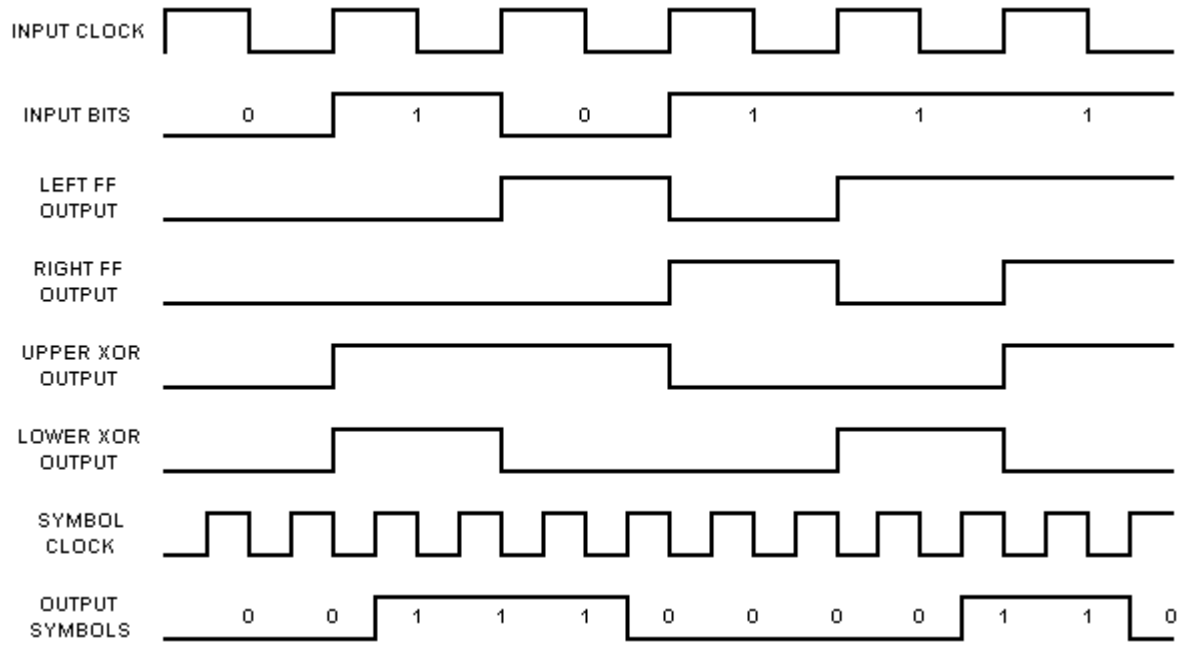


Figure D.4: Timing Diagram.

After all of the inputs have been presented to the encoder, the output sequence will be:

$$00\ 11\ 10\ 00\ 01\ 10\ 01\ 11\ 11\ 10\ 00\ 10\ 11\ 00\ 11_2.$$

Notice that the encoder paired the outputs, the first bit in each pair is the output of the upper modulo-two adder; the second bit in each pair is the output of the lower modulo-two adder.

You can see from the structure of the rate 1/2 $K = 3$ convolutional encoder and from the example given above that each input bit has an effect on three successive pairs of output symbols. That is an extremely important point and that is what gives the convolutional code its error-correcting power. The reason why will become evident when we get into the Viterbi decoder algorithm.

Now if we are only going to send the 15 data bits given above, in order for the last bit to affect three pairs of output symbols, we need to output two more pairs of symbols. This is accomplished in our example encoder by clocking the convolutional encoder flip-flops two

(equal to *m*) more times, while holding the input at zero. This is called "flushing" the encoder, and results in two more pairs of output symbols. The final binary output of the encoder is thus 00 11 10 00 01 10 01 11 11 10 00 10 11 00 11 10 11_2. If we don't perform the flushing operation, the last m bits of the message have less error-correction capability than the first through (*m* - 1)th bits had.

This is a pretty important thing to remember if you're going to use this FEC technique in a burst-mode environment. So is the step of clearing the shift register at the beginning of each burst. The encoder must start in a known state and end in a known state for the decoder to be able to reconstruct the input data sequence properly.

Now, let's look at the encoder from another perspective. You can think of the encoder as a simple state machine. The example encoder has two bits of memory, so there are four possible states. Let's give the left-hand flip-flop a binary weight of 2^1, and the right-hand flip-flop a binary weight of 2^0. Initially, the encoder is in the all-zeroes state. If the first input bit is a zero, the encoder stays in the all zeroes state at the next clock edge.

But if the input bit is a one, the encoder transitions to the 10_2 state at the next clock edge. Then, if the next input bit is zero, the encoder transitions to the 01_2 state, otherwise, it transitions to the 11_2 state. The following table gives the next state given the current state and the input, with the states given in binary:

	Next State, if	
Current State	**Input = 0:**	**Input = 1:**
00	00	10
01	00	10
10	01	11
11	01	11

Table D.2: Transition table of the encoder.

The above table is often called a state transition table. We'll refer to it as the `next state` table. Now let us look at a table that lists the channel output symbols, given the current state and the input data, which we'll refer to as the `output` table (table D.3).

You should now see that with these two tables, you can completely describe the behavior of the example rate 1/2, K = 3 convolutional encoder. Note that both of these tables have $2^{(K-1)}$ rows, and 2^k columns, where K is the constraint length and k is the number of bits input to the encoder for each cycle. These two tables will come in handy when we start discussing the Viterbi decoder algorithm.

	Output Symbols, if	
Current State	**Input = 0:**	**Input = 1:**
00	00	11
01	11	00
10	10	01
11	01	10

Table D.3: Channel output symbols.

D.2 PERFORMING VITERBI DECODING

The Viterbi decoder itself is the primary focus of this sectionl. Perhaps the single most important concept to aid in understanding the Viterbi algorithm is the trellis diagram. The figure D.5 below shows the trellis diagram for our example rate 1/2 $K = 3$ convolutional encoder, for a 15-bit message:

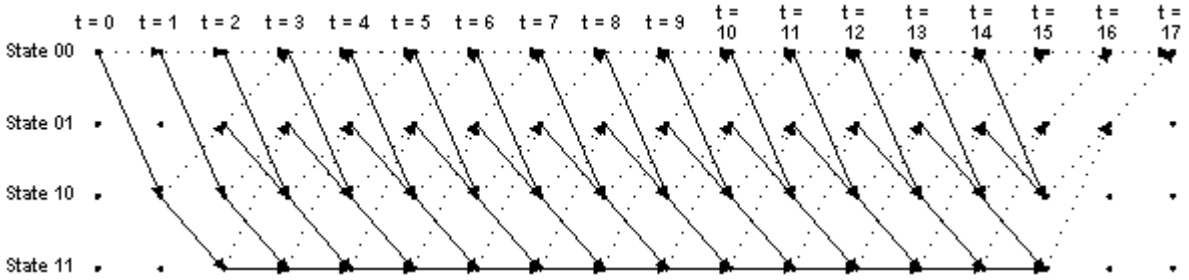


Figure D.5: Trellis diagram for a convolutional encoder.

The four possible states of the encoder are depicted as four rows of horizontal dots. There is one column of four dots for the initial state of the encoder and one for each time instant during the message.

For a 15-bit message with two encoder memory flushing bits, there are 17 time instants in addition to t = 0, which represents the initial condition of the encoder. The solid lines connecting dots in the diagram represent state transitions when the input bit is a one. The dotted lines represent state transitions when the input bit is a zero.

Notice the correspondence between the arrows in the trellis diagram and the state transition table (table D.2) discussed above. Also notice that since the initial condition of the encoder is State 00_2, and the two memory flushing bits are zeroes, the arrows start out at State 00_2 and end up at the same state.

The following diagram shows the states of the trellis that are actually reached during the encoding of our example 15-bit message:

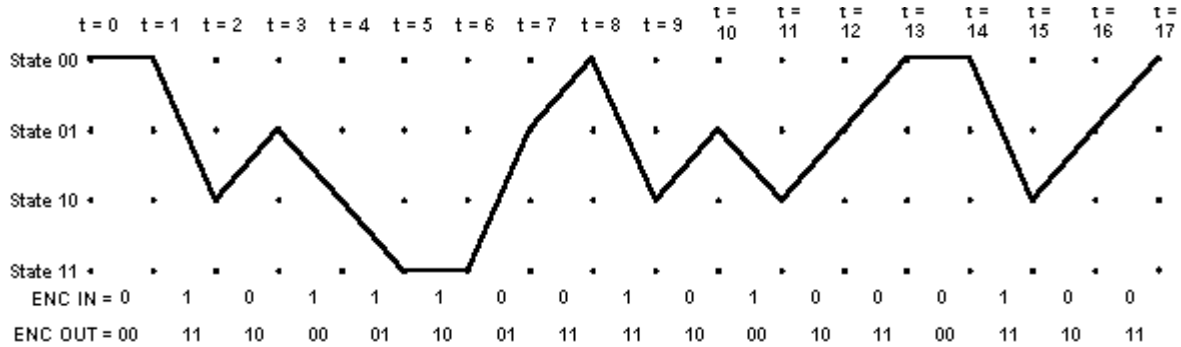


Figure D.6: Status of the trellis that are searching.

The encoder input bits and output symbols are shown at the bottom of the diagram. Notice the correspondence between the encoder output symbols and the output table (Table D.3)

discussed above. Let's look at that in more detail, using the expanded version of the transition between one time instant to the next shown below:

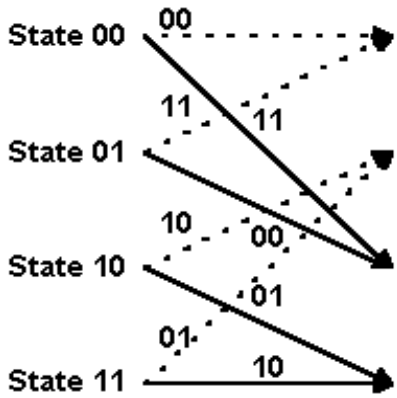


Figure D.7: Changes between one state to another.

The two-bit numbers labeling the lines are the corresponding convolutional encoder channel symbol outputs. Remember that dotted lines represent cases where the encoder input is a zero, and solid lines represent cases where the encoder input is a one. (In the figure D.7, the two-bit binary numbers labeling dotted lines are on the left, and the two-bit binary numbers labeling solid lines are on the right.)

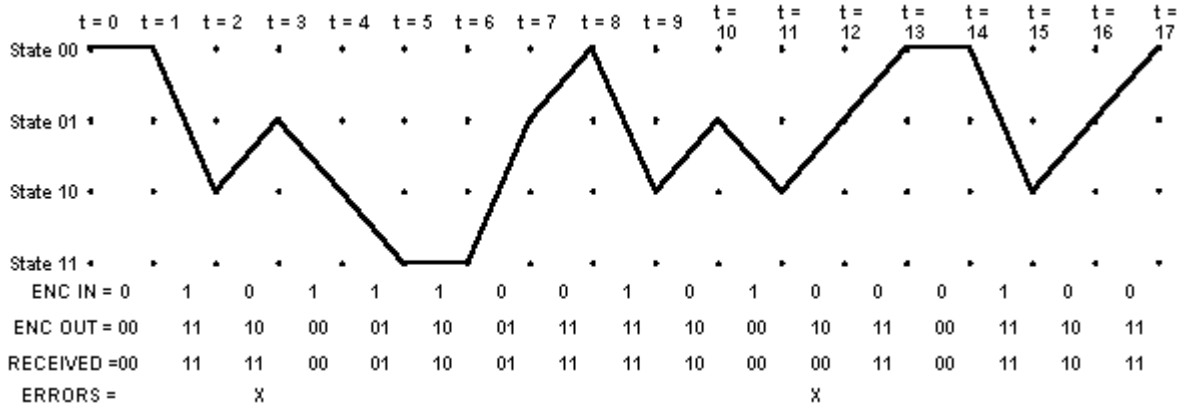


Figure D.8: Trellis message with a couple of errors.

For our example, we're going to use hard-decision symbol inputs to keep things simple. (The example source code uses soft-decision inputs to achieve better performance.) Suppose we receive the above encoded message with a couple of bit errors (Figure D.8).

Each time we receive a pair of channel symbols, we're going to compute a metric to measure the "distance" between what we received and all of the possible channel symbol pairs we could have received. Going from t = 0 to t = 1, there are only two possible channel symbol pairs we could have received: 00_2, and 11_2. That's because we know the convolutional encoder was initialized to the all-zeroes state, and given one input bit = one or zero, there are only two states we could transition to and two possible outputs of the encoder. These possible outputs of the encoder are 00_2 and 11_2. The metric we're going to

use for now is the Hamming distance between the received channel symbol pair and the possible channel symbol pairs.

The Hamming distance is computed by simply counting how many bits are different between the received channel symbol pair and the possible channel symbol pairs. The results can only be zero, one, or two. The Hamming distance (or other metric) values we compute at each time instant for the paths between the states at the previous time instant and the states at the current time instant are called branch metrics. For the first time instant, we're going to save these results as "accumulated error metric" values, associated with states. For the second time instant on, the accumulated error metrics will be computed by adding the previous accumulated error metrics to the current branch metrics.

At t = 1, we received 00_2. The only possible channel symbol pairs we could have received are 00_2 and 11_2. The Hamming distance between 00_2 and 00_2 is zero. The Hamming distance between 00_2 and 11_2 is two. Therefore, the branch metric value for the branch from State 00_2 to State 00_2 is zero, and for the branch from State 00_2 to State 10_2 it's two. Since the previous accumulated error metric values are equal to zero, the accumulated metric values for State 00_2 and for State 10_2 are equal to the branch metric values. The accumulated error metric values for the other two states are undefined. The figure D.9 illustrates the results at t = 1:

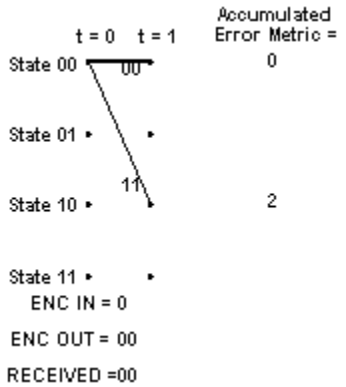


Figure D9: Trellis diagram in time T = 1.

Note that the solid lines between states at t = 1 and the state at t = 0 illustrate the predecessor-successor relationship between the states at t = 1 and the state at t = 0 respectively. This information is shown graphically in the figure, but is stored numerically in the actual implementation. To be more specific, or maybe clear is a better word, at each time instant t, we will store the number of the predecessor state that led to each of the current states at t.

Now let's look what happens at t = 2. We received a 11_2 channel symbol pair. The possible channel symbol pairs we could have received in going from t = 1 to t = 2 are 00_2 going from State 00_2 to State 00_2, 11_2 going from State 00_2 to State 10_2, 10_2 going from State 10_2 to State 01_2, and 01_2 going from State 10_2 to State 11_2. The Hamming distance between 00_2 and 11_2 is two, between 11_2 and 11_2 is zero, and between 10_2 or 01_2 and 11_2 is one. We add these branch metric values to the previous accumulated error metric values associated

with each state that we came from to get to the current states. At t = 1, we could only be at State 00_2 or State 10_2. The accumulated error metric values associated with those states were 0 and 2 respectively. The figure below shows the calculation of the accumulated error metric associated with each state, at t = 2.

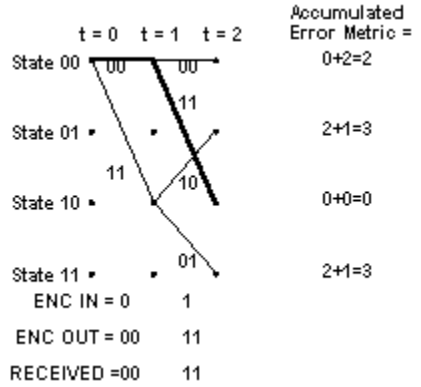


Figure D.10: Trellis diagram in time T = 2.

That's all the computation for t = 2. What we carry forward to t = 3 will be the accumulated error metrics for each state, and the predecessor states for each of the four states at t = 2, corresponding to the state relationships shown by the solid lines in the illustration of the trellis. Now look at the figure for t = 3. Things get a bit more complicated here, since there are now two different ways that we could get from each of the four states that were valid at t = 2 to the four states that are valid at t = 3.

The accumulated error metrics associated with each branch, and discard the larger one of each pair of branches leading into a given state. If the members of a pair of accumulated error metrics going into a particular state are equal, we just save that value.

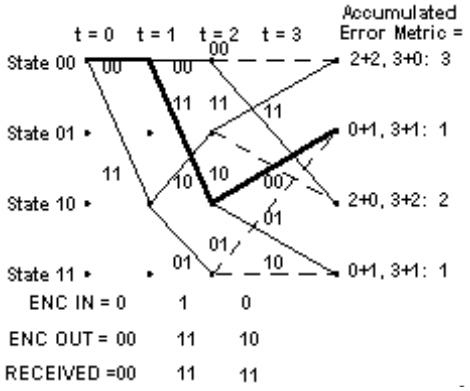


Figure D.11: Trellis diagram in time T = 3.

The other thing that's affected is the predecessor-successor history we're keeping. For each state, the predecessor that survives is the one with the lower branch metric. If the two accumulated error metrics are equal, some people use a fair coin toss to choose the surviving predecessor state. Others simply pick one of them consistently, i.e. the upper branch or the lower branch. It probably doesn't matter which method you use. The operation of adding the previous accumulated error metrics to the new branch metrics,

comparing the results, and selecting the smaller (smallest) accumulated error metric to be retained for the next time instant is called the add-compare-select operation. The figure D.11 shows the results of processing t = 3. Note that the third channel symbol pair we received had a one-symbol error. The smallest accumulated error metric is a one, and there are two of these. Let's see what happens now at t = 4. The processing is the same as it was for t = 3. The results are shown in the figure:

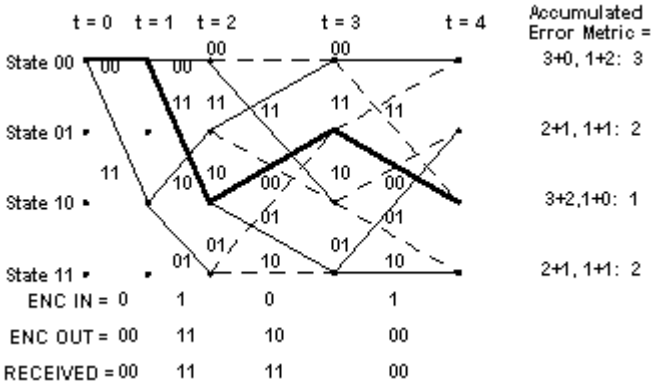


Figure D.12: Trellis diagram in time T = 4.

Notice that at t = 4, the path through the trellis of the actual transmitted message, shown in bold, is again associated with the smallest accumulated error metric. Let's look at t = 5:

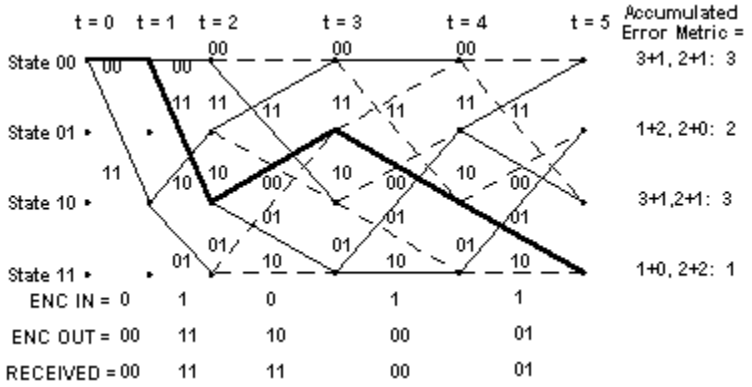


Figure D.13: Trellis diagram in time T = 5.

At t = 5, the path through the trellis corresponding to the actual message, shown in bold, is still associated with the smallest accumulated error metric. This is the thing that the Viterbi decoder exploits to recover the original message.

At t = 17, the trellis looks like this, with the clutter of the intermediate state history removed (figure D.14).

The decoding process begins with building the accumulated error metric for some number of received channel symbol pairs, and the history of what states preceded the states at each time instant t with the smallest accumulated error metric. Once this information is built up, the Viterbi decoder is ready to recreate the sequence of bits that were input to the convolutional encoder when the message was encoded for transmission.

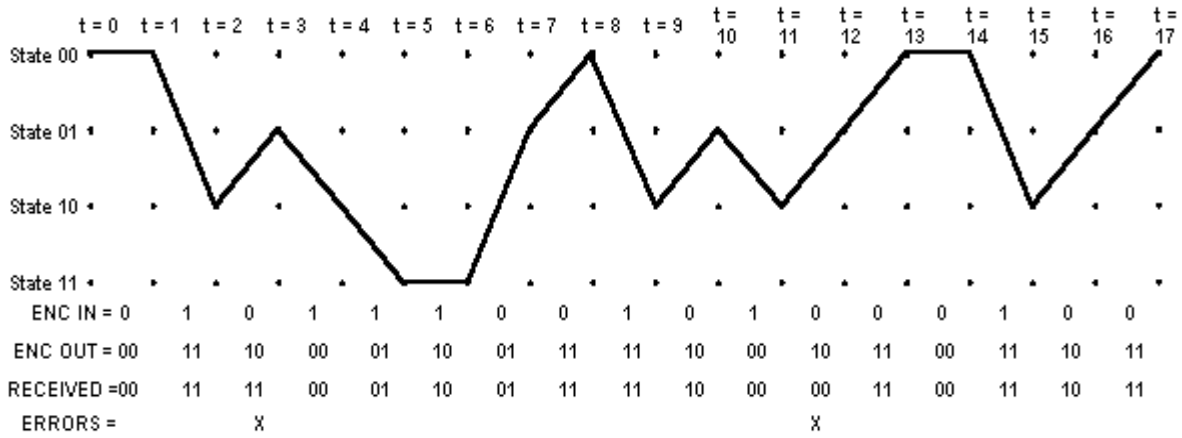


Figure D.14: Trellis diagram in time T = 17.

This is accomplished by the following steps: First, select the state having the smallest accumulated error metric and save the state number of that state. Iteratively perform the following step until the beginning of the trellis is reached: Working backward through the state history table, for the selected state, select a new state which is listed in the state history table as being the predecessor to that state. Save the state number of each selected state. This step is called traceback. Now work forward through the list of selected states saved in the previous steps. Look up what input bit corresponds to a transition from each predecessor state to its successor state. That is the bit that must have been encoded by the convolutional encoder.

The following table (table D.4) shows the accumulated metric for the full 15-bit (plus two flushing bits) example message at each time t:

t =	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
State 00_2		0	2	3	3	3	3	4	1	3	4	3	3	2	2	4	5	2
State 01_2			3	1	2	2	3	1	4	4	1	4	2	3	4	4	2	
State 10_2		2	0	2	1	3	3	4	3	1	4	1	4	3	3	2		
State 11_2			3	1	2	1	1	3	4	4	3	4	2	3	4	4		

Table D.4: Accumulated metric for the full 15 bits.

It is interesting to note that for this hard-decision-input Viterbi decoder example, the smallest accumulated error metric in the final state indicates how many channel symbol errors occurred.

The following state history table shows the surviving predecessor states for each state at each time t:

t =	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
State 00_2	0	0	0	1	0	1	1	0	1	0	0	1	0	1	0	0	0	1
State 01_2	0	0	2	2	3	3	2	3	3	2	2	3	2	3	2	2	2	0
State 10_2	0	0	0	0	1	1	1	0	1	0	0	1	1	0	1	0	0	0
State 11_2	0	0	2	2	3	2	3	2	3	2	2	3	2	3	2	2	0	0

Table D.5: Surviving predecessor states for each time.

The following table shows the states selected when tracing the path back through the survivor state table shown above:

t =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	0	0	2	1	2	3	3	1	0	2	1	2	1	0	0	2	1	0

Table D.6: Accumulated metric for the full 15 bits.

Using a table that maps state transitions to the inputs that caused them, we can now recreate the original message. Here is what this table looks like for our example rate 1/2 K = 3 convolutional codes:

	Input was, Given Next State =			
Current State	**$00_2 = 0$**	**$01_2 = 1$**	**$10_2 = 2$**	**$11_2 = 3$**
$00_2 = 0$	0	x	1	x
$01_2 = 1$	0	x	1	x
$10_2 = 2$	x	0	x	1
$11_2 = 3$	x	0	x	1

Table D.7: Map state transitions to the inputs.

Note: In the above table, x denotes an impossible transition from one state to another state. So now we have all the tools required to recreate the original message from the message we received:

t =	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	1	0	1	1	1	0	0	1	0	1	0	0	0	1

Table D.8: Original message recreation.

The two flushing bits are discarded. Here's an insight into how the trace back algorithm eventually finds its way onto the right path even if it started out choosing the wrong initial state. This could happen if more than one state had the smallest accumulated error metric, for example. I'll use the figure for the trellis at t = 3 again to illustrate this point:

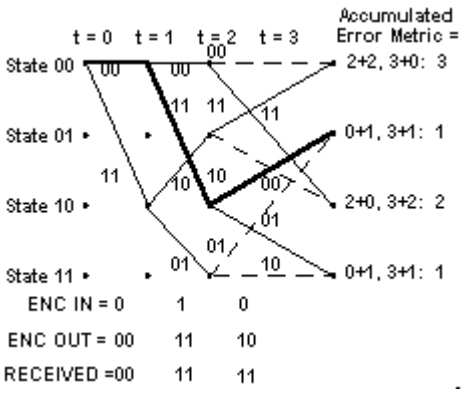


Figure D.15: Error accumulation in trellis diagram

See how at t = 3, both States 01_2 and 11_2 had an accumulated error metric of 1. The correct path goes to State 01_2 -notice that the bold line showing the actual message path goes into this state. But suppose we choose State 11_2 to start our trace back. The predecessor state for

State 11_2 , which is State 10_2 , is the same as the predecessor state for State 01_2! This is because at t = 2, State 10_2 had the smallest accumulated error metric. So after a false start, we are almost immediately back on the correct path.

For the example 15-bit message, we built the trellis up for the entire message before starting traceback. For longer messages, or continuous data, this is neither practical nor desirable, due to memory constraints and decoder delay. Research has shown that a traceback depth of *K* x 5 is sufficient for Viterbi decoding with the type of codes we have been discussing. Any deeper traceback increases decoding delay and decoder memory requirements, while not significantly improving the performance of the decoder. The exception is punctured codes, which I'll describe later. They require deeper traceback to reach their final performance limits.

To implement a Viterbi decoder in software, the first step is to build some data structures around which the decoder algorithm will be implemented. These data structures are best implemented as arrays. The primary six arrays that we need for the Viterbi decoder are as follows:

1. A copy of the convolutional encoder `next state` table, the state transition table of the encoder. The dimensions of this table (rows x columns) are $2^{(K - 1)}$ x 2^k. This array needs to be initialized before starting the decoding process.
2. A copy of the convolutional encoder `output` table. The dimensions of this table are $2^{(K - 1)}$ x 2^k. This array needs to be initialized before starting the decoding process.
3. An array (table) showing for each convolutional encoder current state and next state, what input value (0 or 1) would produce the next state, given the current state. We'll call this array the `input` table. Its dimensions are $2^{(K - 1)}$ x $2^{(K - 1)}$. This array needs to be initialized before starting the decoding process.
4. An array to store state predecessor history for each encoder state for up to K x 5 + 1 received channel symbol pairs. We'll call this table the `state history` table. The dimensions of this array are $2^{(K - 1)}$ x (K x 5 + 1). This array does not need to be initialized before starting the decoding process.
5. An array to store the accumulated error metrics for each state computed using the add-compare-select operation. This array will be called the `accumulated error metric` array. The dimensions of this array are $2^{(K - 1)}$ x 2. This array does not need to be initialized before starting the decoding process.
6. An array to store a list of states determined during traceback (term to be explained below). It is called the `state sequence` array. The dimensions of this array are (K x 5) + 1. This array does not need to be initialized before starting the decoding process.

Before getting into the example source code, for purposes of completeness, I want to talk briefly about other rates of convolutional codes that can be decoded with Viterbi decoders. Earlier, I mentioned punctured codes, which are a common way of achieving higher code rates, i.e. larger ratios of k to n. Punctured codes are created by first encoding data using a rate 1/n encoder such as the example encoder described in this tutorial, and then deleting some of the channel symbols at the output of the encoder. The process of deleting some of

the channel output symbols is called puncturing. For example, to create a rate 3/4 code from the rate 1/2 code described in this tutorial, one would simply delete channel symbols in accordance with the following puncturing pattern:

1	0	1
1	1	0

Table D.9: Puncturing patter table with three columns.

where a one indicates that a channel symbol is to be transmitted, and a zero indicates that a channel symbol is to be deleted. To see how this make the rate be 3/4, think of each column of the above table as corresponding to a bit input to the encoder, and each one in the table as corresponding to an output channel symbol. There are three columns in the table, and four ones. You can even create a rate 2/3 code using a rate 1/2 encoder with the following puncturing pattern which has two columns and three ones.

1	1
1	0

Table D.10: Puncturing patter table with two columns.

To decode a punctured code, one must substitute null symbols for the deleted symbols at the input to the Viterbi decoder. Null symbols can be symbols quantized to levels corresponding to weak ones or weak zeroes, or better, can be special flag symbols that when processed by the ACS circuits in the decoder, result in no change to the accumulated error metric from the previous state.

Of course, n does not have to be equal to two. For example, a rate 1/3, $K = 3$, (7, 7, 5) code can be encoded using the encoder shown below:

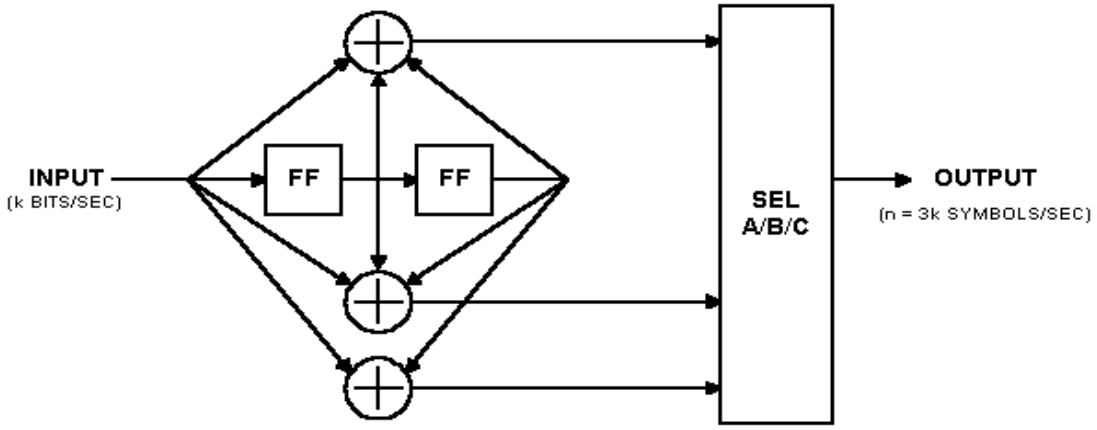


Figure D.16: Encoder with rate 1/3 and $K = 3$.

This encoder has three modulo-two adders, so for each input bit, it can produce three channel symbol outputs. Of course, with suitable puncturing patterns, you can create higher-rate codes using this encoder as well.