

**ANÁLISIS COMPARATIVO DE ALGORITMOS “LATTICE REDUCTION-AIDED”
PARA LA DETECCIÓN DE SEÑALES EN SISTEMAS MIMO**



**DIEGO FERNANDO MOSQUERA BETANCOURT
LEONARDO NIETO CORTÉS**

Universidad del Cauca
Facultad de Ingeniería Electrónica y de Telecomunicaciones
Departamento de Telecomunicaciones
Grupo Nuevas Tecnologías en Telecomunicaciones GNTT
Gestión Integrada de Redes, Servicios y Arquitecturas de Telecomunicaciones
Popayán
2012

**ANÁLISIS COMPARATIVO DE ALGORITMOS “LATTICE REDUCTION-AIDED”
PARA LA DETECCIÓN DE SEÑALES EN SISTEMAS MIMO**



**Trabajo de grado presentado para optar por el título de Ingeniero en Electrónica y
Telecomunicaciones**

**DIEGO FERNANDO MOSQUERA BETANCOURT
LEONARDO NIETO CORTÉS**

**Directora:
Mag. Claudia Milena Hernández Bonilla**

Universidad del Cauca
Facultad de Ingeniería Electrónica y de Telecomunicaciones
Departamento de Telecomunicaciones
Grupo Nuevas Tecnologías en Telecomunicaciones GNTT
Gestión Integrada de Redes, Servicios y Arquitecturas de Telecomunicaciones
Popayán
2012

CONTENIDO

pág.

1	INTRODUCCIÓN	1
1.1	INTRODUCCIÓN A LOS SISTEMAS MIMO Y A LOS ALGORITMOS “ <i>LATTICE REDUCTION-AIDED</i> ”	1
1.2	MOTIVACIÓN DEL TRABAJO DE GRADO	2
1.3	OBJETIVOS	2
1.3.1	Objetivo General	2
1.3.2	Objetivos Específicos	2
1.4	ESTRUCTURA DEL TRABAJO DE GRADO	3
2	GENERALIDADES EN SISTEMAS MIMO Y ALGORITMOS DE DETECCIÓN	4
2.1	SISTEMAS MIMO 2X2 Y 4X4	4
2.1.1	Diversidad Espacial	5
2.1.2	Multiplexación Espacial	5
2.1.3	Diagrama en Bloques de un Sistema MIMO	6
2.1.4	Modulación	6
2.1.5	Detección en Sistemas MIMO	8
2.1.6	Algoritmo de Máxima Verosimilitud - ML	9
2.2	ALGORITMOS <i>LATTICE REDUCTION-AIDED</i> - LRA	11
2.2.1	<i>Lattices</i>	13
2.2.2	Propiedades de las <i>Lattices</i>	15
2.2.3	Técnicas de Reducción <i>Lattice</i>	16
2.3	COMPARACIÓN ALGORITMOS LRA	19
2.4	ALGORITMO <i>LENSTRA LENSTRA LOVÁSZ</i>	20
2.4.1	Procedimiento Algoritmo <i>Lenstra Lenstra Lovász</i>	20
2.5	ALGORITMO <i>SEYSEN</i>	22
2.5.1	Criterio de Reducción de <i>Seysen</i>	23

2.5.2	Procedimiento Algoritmo de <i>Seysen</i>	24
3	METODOLOGÍA.....	26
3.1	METODOLOGÍA PARA EL DESARROLLO DE SIMULACIONES.....	26
3.1.1	Fase 1: Definición del Sistema.....	27
3.1.2	Fase 2: Análisis	27
3.1.3	Fase 3: Formulación del Modelo	28
3.1.4	Fase 4: Selección del Lenguaje	33
3.1.5	Fase 5: Codificación del Modelo	33
3.1.6	Fase 6: Validación.....	43
3.1.7	Fase 7: Experimentación	43
3.1.8	Fase 8: Implantación	43
4	VALIDACIÓN DE RESULTADOS.....	44
4.1	COMPARACIÓN CON OTROS TRABAJOS DE GRADO	44
4.1.1	Algoritmo <i>Lenstra Lenstra Lovász</i>	44
4.1.2	Algoritmo <i>Seysen</i>	45
4.2	VERIFICACIÓN DE LOS PRINCIPIOS DE LA TEORÍA DE <i>LATTICES</i>	46
4.2.1	Índice de Ortogonalidad.....	46
5	ANÁLISIS DE RESULTADOS.....	48
5.1	ANÁLISIS DE DESEMPEÑO	48
5.1.1	Escenario 1	48
5.1.2	Escenario 2	49
5.1.3	Escenario 3	52
5.1.4	Escenario 4	53
5.2	ANÁLISIS DE TIEMPOS DE REDUCCIÓN	57
5.2.1	Sistema MIMO 2X2.....	57
5.2.2	Sistema MIMO 4X4.....	57
6	CONCLUSIONES Y TRABAJOS FUTUROS	59
6.1	CONCLUSIONES	59

6.2 TRABAJOS FUTUROS.....	60
REFERENCIAS.....	61
ANEXO A.....	65
DIFERENCIA DE VALORES DE DESEMPEÑO	65
ANEXO B.....	68
DIFERENCIA DE VALORES DE DESEMPEÑO ENTRE ESCENARIOS.....	68
ANEXO C.....	73
CURVAS DE DESEMPEÑO POR ALGORITMOS Y MODULACIÓN.....	73

LISTA DE FIGURAS

	Pág.
Figura 2.1. Sistema MIMO	4
Figura 2.2. Diagrama en bloques de un sistema MIMO	6
Figura 2.3. Constelaciones M-QAM.....	7
Figura 2.4. Esquema de modulación 16QAM.....	8
Figura 2.5. Árbol de decisión ML.....	10
Figura 2.6. Sistema MIMO equivalente para algoritmos <i>Lattice Reduction-Aided</i>	11
Figura 2.7. Reducción Lattice.....	12
Figura 2.8. Transformación de la solución al dominio original	13
Figura 2.9. Bases generadoras y lattices	14
Figura 2.10. Región fundamental (azul), región voronoi (rojo) [16]	15
Figura 3.1. Fases de la metodología	26
Figura 3.2. Diagrama de bloques sistema MIMO	28
Figura 3.3. Diagrama de flujo del sistema MIMO	30
Figura 3.4. Diagrama de flujo algoritmo LLL	31
Figura 3.5. Diagrama de flujo algoritmo de Seysen.....	32
Figura 3.6. Código en Matlab® del bloque generador de datos.....	34
Figura 3.7. Código en Matlab® del bloque modulador.....	35
Figura 3.8. Código en Matlab® del bloque demultiplexor.....	35
Figura 3.9. Código en Matlab® del bloque canal	36
Figura 3.10. Código en Matlab® del detector ML	37
Figura 3.11. Código en Matlab® del detector ZF.....	38
Figura 3.12. Código en Matlab® del algoritmo LLL	39
Figura 3.13. Código en Matlab® del algoritmo de Seysen.....	40
Figura 3.14. Código en Matlab® del bloque demodulador	41
Figura 3.15. Código en Matlab® del bloque medidor de BER – Errores.....	42

Figura 3.16. Código en Matlab® del bloque medidor de BER - BER	42
Figura 5.1. Curvas de desempeño para el escenario 1	48
Figura 5.2. Curvas de desempeño para el escenario 2	50
Figura 5.3. Curvas de desempeño de los algoritmos simulados para sistema MIMO 2x2	51
Figura 5.4. Curvas de desempeño para el escenario 3	52
Figura 5.5. Curvas de desempeño para el escenario 4	54
Figura 5.6. Curvas de desempeño de los algoritmos simulados para sistema MIMO 4x4	56

LISTA DE TABLAS

	Pág.
Tabla 2-1. Tabla comparativa de algoritmos LRA	19
Tabla 3-1. Escenarios de simulación	43
Tabla 4-1. Diferencia de SNR para algoritmo de referencia y LLL simulado	45
Tabla 4-2. Diferencia de SNR para algoritmo de referencia y Seysen simulado	45
Tabla 4-3. Media y varianza del índice de ortogonalidad sin reducción lattice	46
Tabla 4-4. Media y varianza del índice de ortogonalidad utilizando LLL	46
Tabla 4-5. Media y varianza del índice de ortogonalidad utilizando Seysen	47
Tabla 5-1. Tabla de desempeño para el escenario 1	49
Tabla 5-2. Tabla de desempeño para el escenario 2	50
Tabla 5-3. Tabla de desempeño para el escenario 3	53
Tabla 5-4. Tabla de desempeño para el escenario 4	55
Tabla 5-5. Tiempos de reducción lattice e iteraciones en un sistema MIMO 2x2.....	57
Tabla 5-6. Tiempos de reducción lattice e iteraciones en un sistema MIMO 4x4.....	58

LISTA DE ACRÓNIMOS

3G	<i>Third Generation</i> - Tercera Generación
4G	<i>Fourth Generation</i> - Cuarta Generación
AWGN	<i>Additive White Gaussian Noise</i> - Ruido Blanco Aditivo Gaussiano
BER	<i>Bit Error Rate</i> - Tasa de Error de Bit
CV	Coeficiente de Variación
D-BLAST	<i>Diagonal-Bell Labs Layered Space Time</i> - Codificación Diagonal Espacio Temporal por Capas de Laboratorios Bell
H-BLAST	<i>Horizontal-Bell Labs Layered Space Time</i> - Codificación Horizontal Espacio Temporal por Capas de Laboratorios Bell
LLL	<i>Lenstra Lenstra Lovász</i>
LR	<i>Lattice Reduction</i> - Reducción <i>Lattice</i>
LRA	<i>Lattice Reduction Aided</i> – Asistido por Reducción <i>Lattice</i>
LRA-MMSE	<i>Lattice Reduction Aided - Minimum Mean Squared Error</i> - Mínimo Error Cuadrático Medio apoyado por Reducción <i>lattice</i>
LTE	<i>Long Term Evolution</i> - Evolución a Largo Termino
MIMO	<i>Multiple Input Multiple Output</i> - Múltiple Entrada Múltiple Salida
ML	<i>Maximum Likelihood</i> - Máxima Verosimilitud
MMSE	<i>Minimum Mean Squared Error</i> - Mínimo Error Cuadrático Medio
MQAM	<i>M-Order Quadrature Amplitude Modulation</i> - Modulación por Amplitud en Cuadratura de Orden M
SNR	<i>Signal to Noise Ratio</i> - Relación Señal a Ruido
V-BLAST	<i>Vertical-Bell Labs Layered Space Time</i> - Codificación Vertical Espacio Temporal por Capas de Laboratorios Bell
ZF	<i>Zero Forcing</i> –Forzado a Cero

NOTACIÓN

La notación utilizada para mencionar las diferentes variables en este documento es la siguiente:

- Variables que representan un valor escalar:

Letra minúscula sin negrita.

a

- Variables que representan un vector:

Letra minúscula en negrita.

a

- Variables que representan matrices:

Letra mayúscula en negrita.

A

1 INTRODUCCIÓN

1.1 INTRODUCCIÓN A LOS SISTEMAS MIMO Y A LOS ALGORITMOS “LATTICE REDUCTION-AIDED”

En vista del constante crecimiento de las tecnologías de transmisión inalámbrica y de la necesidad de prestar servicios de mayor calidad, se desarrollaron los sistemas de Múltiples Entradas y Múltiples Salidas (MIMO, *Multiple Input Multiple Output*) los cuales aprovechan mejor el recurso radioeléctrico, incrementan la velocidad de transmisión de datos y combaten efectos indeseados en el canal de comunicación.

Los sistemas MIMO usan múltiples antenas en transmisión y recepción y su objetivo principal es sacar el máximo provecho del efecto de multitrayecto mediante la multiplexación espacial o diversidad espacial. Las múltiples antenas utilizadas en recepción entregan al sistema una superposición lineal de símbolos de información transmitidos por separado [1], por lo cual se necesita un sistema de detección que determine de forma precisa los símbolos enviados. El sistema de detección se basa en un algoritmo matemático que permite estimar las señales transmitidas teniendo en cuenta la interferencia por multitrayectoria, la respuesta del canal y el ruido sumado en recepción[2].

Para implementar la detección en un sistema MIMO se pueden usar diferentes algoritmos, uno de ellos es el algoritmo de Máxima Verosimilitud (ML, *Maximum Likelihood*), el cual calcula la distancia euclidiana entre el símbolo recibido y cada uno de los posibles símbolos transmitidos, escogiendo aquel que se encuentra a la distancia mínima. El algoritmo ML es el método de mayor precisión y de mayor complejidad computacional, por lo que su implementación no se realiza en escenarios reales. Para obtener resultados similares con menor complejidad computacional pueden usarse técnicas de aproximación al algoritmo ML como los algoritmos asistidos por reducción *lattice* (LRA, *Lattice Reduction Aided*).

Los algoritmos LRA son algoritmos de detección de señales que trabajan en conjunto con técnicas de reducción *lattice* para lograr un mejor desempeño del sistema MIMO. Las técnicas de reducción *lattice* facilitan el proceso de detección utilizando algoritmos como el de *Seysen* y el de *Lenstra, Lenstra y Lovász* (LLL, *Lenstra Lenstra Lovász*) que por medio de diferentes procesos de reducción matricial mejoran la condición de la matriz del canal permitiendo obtener menores valores de Tasa de Error de Bit (BER, *Bit Error Rate*) para iguales valores de Relación Señal a Ruido (SNR, *Signal to Noise Ratio*) [3].



1.2 MOTIVACIÓN DEL TRABAJO DE GRADO

El desarrollo de las telecomunicaciones en la actualidad está enfocado principalmente hacia entornos inalámbricos, donde los dispositivos móviles cuentan con altas capacidades de procesamiento, y donde se requieren conexiones confiables y de alta velocidad para ofrecer servicios que vayan a la par con las características de estos equipos. El uso de sistemas MIMO permite el desarrollo de tecnologías que mejoran las velocidades de transmisión y la confiabilidad de la comunicación como sistemas de Tercera Generación (3G, *Third Generation*), Cuarta Generación (4G, *Fourth Generation*) o Evolución a Largo Término (LTE, *Long Term Evolution*) [4], gracias a las cuales es posible la implementación de servicios móviles rápidos y confiables.

Aunque los sistemas MIMO han podido satisfacer los requerimientos de velocidad y confiabilidad de las comunicaciones inalámbricas, el desarrollo de la tecnología implica una mejora constante, por lo que este trabajo de grado, toma como objeto de estudio los algoritmos LRA para la detección de señales en sistemas MIMO, con el fin de complementar trabajos anteriores que determinaron que su implementación ofrece mejoras en el desempeño de un sistema a cambio de un aumento en la complejidad computacional [5-8], pero que no realizaron un análisis completo en sistemas MIMO 2x2 y 4x4 con Modulación por Amplitud en Cuadratura de Orden 4 (4QAM, *4-Order Quadrature Amplitude Modulation*) y Modulación por Amplitud en Cuadratura de Orden 16 (16QAM, *16-Order Quadrature Amplitude Modulation*).

1.3 OBJETIVOS

1.3.1 Objetivo General

Analizar comparativamente el desempeño de los algoritmos de detección de señal en sistemas MIMO punto a punto 2x2 y 4x4, basados en algoritmos *Lattice Reduction-Aided*.

1.3.2 Objetivos Específicos

1. Analizar y seleccionar los algoritmos *Lattice Reduction-Aided* para implementar la etapa de detección de señal en un sistema MIMO¹ punto a punto 2x2 y 4x4.

¹ Sistema con canal invariante en el tiempo, conocimiento perfecto del canal y esquemas de modulación 4QAM y 16QAM.



2. Modelar y simular dos algoritmos de detección de señal en un sistema MIMO punto a punto 2x2 y 4x4.
3. Comparar el desempeño de los algoritmos modelados con el algoritmo de máxima verosimilitud.

1.4 ESTRUCTURA DEL TRABAJO DE GRADO

El trabajo está organizado de la siguiente forma:

En el capítulo dos se presentan los conceptos básicos relacionados con los sistemas MIMO y se explica el proceso de detección mediante el algoritmo ML. Se describen la teoría de *lattices* y las diferentes técnicas de reducción, haciendo énfasis en los algoritmos de *Seysen* y LLL.

El capítulo tres desarrolla la metodología empleada para cumplir con los objetivos establecidos, se explica cada una de las fases y se presentan diagramas en bloques, diagramas de flujo y partes del código computacional desarrollado para su implementación.

En el capítulo cuatro se presenta la validación de los resultados obtenidos comparándolos con resultados de otras investigaciones sobre los algoritmos en estudio.

En el capítulo cinco se presenta la evaluación del desempeño de los algoritmos y el análisis de los resultados obtenidos.

El capítulo seis contiene las conclusiones del trabajo de grado y las recomendaciones para el desarrollo de futuros trabajos en esta línea de investigación.

2 GENERALIDADES EN SISTEMAS MIMO Y ALGORITMOS DE DETECCIÓN

2.1 SISTEMAS MIMO 2X2 Y 4X4

Los sistemas MIMO surgen como una forma de aprovechar el multitrayecto, en el cual la señal transmitida llega al receptor por diferentes caminos generando interferencia y dispersión en tiempo, entre otros fenómenos que están presentes en las comunicaciones inalámbricas[9]. La mejora en el desempeño de estos sistemas se debe al aprovechamiento del multitrayecto por medio de la implementación de la diversidad espacial con la utilización de códigos espacio temporales[10].

La estructura básica de un sistema MIMO consiste en un arreglo de m antenas en el transmisor, desde las cuales se envía la información por múltiples trayectorias y otro arreglo de n antenas en el receptor que puede ser igual o diferente a las del transmisor, donde h_{ij} son las funciones de transferencia compleja, como se puede observar en la figura 2.1. Además de las antenas se necesitan etapas de procesamiento en transmisión y recepción. En el receptor es necesario implementar mecanismos de detección con el fin de interpretar correctamente los distintos flujos de información recibidos[11].

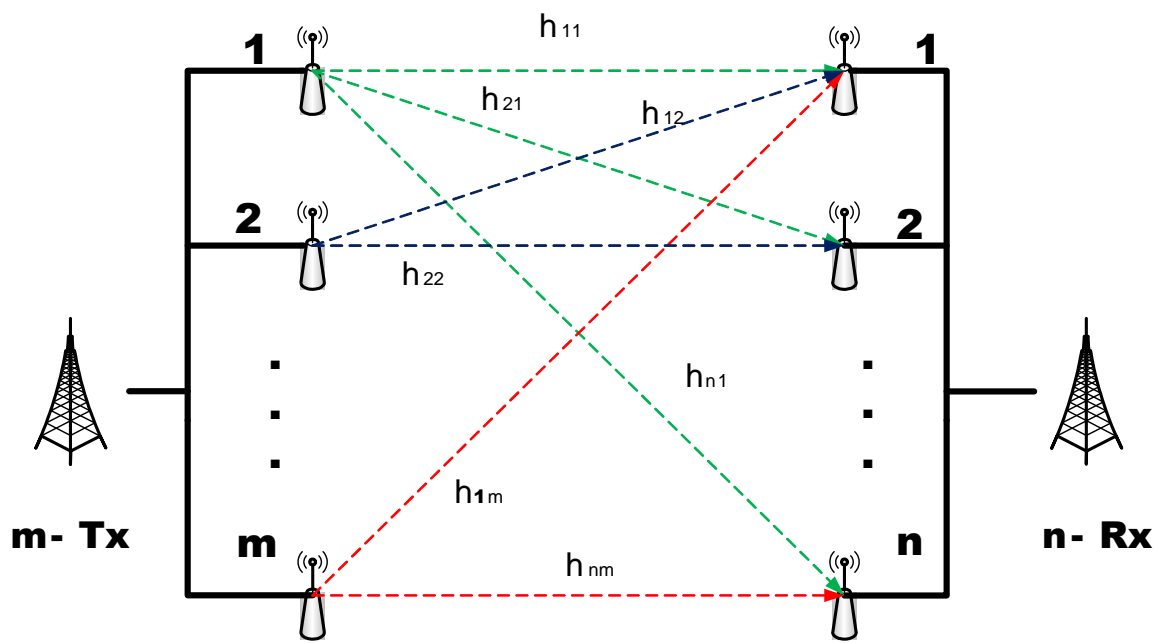


Figura 2.1. Sistema MIMO



Para el estudio de la propagación en sistemas MIMO, se toma el canal como una matriz de transmisión \mathbf{H} de tamaño $m \times n$ donde m es el número de antenas transmisoras y n es el número de antenas receptoras, de esta forma, los elementos h_{ij} corresponden a las funciones de transferencia compleja de la señal enviada por la antena de transmisión j y recibida por la antena de recepción i .

La señal transmitida en un sistema MIMO se ve afectada por la matriz de canal \mathbf{H} y por el ruido que se adiciona durante la transmisión, lo cual se refleja en la ecuación 2.1:

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \quad (2.1)$$

donde \mathbf{y} es el vector señal de tamaño n recibido por el arreglo de antenas, \mathbf{H} es la matriz de tamaño $n \times m$ que representa el canal, \mathbf{s} es el vector de m elementos que representa la señal enviada y \mathbf{n} es el componente de ruido de dimensión n que se adiciona a la señal[12].

Los sistemas MIMO pueden funcionar de acuerdo a dos esquemas: diversidad o multiplexación espacial[13].

2.1.1 Diversidad Espacial

En este esquema la transmisión de la información se realiza enviando por las diferentes antenas del transmisor flujos iguales de información de manera simultánea, obteniendo una redundancia que contrarresta los efectos del desvanecimiento y permite obtener menores valores de BER, brindando mayor fiabilidad a la comunicación.

2.1.2 Multiplexación Espacial

En la multiplexación espacial los símbolos de información a transmitir se dividen entre el número de antenas del transmisor, generando igual número de subcadenas simultáneas en la misma banda de frecuencias, en las cuales cada antena transmisora pone un identificador para determinar en recepción cuál antena es la fuente de la información recibida. El receptor tiene que realizar procesos de reconstrucción del mensaje a partir de la superposición de señales que lleguen. La multiplexación espacial aumenta la capacidad de transmisión del sistema pero incrementa la complejidad en recepción.

Las técnicas más utilizadas en la multiplexación espacial son la codificación Vertical Espacio Temporal por Capas del Laboratorio Bell (V-BLAST, *Vertical-Bell Labs Layered Space Time*), codificación Horizontal Espacio Temporal (H-BLAST, *Horizontal-Bell Labs Layered Space Time*) y la codificación Diagonal Espacio Temporal por Capas del Laboratorio Bell (D-BLAST, *Diagonal-Bell Labs Layered Space Time*) [9].



V-BLAST: el flujo inicial de datos se divide y se reparte entre las antenas transmisoras, cada uno de estos flujos se codifica y se modula a la constelación utilizada de forma independiente y se transmiten simultáneamente.

D-BLAST: divide el flujo de datos de forma diagonal, de tal manera que cada subcadena generada se transmite por cada una de las antenas. Esta técnica presenta un uso inadecuado del canal al inicio y final de cada transmisión ya que no transmiten todas las antenas. En cuanto a la recepción se presenta propagación de error porque la detección se realiza de tal forma que los primeros símbolos detectados se toman como referencia para la detección de los siguientes.

H-BLAST: el flujo de datos se divide en subcadenas que se codifican en forma separada, la matriz generadora se divide en capas, donde cada capa corresponde a un modulador específico, que posteriormente entrega los datos a su respectiva antena.

2.1.3 Diagrama en Bloques de un Sistema MIMO

Un sistema MIMO se compone de diferentes subsistemas, en la figura 2.2 se observa el diagrama de bloques de un sistema MIMO, donde la transmisión se compone de los bloques generador de datos, codificador de canal, modulador, demultiplexor espacial y un arreglo de antenas y la recepción por los bloques detector, multiplexor espacial, demodulador decodificador de canal, modulador de BER.

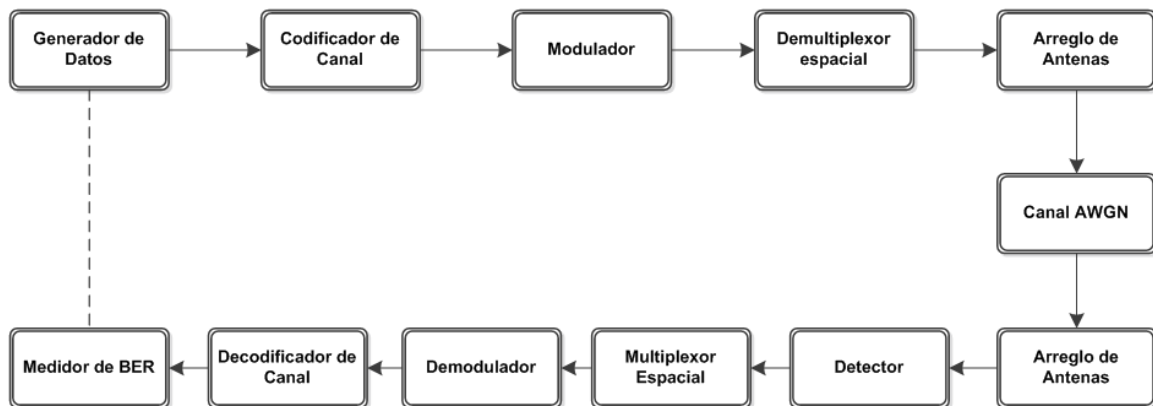


Figura 2.2. Diagrama en bloques de un sistema MIMO

2.1.4 Modulación

El bloque modulador en un sistema MIMO aplica técnicas que permiten un mejor aprovechamiento del canal y mejorar la resistencia contra posibles ruidos o interferencias. Entre estas técnicas se tiene QAM y MQAM.



2.1.4.1 Modulación por Amplitud en Cuadratura

La Modulación QAM es una técnica altamente eficiente y ampliamente usada en comunicaciones inalámbricas, consiste en modular por desplazamiento en amplitud de forma independiente dos portadoras ortogonales en la misma frecuencia para hacer un mejor uso del ancho de banda y puede o no variar la amplitud de las diferentes portadoras.

La representación matemática de la modulación QAM viene dada por:

$$A_n \cos(wt) + B_n \sin(wt), \quad (2.2)$$

donde A_n y B_n son las amplitudes de las dos señales moduladas, y toman de forma independiente los valores discretos correspondientes al total de los estados de la señal moduladora.

La modulación QAM se usa para la transmisión de datos a alta velocidad por canales con poca capacidad. QAM modula los símbolos con una única frecuencia; cabe resaltar que la mitad de los símbolos se modulan con la frecuencia en fase y la otra mitad con la misma frecuencia pero desfasada 90° , así como resultado se tienen dos componentes ($\cos(wt)$ y $\sin(wt)$) a las cuales se les modifica la amplitud dando como resultado una señal modulada en QAM óptima para ser transmitida. Esta modulación se emplea actualmente en sistemas de transmisión de televisión, microondas, satélite y sistemas de telefonía celular.

Para M-QAM, los bits se codifican en M estados que se grafican como puntos uniformemente dispuestos en una rejilla cuadrada con igual separación vertical y horizontal, formando el diagrama de constelación mostrado en la figura 2.3

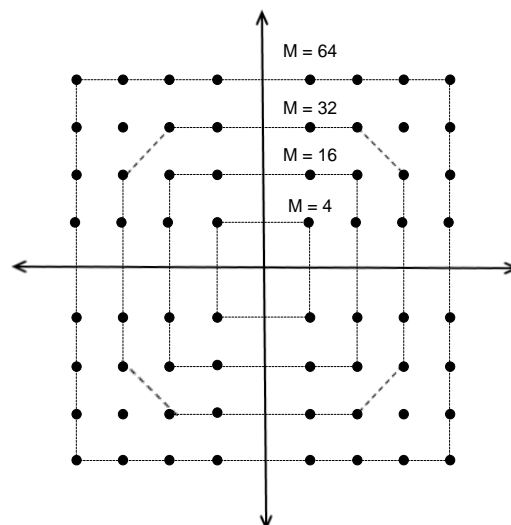


Figura 2.3. Constelaciones M-QAM



2.1.4.2 Modulación de Amplitud en Cuadratura de 16 Estados

En la figura 2.4 se muestra la constelación para la modulación 16QAM; la cual consta de 16 símbolos de 4 bits cada uno.

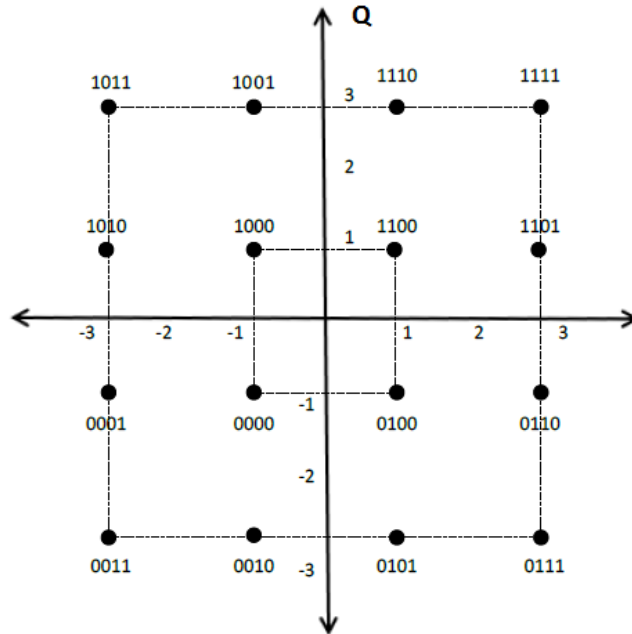


Figura 2.4. Esquema de modulación 16QAM

La posición en la constelación de los símbolos se denota por cuatro bits; los primeros indican en que cuadrante se ubican y los últimos dos la posición dentro del cuadrante, lo que facilita la detección del símbolo transmitido. Para calcular la probabilidad de error en M-QAM se usa la fórmula de probabilidad de error de símbolo, descrita para este tipo de modulación por la siguiente ecuación[14]:

$$P_e \cong 2 \left(1 - \frac{1}{\sqrt{M}} \right) \text{ferc} \left(\sqrt{\frac{E_b}{N_0}} \right), \quad (2.3)$$

donde M es igual al número de símbolos, ferc es la función de error complementario de Gauss, Eb es la energía de bit y No la densidad espectral de potencia de ruido.

2.1.5 Detección en Sistemas MIMO

Uno de los procesos más críticos de las comunicaciones es la detección y para hacer más eficiente este proceso se utilizan algoritmos que pueden ser óptimos o sub-óptimos. Los algoritmos óptimos ofrecen un alto desempeño pero son de gran complejidad computacional y los algoritmos sub-óptimos presentan una complejidad computacional menor a costa de un menor desempeño [1].



Algunos de los algoritmos de detección más utilizados en los sistemas MIMO son Forzado a Cero (ZF, *Zero Forcing*), algoritmo de Error Mínimo Cuadrático Medio (MMSE, *Minimum Mean Square Error*) o el algoritmo de Máxima Verosimilitud.

El algoritmo ZF divide el canal en subcanales paralelos eliminando la interferencia cocanal pero amplifica el ruido durante el proceso de detección. El algoritmo MMSE limita el efecto del ruido a costa de una ligera interferencia cocanal. El algoritmo ML elige la distancia euclidiana más corta entre el vector recibido \mathbf{y} y el vector mensaje \mathbf{s} , este algoritmo presenta una alta complejidad de implementación, que aumenta de manera exponencial con el número de antenas y el orden de la modulación. Otro algoritmo ampliamente usado es el algoritmo esférico que establece una esfera centrada en el símbolo recibido para reducir la complejidad de la búsqueda.

2.1.6 Algoritmo de Máxima Verosimilitud - ML

En las comunicaciones, el receptor frecuentemente observa una superposición lineal de símbolos de información, esto hace que la principal dificultad sea separar e identificar los símbolos transmitidos, es decir, recuperar correctamente el vector enviado desde cada antena del transmisor.

En un sistema MIMO, desde el transmisor se envía la información en un vector de símbolos modulados como el que se muestra en la ecuación 2.4.

$$\mathbf{s} = (s_1, s_2, s_3, \dots, s_n), s_m \in \mathbf{S}, \quad (2.4)$$

siendo \mathbf{S} la constelación con 2^b elementos, donde b es el número de bits que corresponde a un símbolo en s_m y n es el número de antenas transmisoras.

El vector recibido que contiene la información transmitida alterada sólo por los efectos del canal, sin la adición de ruido, es el expresado en la ecuación 2.5.

$$\mathbf{a} = \mathbf{H}\mathbf{s} \quad (2.5)$$

Después de recibir cada vector \mathbf{y} , el receptor debe estimar cuál de los posibles vectores \mathbf{s} fue transmitido haciendo uso de la información recibida y del conocimiento de la matriz de canal \mathbf{H} .

La ecuación 2.6 indica matemáticamente cómo el algoritmo ML escoge la estimación $\hat{\mathbf{s}}$ como el vector elemento de \mathbf{S} más cercano a \mathbf{y} [15]; dicha expresión también se conoce como criterio de Máxima Verosimilitud.

$$\hat{\mathbf{s}} = \underset{\mathbf{s} \in \mathbf{S}}{\operatorname{argmin}} \|\mathbf{s} - \mathbf{y}\|^2 \quad (2.6)$$

La ecuación 2.7 es otra manera de escribir el criterio de máxima verosimilitud.



$$\min_{\mathbf{s} \in \mathbf{S}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\| \quad (2.7)$$

Donde \mathbf{H} es la matriz que representa el canal MIMO, \mathbf{y} es la señal recibida y \mathbf{s} es el vector transmitido, usando la descomposición de matrices QR sobre la matriz del canal, se reformula el criterio según se muestra en las ecuaciones 2.8, 2.9 y 2.10 [15]:

$$\min_{\mathbf{s} \in \mathbf{S}} \|\hat{\mathbf{y}} - \mathbf{L}\mathbf{s}\|, \text{ Donde } \hat{\mathbf{y}} \triangleq \mathbf{Q}^T \mathbf{y} \quad (2.8)$$

$$\min_{\{s_1, \dots, s_{nt}\}} \{f_1(s_1) + f_2(s_1, s_2) + \dots + f_n(s_1, \dots, s_{nt})\}, \quad (2.9)$$

donde,

$$f_k(s_1, \dots, s_k) \triangleq \left(\hat{y}_k - \sum_{l=1}^k l_{k,l} s_l \right)^2 \quad (2.10)$$

El criterio de máxima verosimilitud se representa como un árbol de decisión con $m + 1$ capas, $|\mathbf{S}|$ ramas saliendo de cada nodo intermedio y $|\mathbf{S}|^m$ nodos hoja.

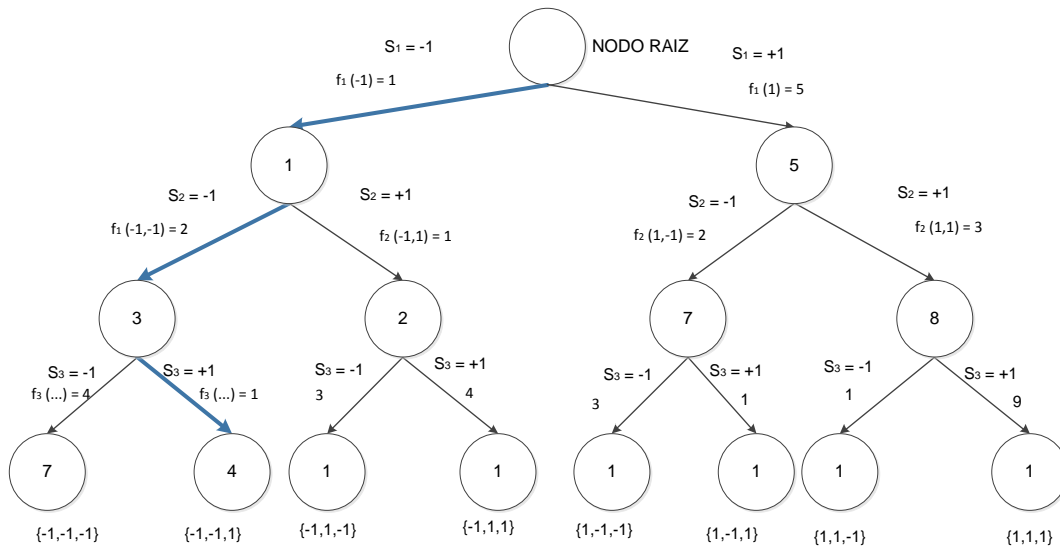


Figura 2.5. Árbol de decisión ML



En el árbol de decisión cada rama tiene dos indicadores, una métrica asociada y una métrica acumulada. La métrica acumulada es la suma de las métricas de todas las ramas por las que se pasa al ir desde el nodo raíz hasta el nodo final [1]. En la figura 2.5 se muestra el camino con la menor métrica acumulada que permite determinar el símbolo enviado.

2.2 ALGORITMOS *LATTICE REDUCTION-AIDED* - LRA

Un algoritmo LRA es un algoritmo que hace uso de las técnicas de reducción *lattice* para alcanzar un mejor desempeño en la detección de señales en sistemas MIMO. La utilización de algoritmos LRA para la detección de señales implica un replanteamiento del sistema MIMO como lo muestra la ecuación 2.11, donde \mathbf{T} es la matriz de transformación de la reducción, \mathbf{H}' es la matriz reducida que se obtiene a partir de \mathbf{HT} y \mathbf{z} es el equivalente de \mathbf{s} para la matriz reducida.

$$\mathbf{y} = \mathbf{HTT}^{-1}\mathbf{s} + \mathbf{n} = \mathbf{H}'\mathbf{z} + \mathbf{n} \quad (2.11)$$

En la figura 2.6 se presenta el sistema MIMO equivalente según la ecuación 2.11, donde se adicionan los bloques correspondientes a la matriz de transformación y su inversa para tener nuevamente la solución en el dominio original.

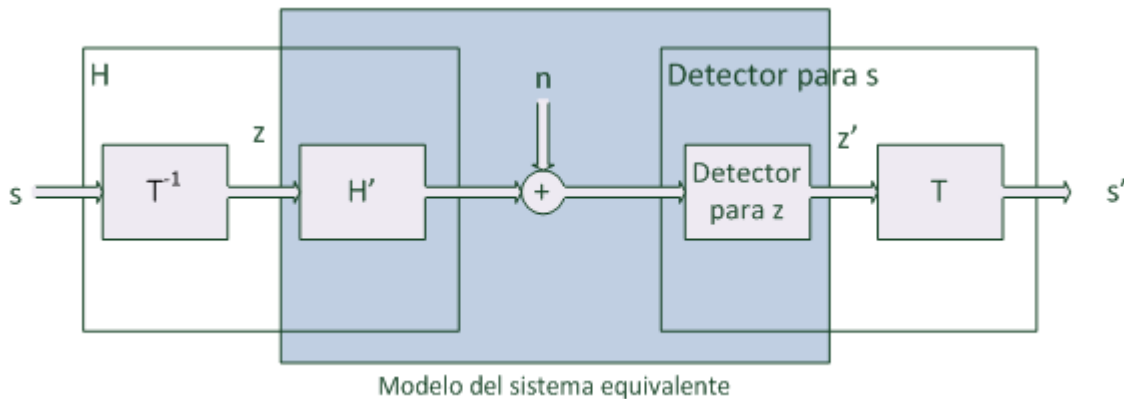


Figura 2.6. Sistema MIMO equivalente para algoritmos *Lattice Reduction-Aided*

Los problemas de codificación y detección pueden ser abordados usando herramientas de la teoría de *lattices*, ya que las técnicas de reducción *lattice* aprovechan la naturaleza discreta de los datos transmitidos y utilizan la matriz de canal que afecta los datos como un generador (base) de una *lattice* de puntos. Sin importar el esquema de utilizado en la transmisión, (diversidad o multiplexación espacial) y siempre y cuando se conozca la matriz de canal o base, ésta se puede reducir para facilitar el proceso de detección mejorando su tolerancia al ruido como se muestra en



la figura 2.7, donde los círculos verdes muestran la distancia máxima a la que un símbolo puede ser detectado correctamente. En el plano izquierdo se puede observar que una base ortogonal (azul) genera áreas de detección más grandes que las generadas por una base no ortogonal (negro) como la observada en el plano derecho.

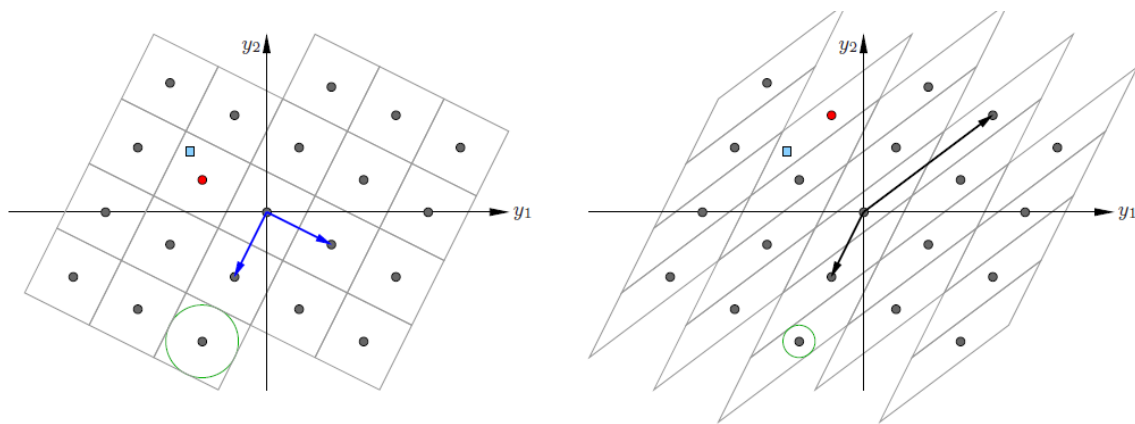


Figura 2.7. Reducción *Lattice*

La aplicación de las técnicas de reducción *lattice* en la detección de señales en sistemas MIMO implica un conocimiento de la matriz de canal en el receptor y comprende tres pasos generales:

1. Determinar una base mejorada para la *lattice* por medio de una reducción que parte del canal. La base original y la mejorada están relacionadas por una matriz unimodular².
2. Resolver el problema con respecto a la base reducida.
3. Transformar la solución al dominio original usando la matriz unimodular. En la figura 2.8 se muestra gráficamente la transformación de las matrices desde el dominio de la reducción *lattice* al dominio original al multiplicar por la inversa de la matriz unimodular.

² Una matriz unimodular es una matriz invertible con determinante ± 1

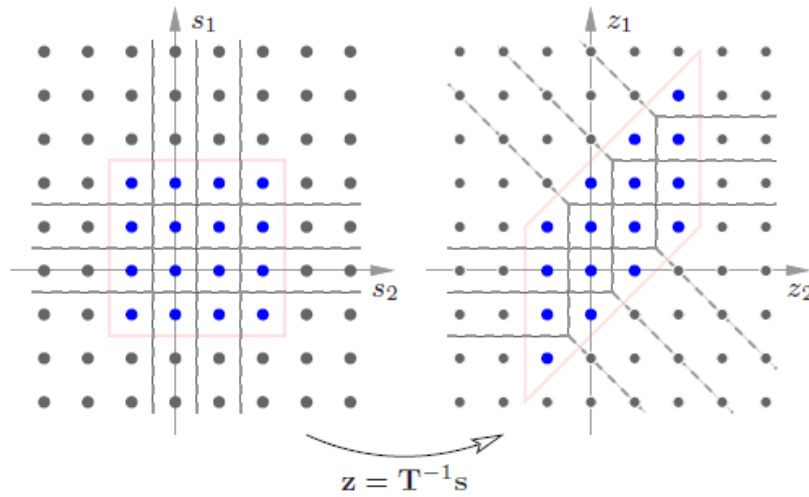


Figura 2.8. Transformación de la solución al dominio original

Las técnicas de reducción *lattice* buscan reducir las dimensiones de las bases con vectores base más cortos y más ortogonales, siempre buscando la mejor relación entre la calidad de las bases reducidas y el esfuerzo computacional requerido para encontrarlas.

El objetivo de los algoritmos de reducción *lattice* es encontrar la secuencia de operaciones elementales (equivalente a una matriz unimodular \mathbf{T}) que transforme una base dada \mathbf{B} en una base reducida $\mathbf{B}' = \mathbf{B}\mathbf{T}$, de acuerdo a los requerimientos específicos del criterio de reducción [16].

2.2.1 Lattices

Una *lattice* es un arreglo periódico de puntos discretos, un grupo aditivo discreto de R^n que se puede caracterizar en términos de un conjunto de $m \leq n$ vectores columna (base generadora) linealmente independientes [16].

Teniendo los vectores generadores como se expresan en la ecuación 2.12.

$$\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_m\}, \quad \mathbf{b}_l \in R^n \quad (2.12)$$

Se puede definir una *lattice* \mathbf{L} como en la ecuación 2.13.

$$\mathbf{L} \triangleq \left\{ x \mid x = \sum_{l=1}^m z_l \mathbf{b}_l, \right\} \quad z_l \in \mathbf{Z}, \quad (2.13)$$



Para cualquier x y y que pertenecen a \mathbf{L} se tiene que $y+kx$ también pertenece a \mathbf{L} para todo k perteneciente a \mathbf{Z} .

Los vectores base de la ecuación 2.12 forman una matriz de $n \times m$ llamada base \mathbf{B} como se puede ver en la ecuación 2.14.

$$\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_m), \quad \mathbf{b} \in \mathbb{R}^n \quad (2.14)$$

Cualquier elemento de la *lattice* puede ser representado como se expresa en la ecuación 2.15.

$$\mathbf{x} = \mathbf{B}\mathbf{z} \quad \mathbf{z} \in \mathbf{Z}^m \quad (2.15)$$

En la figura 2.9 se muestran algunos ejemplos de bases generadoras y sus correspondientes *lattices*.

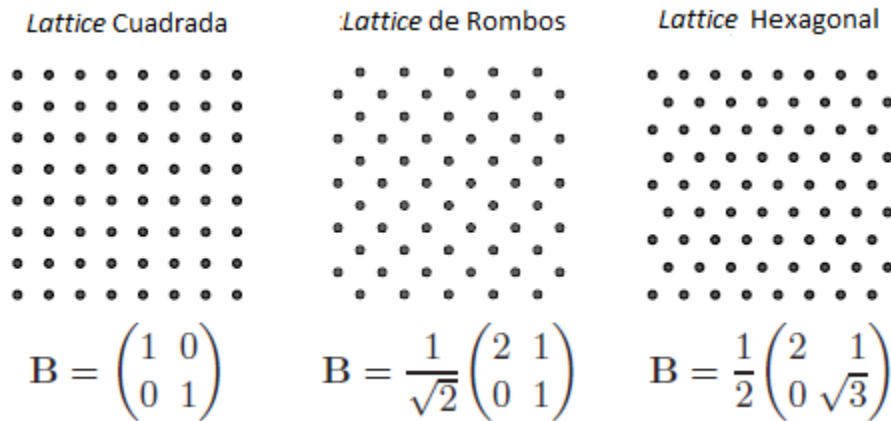


Figura 2.9. Bases generadoras y *lattices*

Para una *lattice* la matriz base no es única, lo cual permite que por medio de los algoritmos de reducción se puedan considerar muchas posibilidades de hallar una nueva matriz base de mejores condiciones para una *lattice* determinada. Para que dos matrices \mathbf{B} y $\mathbf{B}\mathbf{T}_{m \times m}$ generen la misma *lattice*, la matriz \mathbf{T} debe ser unimodular según se muestra en las ecuaciones 2.16 y 2.17.

$$\mathbf{B}\mathbf{Z}^m = \mathbf{B}\mathbf{T}\mathbf{Z}^m \quad (2.16)$$

$$\mathbf{Z}^m = \mathbf{T}\mathbf{Z}^m \quad (2.17)$$



2.2.2 Propiedades de las *Lattices*

Toda *lattice* tiene las siguientes propiedades:

2.2.2.1 Región Fundamental P

La región fundamental es el conjunto de puntos que se pueden escribir según la ecuación 2.18 y que se puede observar en la figura 2.10.

$$x = \sum_{l=1}^m \theta_l \mathbf{b}_l \quad 0 \leq \theta < 1 \quad (2.18)$$

De la ecuación 2.18 se observa que esta propiedad depende de la base generadora, es decir, hay una región fundamental asociada a cada base, pero el volumen (m-dimensional) cubierto por la región fundamental es el mismo para todas.

2.2.2.2 Región Voronoi V

Es el conjunto de los puntos que están más cerca de su punto de origen que de otro punto de la *lattice*, es decir, la región alrededor de cada punto que contiene el área más cercana a él mismo, como lo muestra la figura 2.10.

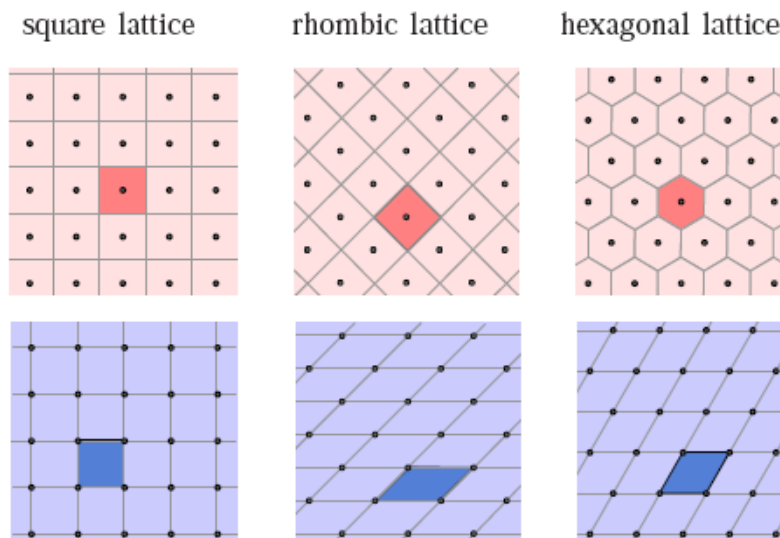


Figura 2.10. Región fundamental (azul), región *voronoi* (rojo) [16]



2.2.2.3 Determinante de una *Lattice*

Se define como el espacio m -dimensional abarcado por las columnas de \mathbf{B} . Es equivalente al volumen (m -dimensional) de la *lattice*, por lo que tiene el mismo valor para todas las bases. Su definición matemática se muestra en la ecuación 2.19.

$$|\mathbf{L}| \triangleq \sqrt{\det(\mathbf{B}^T \mathbf{B})} \quad (2.19)$$

2.2.2.4 Índice de Ortogonalidad

Es la medida de la calidad de la base de una *lattice*, el cual es igual a la unidad si y sólo si \mathbf{B} es ortogonal, de lo contrario toma valores mayores que la unidad. Su definición matemática se muestra en la ecuación 2.20.

$$\varepsilon(\mathbf{B}) = \frac{1}{|\mathbf{L}|} \prod_{l=1}^m \|b_l\| \quad (2.20)$$

2.2.2.5 *Lattice* Dual \mathbf{L}^*

Para cualquier *lattice* se tiene asociada una *lattice* dual definida por la ecuación 2.21.

$$\mathbf{L}^* \triangleq \{x^* \in \text{span}^\ddagger(\mathbf{B}) \mid x^T x^* \in \mathbf{Z} \text{ para todo } x \in \mathbf{L}\} \quad (2.21)$$

Si \mathbf{B} es una base para \mathbf{L} , \mathbf{B}^* lo es para \mathbf{L}^* . La base dual de una *lattice* se puede calcular a partir de su determinante como se muestra en la ecuación 2.22.

$$\mathbf{B}^* = \mathbf{B}(\mathbf{B}^T \mathbf{B})^{-1} |\mathbf{L}^*| = \frac{1}{|\mathbf{L}|} \quad (2.22)$$

2.2.3 Técnicas de Reducción *Lattice*

Existen diferentes técnicas y métodos para encontrar una base reducida, a continuación se listan las más comunes.

[‡] La intersección de todos los sub espacios que contienen un conjunto particular de vectores.



2.2.3.1 Reducción *Minkowski* y *Hermite-Korkine-Zolotareff*

Esta técnica requiere que \mathbf{b}'_1 , de la base ordenada \mathbf{B}' sea el vector más corto de $\mathbf{L}(\mathbf{B}')$. Los demás vectores \mathbf{b}'_l ($2 \leq l \leq m$) son los vectores más cortos en $\mathbf{L}(\mathbf{B}')$ que no son una combinación lineal de $\mathbf{b}'_1, \dots, \mathbf{b}'_{l-1}$.

La proyección del vector base \mathbf{b}'_l sobre el complemento ortogonal del espacio cubierto por $\{\mathbf{b}'_1, \dots, \mathbf{b}'_{l-1}\}$ es el vector más corto de la correspondiente matriz proyectada [17]. El principal inconveniente de esta técnica es que implica una complejidad computacional muy alta.

2.2.3.2 Reducción de Tamaño

Tomando la matriz reducida \mathbf{B}' y aplicándole la descomposición QR se obtiene $\mathbf{B}' = \mathbf{Q}'\mathbf{R}'$. Los elementos de \mathbf{R}' deben satisfacer la condición 2.23.

$$|r'_{k,l}| \leq \frac{1}{2}|r'_{k,k}| \quad 1 \leq k < l \leq m \quad (2.23)$$

El componente de cualquier vector \mathbf{b}'_l en la dirección de \mathbf{q}_k para $k < l$ no es más largo que la mitad de \mathbf{b}'_k . Esta reducción se considera muy débil y de baja calidad[16].

2.2.3.3 Reducción *Gauss*

Está restringida a matrices de rango $m = 2$, satisface las condiciones de *Minkowski* y *Hermite-Korkine-Zolotareff* realizando una reducción de tamaño y luego un intercambio de columnas hasta que \mathbf{b}'_1 sea más corto que \mathbf{b}'_2 , donde \mathbf{b}'_1 y \mathbf{b}'_2 son los vectores más cortos en la *lattice* \mathbf{L} que forman una base para \mathbf{L} [18].

2.2.3.4 Reducción *Lenstra Lenstra Lovász*

Es una extensión de la técnica de reducción *Gauss* para *lattices* de rango $m > 2$. Una matriz base \mathbf{B}' a la cual se le aplica descomposición QR es LLL reducida si se satisfacen las desigualdades 2.24 y 2.25.

$$|r'_{k,l}| \leq \frac{1}{2}|r'_{k,k}| \quad 1 \leq k < l \leq m \quad (2.24)$$

(Reducción de Tamaño)



$$\delta |r'_{l-1,l-1}|^2 \leq |r'_{l,l}|^2 + |r'_{l-1,l}|^2 \quad \text{para } l = 2, \dots, m \quad (2.25)$$

(Condición de Lovász)

El parámetro δ afecta la calidad de la base reducida, entre más alto, mejor es la calidad y mayor la complejidad, el valor típico de este parámetro es $\frac{3}{4}$ y varía entre $\frac{1}{4} < \delta \leq 1$.

Si dos vectores \mathbf{b}'_{l-1} y \mathbf{b}'_l no cumplen la condición de *Lovász*, se intercambian y la matriz resultante se descompone en QR y se reduce.

Este proceso se repite hasta que se cumplan las condiciones expresadas en las desigualdades 2.24 y 2.25 [19].

2.2.3.5 Reducción *Lattice Dual*

Se realiza la reducción a la base dual \mathbf{B}^* por una matriz \mathbf{T} , lo que es equivalente a reducir \mathbf{B} por \mathbf{T}^{-1} . En la mayoría de los casos su utilización brinda mejores factores de proximidad [20], los cuales fijan los límites superiores del desempeño de los algoritmos LRA con relación al algoritmo ML [21]. Esta técnica no es un algoritmo de reducción como tal si no un método que consiste en aplicar la técnica de reducción a la base dual en lugar de a la base original.

2.2.3.6 Reducción *Lattice Seysen*

El principio básico de esta reducción consiste en reducir simultáneamente \mathbf{B} y \mathbf{B}^* . El criterio de ortogonalidad se define en la ecuación 2.26.

$$S(\mathbf{B}) = \sum_{l=1}^m \|\mathbf{b}_l\|^2 \|\mathbf{b}_l^*\|^2 \quad (2.26)$$

Si y sólo si \mathbf{B} es ortogonal, $S(\mathbf{B}) = m$.

Se dice que una matriz es S-reducida cuando $S(\mathbf{B}) \leq S(\mathbf{BT})$ para cualquier \mathbf{T} [22].



2.2.3.7 Algoritmo de Brun

Algoritmo creado para encontrar vectores más cortos y ortogonales a un vector dado \mathbf{u} . Este algoritmo de reducción *lattice* se aplica realizando cambios iterativos en las columnas, su desempeño es bajo, pero es poco complejo.

2.3 COMPARACIÓN ALGORITMOS LRA

El estudio de las diferentes técnicas de reducción *lattice* utilizadas más comúnmente en la actualidad y la recolección de información referente a investigaciones realizadas [17], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33] y [34] sobre estas técnicas permitió elaborar la tabla comparativa 2.1. En esta tabla se tuvo en cuenta para cada técnica el nivel de complejidad, la eficiencia en la reducción, los entornos en que han sido utilizadas y las principales aplicaciones que se le han dado a cada una de ellas.

Tabla 2-1. Tabla comparativa de algoritmos LRA

TÉCNICA	COMPLEJIDAD	REDUCCIÓN	ENTORNOS DE PRUEBA	APLICACIONES
Minkowski	Alta	Media	Simulación	Detección MIMO
Reducción de Tamaño	Baja	Baja	Simulación, FPGA	Detección MIMO, Generador de bases para otras técnicas de reducción.
Reducción Gauss	Baja	Media	Simulación	Detección MIMO, Generador de bases para otras técnicas de reducción.
Lenstra Lenstra Lovász	Media	Alta	Simulación, FPGA	Detección MIMO, Criptoanálisis.
Seysen	Media	Alta	Simulación	Detección MIMO, Problemas de suma de subconjuntos.
Algoritmo de Brun	Baja	Baja	Simulación	Análisis matemáticos

De acuerdo con la información provista en la tabla, se escogieron las técnicas de reducción LLL y *Seysen* para realizar las implementaciones y pruebas necesarias para cumplir a cabalidad los objetivos propuestos por este trabajo de grado. La decisión se tomó teniendo en cuenta la mejor relación entre la complejidad y la capacidad de reducción, así como la ventaja que presenta la técnica LLL al ser una de las más estudiadas y por lo tanto la que cuenta con más información



disponible referente a implementaciones y análisis de desempeño, por su parte la técnica *Seysen* es una de las más utilizadas en la actualidad y representa un campo de estudio prometedor.

En cuanto a las otras técnicas estudiadas, se encontró que algunas tenían una inconveniente relación entre complejidad y capacidad de reducción como la técnica *Minkowski* o Reducción de *Gauss*, siendo la técnica *Minkowski* la que requiere mayor capacidad computacional para su implementación. Las técnicas restantes Reducción de Tamaño y el Algoritmo de *Brun* no presentan altos requerimientos computacionales, pero así mismo su capacidad de reducción no es la esperada para que sean parte de este estudio, y como desventaja adicional, no es común la aplicación del algoritmo de *Brun* en los sistemas de detección MIMO que son el tema de estudio de este trabajo de grado.

2.4 ALGORITMO LENSTRA LENSTRA LOVÁSZ

En 1982 *Hendrik Lenstra*, *Arjen Lenstra* y *László Lovász* publicaron un artículo titulado “*Factoring Polynomials with Rational Coefficients*” donde describían una nueva técnica de reducción *lattice* a la que posteriormente llamarían LLL [19] y que es una extensión del algoritmo de reducción de *Gauss*. El algoritmo LLL determina que una base está reducida si cumple con la condición de Reducción de Tamaño mostrada en la desigualdad 2.24 y con la condición de *Lovász* mostrada en la desigualdad 2.25. La complejidad de este algoritmo es variable y dependiente del parámetro δ [16], cuyos valores varían entre $\frac{1}{4}$ y 1, donde $\frac{1}{4}$ implica la menor complejidad y 1 la mayor complejidad y precisión [16].

2.4.1 Procedimiento Algoritmo *Lenstra Lenstra Lovász*

El proceso general que usa este método para reducir una matriz se explica a continuación:

1. Definir el parámetro δ de reducción y descomponer la matriz base $\mathbf{B} = \mathbf{QR}$, donde \mathbf{Q} es una matriz ortogonal y \mathbf{R} es una matriz triangular superior.
2. Evaluar la condición de Reducción de Tamaño para un par de vectores \mathbf{r}_i y \mathbf{r}_j . Si no se cumple se reduce el vector j según la ecuación 2.27.

$$\mathbf{r}_i = \mathbf{r}_i - [U_{ij}] \mathbf{r}_j, \quad (2.27)$$

donde U_{ij} es un coeficiente de Gram Schmidt y se calcula con la ecuación 2.28.



$$\mu_{i,j} = \left\lfloor \frac{r_{i,j}}{r_{i,i}} \right\rfloor \quad (2.28)$$

3. Reflejar los cambios hechos a la matriz base en la matriz de transformación \mathbf{T} como se muestra en la ecuación 2.29.

$$\mathbf{t}_i = \mathbf{t}_i - [U_{ij}] \mathbf{t}_j \quad (2.29)$$

4. Evaluar para el par de vectores \mathbf{r}_i y \mathbf{r}_j la condición de *Lovász* mostrada en la desigualdad 2.25. De no cumplir con la condición de *Lovász* los vectores \mathbf{r}_i y \mathbf{r}_j se intercambian como se muestra en la ecuación 2.30.

$$\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_i, \mathbf{r}_j, \dots, \mathbf{r}_m] \rightarrow \mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_j, \mathbf{r}_i, \dots, \mathbf{r}_m] \quad (2.30)$$

5. Reflejar los cambios hechos a la matriz \mathbf{R} en la matriz \mathbf{T} como se muestra en la ecuación 2.31.

$$\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_i, \mathbf{t}_j, \dots, \mathbf{t}_m] \rightarrow \mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_j, \mathbf{t}_i, \dots, \mathbf{t}_m] \quad (2.31)$$

6. Aplicar la rotación de *Givens* [33] a la matriz \mathbf{R} para recuperar su forma triangular superior según lo indicado en la ecuación 2.32.

$$\mathbf{R} \leftarrow \Theta_i \mathbf{R} \quad (2.32)$$

Al realizar esta modificación en la matriz \mathbf{R} , la matriz \mathbf{Q} se debe transformar como en la expresión 2.33 de tal forma que se conserve la igualdad $\mathbf{B} = \mathbf{QR}$.

$$\mathbf{Q} \leftarrow \mathbf{Q} \Theta_i^T \quad (2.33)$$



Donde Θ_i es la rotación ortogonal de la matriz como se muestra en la ecuación 2.34.

$$\Theta_i = \begin{bmatrix} \alpha_i & \beta_i & 0 & 0 \\ -\beta_i & -\alpha_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.34)$$

Donde los elementos α_i y β_i son representadas por las ecuaciones 2.35 y 2.36.

$$\alpha_i = \frac{r_{i,i}}{\sqrt{r_{i,i}^2 + r_{j,i}^2}} \quad (2.35)$$

$$\beta_i = \frac{r_{j,i}}{\sqrt{r_{i,i}^2 + r_{j,i}^2}} \quad (2.36)$$

Es necesario repetir este proceso iterativamente para todos los vectores de la matriz \mathbf{R} hasta satisfacer la condición de *Lóvasz*.

2.5 ALGORITMO SEYSEN

Martin Seysen propuso este nuevo método de reducción *lattice* cuando buscaba una manera de reducir simultáneamente una *lattice* y su dual [35]. El Algoritmo de *Seysen* se diferencia del algoritmo LLL y sus variantes en que éste considera simultáneamente todos los vectores de la *lattice*, y las operaciones de renglones que realiza se caracterizan porque reducen tanto la *lattice* original como su dual [36].

Está basado en una medida de ortogonalidad que es diferente del índice de ortogonalidad, llamada Medida de Ortogonalidad de *Seysen* [35], la cual sirve para saber si una base cumple o no con el criterio de reducción de *Seysen* representado matemáticamente por la ecuación 2.37.

$$S(\mathbf{A}) = \sum_{i=1}^m \|\mathbf{b}_i\|^2 \|\mathbf{b}_i^*\|^2 \quad (2.37)$$



2.5.1 Criterio de Reducción de *Seysen*

Suponiendo que $L(\mathbf{B})$ y $L^*(\mathbf{B})$ corresponden a una *lattice* y su dual, las cuales se quieren reducir simultáneamente, se definen \mathbf{A} y \mathbf{A}^* como las formas cuadráticas asociadas de $L(\mathbf{B})$ y $L^*(\mathbf{B})$, respectivamente y se calculan a partir de las ecuaciones 2.38 y 2.39.

$$\mathbf{A} = [a_{i,j}]_{1 \leq i,j \leq n} = [(\mathbf{b}_i, \mathbf{b}_j)]_{1 \leq i,j \leq n} \quad (2.38)$$

$$\mathbf{A}^* = [a_{i,j}^*]_{1 \leq i,j \leq n} = [(\mathbf{b}_i^*, \mathbf{b}_j^*)]_{1 \leq i,j \leq n} \quad (2.39)$$

Si $L(\mathbf{B})$ es la *lattice* definida por la base \mathbf{B} , entonces cualquier otra base \mathbf{B}' de $L(\mathbf{B})$ puede ser escrita como en la ecuación 2.40.

$$\mathbf{B}' = \mathbf{B}\mathbf{T}, \quad (2.40)$$

donde \mathbf{T} es una matriz entera unimodular con determinante igual a la unidad [33]. De (2.38), la forma cuadrática asociada de \mathbf{B}' puede ser escrita como la ecuación 2.41:

$$\mathbf{A}' = \mathbf{T}^t \mathbf{A} \mathbf{T} \quad (2.41)$$

Se dice que una base cumple con el criterio de reducción de *Seysen* si cumple con la desigualdad 2.42.

$$S(\mathbf{A}) \leq S(\mathbf{T}^t \mathbf{A} \mathbf{T}) \text{ para todo } \mathbf{T} \quad (2.42)$$

Para obtener la matriz de transformación \mathbf{T} óptima para una *lattice* $L(\mathbf{B})$, el tipo de matrices de transformación \mathbf{T} está definido por las ecuaciones 2.43 y 2.44.

$$\mathbf{T}_{i,j}^{\lambda_{ij}} = \mathbf{I}_m + \lambda_{ij} \mathbf{U}_{ij} \quad i \neq j, \quad \lambda \in \mathbf{Z} \quad (2.43)$$



$$\mathbf{U}_{ij} = [u_{k,l}]_{1 \leq k, l \leq n}, \quad u_{k,l} = \begin{cases} 1, & \text{si } k = i \text{ y } l = j, \\ 0, & \text{si } k \neq i \text{ y } l \neq j. \end{cases} \quad (2.44)$$

Donde \mathbf{I}_m es la matriz identidad, la matriz \mathbf{U}_{ij} tiene sólo un elemento diferente de cero, $\mathbf{T}_{ij}^{\lambda_{ij}}$ es una matriz con todos los elementos de su diagonal iguales a 1, y sólo un elemento fuera de la diagonal diferente de cero, y λ_{ij} esta dada por la ecuación 2.45:

$$\lambda_{ij} = \text{round}^{\S} \left\{ \frac{1}{2} \cdot \left(\frac{a_{i,j}^*}{a_{j,j}^*} - \frac{a_{i,j}}{a_{i,i}} \right) \right\} \quad (2.45)$$

Donde λ_{ij} minimiza la medida de ortogonalidad de *Seysen* para el par de columnas escogidas i, j [33].

2.5.2 Procedimiento Algoritmo de *Seysen*

El algoritmo de *Seysen* obtiene una base reducida disminuyendo de forma iterativa la medida de ortogonalidad de *Seysen*[32]. Dentro de cada iteración se realizan los siguientes pasos:

1. Calcular Lambda:

Se calculan los valores de λ_{ij} utilizando la ecuación 2.45 para todos los pares de índices i, j excepto cuando i es igual a j .

2. Calcular Delta:

Para cada valor de λ_{ij} cuantificar la reducción de la medida de ortogonalidad de *Seysen* con la ecuación 2.46.

$$\Delta_{ij} = \mathbf{S}(\mathbf{A}) - \mathbf{S}(\mathbf{A}') \quad (2.46)$$

3. Seleccionar índices:

Se escogen el par de índices i, j que permiten la mayor reducción de la medida de ortogonalidad de *Seysen*.

[§] Aproxima al entero más cercano.



4. Actualizar la base:

Con los índices que minimizan la medida de *Seysen* se realiza la actualización de la base.

El algoritmo termina cuando todos los valores de λ_{ij} son iguales a cero para todo $1 \leq i, j \leq m$ alcanzando un mínimo de la medida de ortogonalidad de *Seysen*.

3 METODOLOGÍA

Este capítulo está dedicado a la descripción de la metodología escogida para llevar a cabo el cumplimiento de los objetivos propuestos por este trabajo de grado.

3.1 METODOLOGÍA PARA EL DESARROLLO DE SIMULACIONES

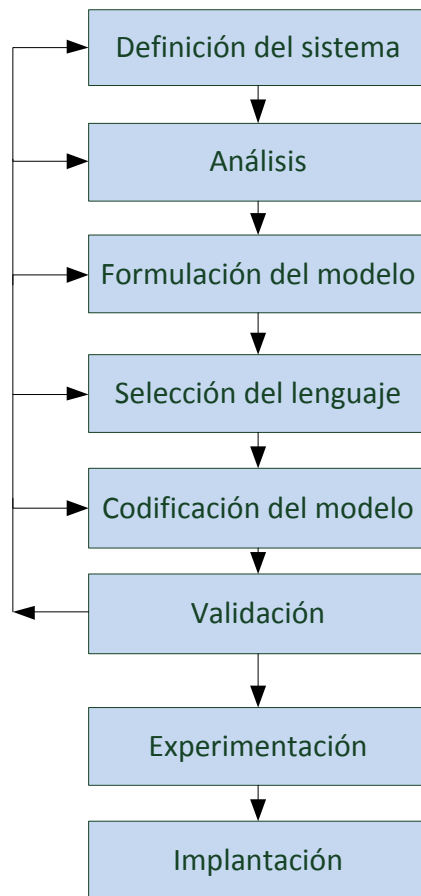


Figura 3.1. Fases de la metodología

La metodología que se utilizó para el desarrollo de este trabajo fue la Metodología para el Desarrollo de Simulaciones de Sistemas de Telecomunicaciones [37]. Esta es una metodología



iterativa que consta de ocho fases: Definición del Sistema, Análisis, Formulación del Modelo, Selección del Lenguaje, Codificación del Modelo, Validación, Experimentación e Implantación como se puede ver en la figura 3.1. Es progresiva ya que permite entrar en un modo cíclico si en la fase de validación se determina que los resultados no son correctos. Al entrar en el modo cíclico es posible volver en primera instancia a la fase de codificación del modelo y corregir los errores encontrados. Si no se obtienen resultados coherentes es necesario volver a la fase de formulación del modelo y revisar si el sistema ha sido modelado sobredimensionando sus características o limitándolas más de lo necesario. Si el error o los resultados incoherentes son persistentes es necesario volver al inicio y revisar la definición del sistema.

3.1.1 Fase 1: Definición del Sistema

Sistema MIMO con dos antenas en transmisión y dos antenas en recepción (sistema 2x2) ó cuatro antenas en transmisión y cuatro antenas en recepción (sistema 4x4), modulación 4QAM ó 16QAM y se utiliza un canal AWGN con el fin de analizar la etapa de detección del sistema con dos algoritmos de reducción *lattice*: el algoritmo *Seysen* y el algoritmo LLL.

3.1.2 Fase 2: Análisis

Es necesario describir los estados y las interacciones lógicas entre las variables utilizadas para evaluar desempeño y capacidad del sistema.

Estado del sistema: se define en el receptor en un instante de tiempo cualquiera de funcionamiento por medio de variables principales y secundarias, donde las variables principales son aquellas que afectan directamente los resultados y las secundarias son aquellas que complementan el modelado del sistema. A continuación se presentan las variables:

Variables Principales

- BER.
- Tiempo de simulación.
- SNR.

Variables Secundarias

- Número de tramas.
- Número de antenas.
- Orden de la modulación.



- Técnica de reducción.
- Algoritmo de detección.

Actividad: la actividad principal del sistema es la detección de los símbolos enviados por los algoritmos de detección y las técnicas de reducción. El tiempo necesario para ejecutar esta operación varía en función del algoritmo utilizado y del número de bits recibidos, los cuales están determinados por las variables número de antenas, orden de modulación y número de tramas.

Entidad: es cualquier objeto utilizado dentro del sistema y según su función puede ser estático o dinámico. En el sistema los objetos estáticos son el modulador, el canal y el demultiplexor, mientras que los objetos dinámicos son las tramas enviadas.

3.1.3 Fase 3: Formulación del Modelo

Teniendo en cuenta el análisis realizado, se modela el sistema mediante el diagrama de bloques de la figura 3.2. La formulación del modelo se complementa con diagramas de flujo que muestran el comportamiento de los algoritmos *lattice* y del sistema en general.

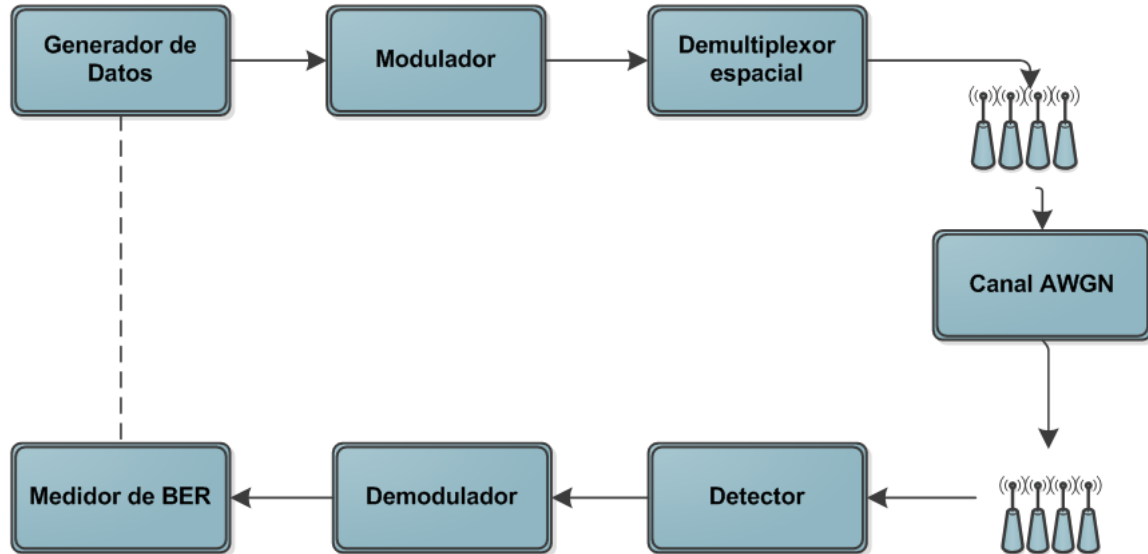


Figura 3.2. Diagrama de bloques sistema MIMO

- **Generador de datos:** Proporciona la información aleatoria a transmitir por cada simulación.



- **Demultiplexor espacial:** Define la forma en que se envía la información por el arreglo de antenas transmisoras.
- **Modulador:** Modula los datos a transmitir utilizando los esquemas 4QAM ó 16QAM.
- **Canal:** Modela el canal AWGN.
- **Detector:** Determina los símbolos enviados a través del sistema por medio de algoritmos de detección. Se supone conocimiento perfecto del canal.
- **Demodulador:** Realiza el proceso de demodulación de los datos detectados según el esquema utilizado en el bloque modulación.
- **Medidor de BER:** Determina los valores de BER según la información detectada.

3.1.3.1 Diagramas de Flujo

Como parte de la formulación del modelo se incluyen los diagramas de flujo de las funciones principales del sistema.

a. Diagrama de Flujo Sistema MIMO

En la figura 3.3 se muestra el diagrama de flujo del sistema MIMO implementado.

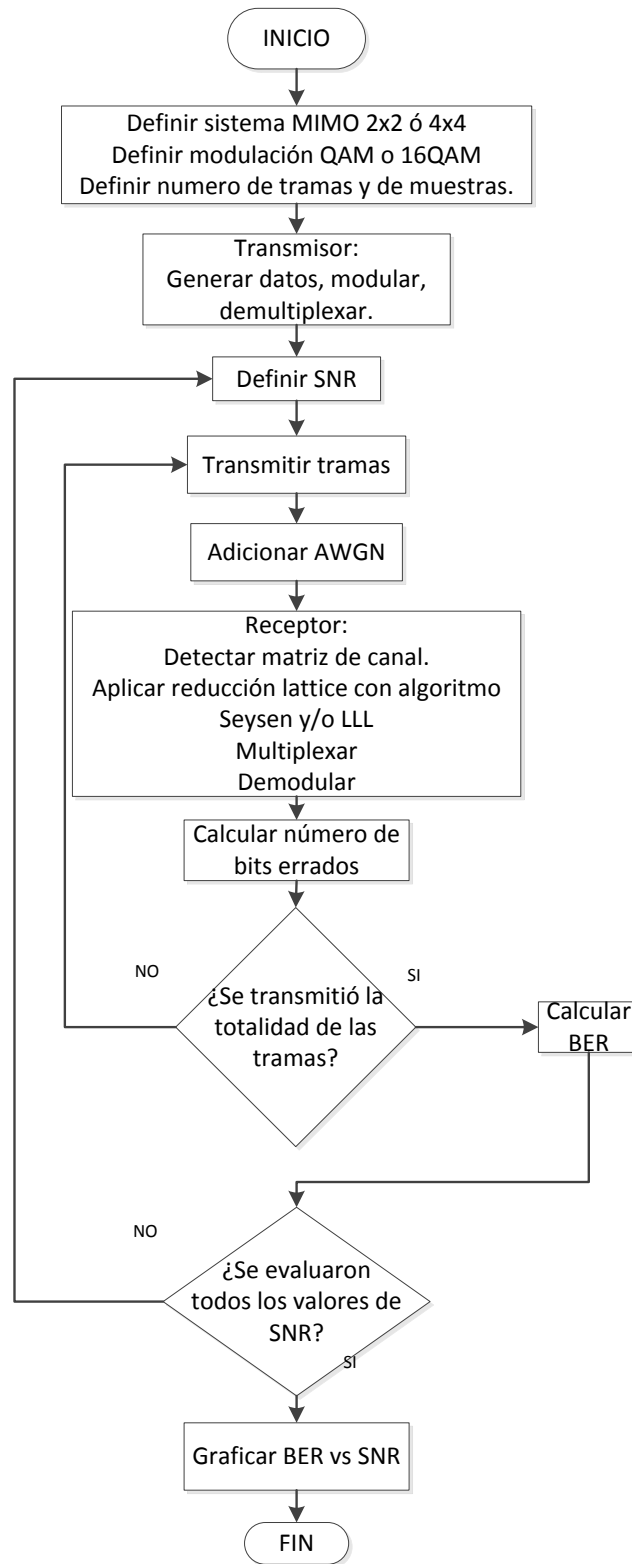


Figura 3.3. Diagrama de flujo del sistema MIMO



c. Diagrama de Flujo Algoritmo Seysen

En la figura 3.5 se muestra el diagrama de flujo del algoritmo *Seysen*

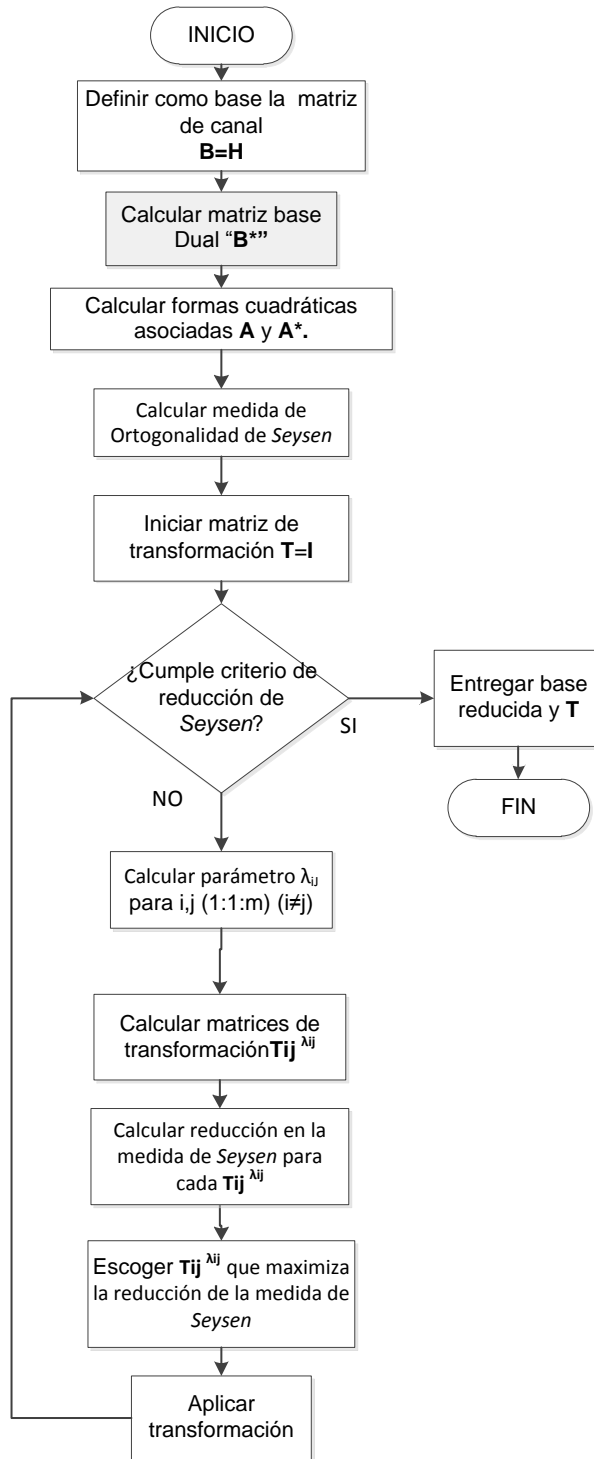


Figura 3.5. Diagrama de flujo algoritmo de *Seysen*



3.1.4 Fase 4: Selección del Lenguaje

Para el desarrollo de las simulaciones se estudian las herramientas Scilab® y Matlab®, reconocidas mundialmente para el desarrollo de este tipo de simulaciones. Estas se caracterizan por ser entornos confiables, versátiles y de alta capacidad en los que el código se escribe en un lenguaje propio.

Matlab® 7.14 es un entorno de programación de alto nivel que permite trabajar de una manera eficiente el desarrollo de algoritmos, visualización y análisis de datos, entornos de simulación de sistemas de comunicaciones, análisis numérico, procesamiento de imágenes, procesamiento de señales, control, entre otros. Esta herramienta permite un manejo amplio y rápido de matrices y vectores así como las diferentes operaciones entre estos, también permite la creación por código de algoritmos propios para manejo de procesos específicos de interés del usuario, además de facilitar la creación de interfaz gráfica. Matlab® es un programa con licencia comercial y en marzo de 2012 lanzó al mercado su versión R2012a para Windows®, Linux y Mac®, por lo que a pesar de ser un programa licenciado es multiplataforma y robusto [38].

Scilab® 5.4 es un entorno de programación de alto nivel al igual que Matlab®, pero tiene una interfaz gráfica más simple. No se requiere de licencia comercial debido a que este software es de uso libre, fue creado y desarrollado por INIRA (*Institut Nationale de Recherche en Informatique et en Automatique*) y en su página web está disponible su descarga para múltiples plataformas como Windows®, Linux y Mac® [39].

Dado que la universidad del Cauca tiene licencia para trabajar con Matlab®, y que esta herramienta brinda multifuncionalidad y facilidad para trabajar en múltiples plataformas se califica a Matlab® como la mejor herramienta para el desarrollo y posterior análisis de los algoritmos de *Seysen* y *LLL*, por lo cual se escoge el lenguaje M propio de esta herramienta.

3.1.5 Fase 5: Codificación del Modelo

En esta fase se genera el código computacional según el modelo diseñado. En las figuras 3.6 a 3.16 se muestra la codificación para cada uno de los bloques del sistema MIMO presentados en la figura 3.2.

3.1.5.1 Generador de datos

Variables de entrada:

modulation: orden de la modulación M-QAM.



array: número de antenas del sistema.

frames_number: número de tramas simuladas.

Variable de salida:

BinaryData: vector columna con la totalidad de los datos binarios a transmitir.

```
%% Generación de los datos binarios

% Se calcula la longitud de la trama según el orden de modulación y el
% número de antenas. 2x2_QAM=4, 2x2_16QAM=8, 4x4_QAM=8, 4x4_16QAM=16.
% Esta longitud garantiza ue cada trama tenga los suficientes bits
para
% proporcionar un símbolo a cada antena.
frame_lenght=log2(modulation)*array;

% Genera vector columna con todos los bits necesarios para armar el
% númerode tramas solicitado de la longitud calculada.
% Genera datos enteros uniformemente distribuidos entre cero y uno.
BinaryData=randi([0 1],frames_number*frame_lenght,1);
```

Figura 3.6. Código en Matlab® del bloque generador de datos

3.1.5.2 Modulador

Variables de entrada:

BinaryData: datos binarios a modular.

modulator: objeto modulador MODEM.QAMMOD de Matlab®.

Variable de salida:

ModulatedData: vector columna con los símbolos M-QAM correspondientes a los datos binarios de entrada.



```
%% Modulación de los datos binarios

% Se utiliza el objeto modulator del tipo MODEM.QAMMOD con la
siguiente
% configuración:
% Objeto Modulator del sistema
% Tipo: QAM
% Orden: modulation
% SymbolOrder: Gray (Típico QAM, facilitadeteccion)
% InputType: bit
ModulatedData=modulate(modulator, BinaryData);
```

Figura 3.7. Código en Matlab® del bloque modulador

3.1.5.3 Demultiplexor espacial

Variables de entrada:

ModulatedData: datos modulados obtenidos del bloque Modulación.

array: número de antenas del sistema.

frames_number: número de tramas simuladas.

Variables de salida:

DataTx: matriz con el número de filas igual al número de antenas del sistema y el número de columnas igual al número de tramas a simular.

```
%% Demultiplexación V-Blast

% Genera una matriz implementando la demultiplexación V-Blast al
% representar en cada columna los símbolos enviados en un instante por
% las antenas del arreglo.
DataTx=reshape(ModulatedData,array,frames_number);
```

Figura 3.8. Código en Matlab® del bloque demultiplexor

3.1.5.4 Canal AWGN

Variables de entrada

array: número de antenas del sistema.

FrameTx: columna de la matriz *DataTx* correspondiente a la trama transmitida.



snr_db : valor en decibeles de la relación señal a ruido.

```
%% Transmisión de la trama a través del canal  
  
% Generación de la matriz de transferencia del canal. Esta matriz  
% se compone de elementos complejos generados con una distribución  
% normal Gaussiana. Como el canal presenta desvanecimiento plano,  
% esta matriz del canal se genera para cada trama enviada.  
Hc=random('Normal',0,1,[array,array])+1i*random('Normal',0,1,[array,array]);  
  
% Se adiciona ruido blanco Gaussiano a los símbolos transmitidos  
% afectados por el canal. Al utilizar el parámetro 'measured',  
% primero se mide la potencia de los símbolos y luego se adiciona  
% el ruido según la SNR en db. Se genera el vector de ruido n con  
% componentes de ruido diferente para cada trama recibida, es decir  
% cada antena receptora trabaja con un componente de ruido de  
% diferente valor. Dando como resultado el vector de señales  
% recibidas.  
Yc=awgn(Hc*FrameTx_c,snr_db,'measured');
```

Figura 3.9. Código en Matlab® del bloque canal

3.1.5.5 Detector

a. Detector ML

Variables de entrada:

H : matriz de transferencia del canal.

Y : información recibida.

Modulation: orden de la modulación utilizada.

X : constelación de la modulación normalizada.

Variable de salida:

$FrameRX$: datos detectados por el algoritmo.



```
function FrameRX = ML(H,Y,modulation,X)
%% Detector ML
% Entradas
% H: Matriz de transferencia del canal
% Y: Información recibida
% modulation: Orden de modulación
% X: Constelación normalizada de la modulación
% Salidas
% FrameRX: Datos detectados

% Se determina la distancia euclidiana más corta entre el símbolo
% recibido los símbolos de la constelación por medio de una métrica.

metric = 10000;
for n = 1:modulation
x_tmp(1) = X(n);
Esti_Y1(:,1) = Y - H(:,1).*x_tmp(1);
for o = 1:modulation
x_tmp(2) = X(o);
Esti_Y2(:,1) = Esti_Y1(:,1) - H(:,2).*x_tmp(2);
metric_tmp=abs(Esti_Y2(1))^2+abs(Esti_Y2(2))^2;
if metric_tmp<metric
FrameRX = x_tmp;
metric = metric_tmp;
end
end
end
```

Figura 3.10. Código en Matlab® del detector ML

b. Detector ZF

Variables de entrada:

H: matriz de transferencia del canal.

Y: información recibida.

Variables de salida:

S_est: datos detectados por el algoritmo.



```
function [s_e]=ZF(H,Y)
%% Función que implementa el detector Zero Forcing
% Entradas
%     H: Matriz del canal
%     Y: Señal recibida
% Salidas
%     S_est: Estimación de los datos enviados

% Se multiplica el vector de señales recibidas por la matriz pseudo-
% inversa Moore-Penrose para eliminar el efecto del canal

s_e=round(pinv(H)*Y);

end
```

Figura 3.11. Código en Matlab® del detector ZF

c. Algoritmo LLL

VARIABLES DE ENTRADA:

H: matriz de transferencia del canal.

VARIABLES DE SALIDA:

B: base reducida.

T: matriz de transformación de la reducción.



```
function [B,T] = LLL(H)
% Función que realiza la reducción lattice utilizando el algoritmo LLL
% Entradas:
%     A: Matriz base a reducir (Matriz del canal)
% Salidas:
%     B: Base reducida
%     T: Matriz de transformación de la reducción

% La matriz recibida se reduce por medio de operaciones de renglones
% hasta que cumpla las condiciones de SizeReduction y la de Lovasz

l = 2;
while l <= m

% Reducción de tamaño del vector columna B(:,1)
for k=l-1:-1:1
mu = round(R(k,l)/R(k,k));      % abs(R(k,l))>0.5*abs(R(k,k))
if abs(mu)
B(:,l) = B(:,l) - mu * B(:,k);
R(1:k,l) = R(1:k,l) - mu * R(1:k,k);
T(:,l) = T(:,l) - mu * T(:,k);
end
end

% Validación de la condición de Lovasz
len = norm(R(l-1:l,l));
if delta*abs(R(l-1,l-1))^2 > len^2

% Intercambio de columnas l-1 y l en B, T y R.
B(:, [l-1 l]) = B(:, [l l-1]);
T(:, [l-1 l]) = T(:, [l l-1]);
R(1:l, [l-1 l]) = R(1:l, [l l-1]);
c = R(l-1,l-1) / len;      % len = ||R(l-1:l,l-1)||
s = R(l,l-1) / len;
Theta = [c' s; -s c];
R(l-1:l,l-1:end) = Theta * R(l-1:l,l-1:end);
Q(:,l-1:l) = Q(:,l-1:l) * Theta' ;
l = max(l-1,2);
else
l = l+1;
end
end
```

Figura 3.12. Código en Matlab® del algoritmo LLL

d. Algoritmo Seysen

VARIABLES DE ENTRADA:

H: matriz de transferencia del canal.



VARIABLES DE SALIDA:

T: matriz de transformación del algoritmo.

H_red: base reducida.

```
function [T,H_red] = SEYSEN(H)
% Función que realiza la reducción lattice utilizando el algoritmo de
% reducción de Seysen
% Entradas:
%   H: Matriz base a reducir (Matriz del canal)
% Salidas:
%   H_red: Base reducida
%   T: Matriz de transformación de la reducción

% Cálculo de los posibles valores de Lambda y de su correspondiente
% reducción en la medida de Seysen

for s = 1:m
for t = 1:m
if s ~= t
    x = 0.5*(A(s,t)/A(s,s)-A(t,s)/A(t,t));
    Lambda(s,t) = round(x);
    AbsLambda = abs(Lambda(s,t))^2;
    if AbsLambda ~= 0
        zw = real(Lambda(s,t))*real(x)+imag(Lambda(s,t))*imag(x);
        Delta(s,t) = A(s,s)*A(t,t)*(2*zw-AbsLambda);
    end
end
end
end

% Encontrar reducción de la medida de Seysen
[zw, max_ind] = max(Delta(:));
[s, t] = ind2sub([m,m],max_ind);

% Aplicar transformación
H_red(:,s) = H_red(:,s) + Lambda(s,t)*H_red(:,t);
T(:,s) = T(:,s) + Lambda(s,t)*T(:,t);
H_red_dual(:,t) = H_red_dual(:,t) - Lambda(s,t)'*H_red_dual(:,s);

end
```

Figura 3.13. Código en Matlab® del algoritmo de Seysen



3.1.5.6 Demodulador

Variables de entrada

demodulator: objeto demodulador MODEM.QAMMOD de Matlab®.

FrameRx_c_Z, *L*, *S*, *M*: tramas complejas entregadas por los diferentes detectores implementados.

Variables de salida

BinaryDataRx_Z, *L*, *S*, *M*: tramas binarias.

```
%% Demodulación de los datos complejos recibidos  
  
% Se utiliza el objeto demodulator del tipo MODEM.QAMMOD con la  
% siguiente  
% configuración:  
% Objeto Demodulador del sistema  
% Tipo: QAM  
% Orden: modulation  
% SymbolOrder: Gray  
% OutputType: bit  
% DecisionType: Hard Decision  
BinaryDataRx_Z=demodulate(demodulator,FrameRx_c_Z);  
BinaryDataRx_L=demodulate(demodulator,FrameRx_c_L);  
BinaryDataRx_S=demodulate(demodulator,FrameRx_c_S);  
BinaryDataRx_M=demodulate(demodulator,FrameRx_c_M);
```

Figura 3.14. Código en Matlab® del bloque demodulador

3.1.5.7 Medidor de BER

El medidor de BER desarrolla dos operaciones: calcular el número de errores por trama para cada valor de SNR y calcular la BER.

a. Cálculo de errores por trama

Variables de entrada

BinaryDataRx_Z, *L*, *S*, *M*: tramas binarias recibidas por cada detector.

FramedBinaryDataTx: tramas binarias enviadas.

Variables de salida

ErrorsBySNR_Z, *L*, *S*, *M*: total de errores encontrados para cada valor de SNR.



```
%% Receptor Final
% Cálculo de errores
% Se calculan los errores del número total de tramas para cada
% valor de SNR. Compara los datos detectados con los originales
% conservando sólo los que son diferentes para contarlos y
% determinar cuantos errores hubo en la trama.

[ErrorsByFrame_Z ~]=size(BinaryDataRx_Z(BinaryDataRx_Z~=FramedBinaryDataTx(:,frame)));
ErrorsBySNR_Z(snr_index)=ErrorsBySNR_Z(snr_index)+ErrorsByFrame_Z;

[ErrorsByFrame_L ~]=size(BinaryDataRx_L(BinaryDataRx_L~=FramedBinaryDataTx(:,frame)));
ErrorsBySNR_L(snr_index)=ErrorsBySNR_L(snr_index)+ErrorsByFrame_L;

[ErrorsByFrame_S ~]=size(BinaryDataRx_S(BinaryDataRx_S~=FramedBinaryDataTx(:,frame)));
ErrorsBySNR_S(snr_index)=ErrorsBySNR_S(snr_index)+ErrorsByFrame_S;

ErrorsByFrame_M ~]=size(BinaryDataRx_M(BinaryDataRx_M~=FramedBinaryDataTx(:,frame)));
ErrorsBySNR_M(snr_index)=ErrorsBySNR_M(snr_index)+ErrorsByFrame_M;
```

Figura 3.15. Código en Matlab® del bloque medidor de BER – Errores

b. Cálculo de BER

Variables de entrada

ErrorsBySNR_Z, L, S, M: total de errores encontrados para cada valor de SNR.

Variables de salida

BER_Z, L, S, M: tasa de error de bits para cada método de detección.

```
%% Cálculo de la BER
% Se calcula la BER para cada valor de SNR dividiendo el número de bits
% errados entre el número total de bits enviados.

BER_Z(snr_index)=ErrorsBySNR_Z(snr_index)/(log2(modulation)*array*f_number);
BER_L(snr_index)=ErrorsBySNR_L(snr_index)/(log2(modulation)*array*f_number);
BER_S(snr_index)=ErrorsBySNR_S(snr_index)/(log2(modulation)*array*f_number);
BER_M(snr_index)=ErrorsBySNR_M(snr_index)/(log2(modulation)*array*f_number);
```

Figura 3.16. Código en Matlab® del bloque medidor de BER - BER



3.1.6 Fase 6: Validación

En la fase de validación se realizan comparaciones entre los resultados obtenidos por la simulación y resultados de otros trabajos similares publicados, también se analiza la coherencia de los resultados experimentales con la teoría de *lattices* estudiada. En el capítulo cuatro se hace una presentación más detallada de esta fase.

3.1.7 Fase 7: Experimentación

En esta fase se definen los escenarios para la simulación de los algoritmos LRA según las variables de entrada del sistema. En la tabla 3-1 se muestran las características de los distintos escenarios de simulación.

Tabla 3-1. Escenarios de simulación

Escenario	Número de Antenas	Orden de Modulación QAM
Escenario 1	2x2	4
Escenario 2	2x2	16
Escenario 3	4x4	4
Escenario 4	4x4	16

3.1.8 Fase 8: Implantación

Una vez completadas las anteriores fases se determina si la simulación desarrollada a lo largo del proceso es adecuada para obtener los resultados necesarios y cumplir a cabalidad los objetivos de este trabajo de grado.

4 VALIDACIÓN DE RESULTADOS

En este capítulo se presenta la validación de los resultados por medio de la comparación con resultados de otros trabajos de grado y la verificación de los principios de la teoría de *lattices*.

En la comparación con resultados de otros trabajos de grado se aplicaron los criterios de equidistancia y ajuste. La equidistancia determina qué tan homogénea es la distancia entre las curvas de desempeño obtenidas en este trabajo de grado y las curvas obtenidas por otros autores. El ajuste es la diferencia de valores de SNR para un mismo valor de BER de dos curvas diferentes [40].

Al aplicar la reducción *lattice* a una matriz se busca convertirla en una matriz ortogonal por lo que la verificación de los principios de la teoría de *lattices* utiliza herramientas matemáticas como el índice de ortogonalidad. Este parámetro varía entre cero y uno, donde uno representa una matriz ortogonal y cero una matriz singular que dificulta el proceso de diferenciación de símbolos, por lo tanto una matriz reducida satisfactoriamente debe tener un índice de ortogonalidad igual o cercano a la unidad [17].

4.1 COMPARACIÓN CON OTROS TRABAJOS DE GRADO

La validación por medio de la comparación con otros trabajos de grado utiliza medidas estadísticas de tendencia central y de dispersión, para determinar por medio del coeficiente de variación si las curvas de desempeño, resultado de la simulación son equidistantes con las encontradas en trabajos similares, tomados como referencia [40].

Se dice que dos curvas de desempeño son equidistantes si el coeficiente de variación de las diferencias es menor al 50%, es decir, si su comportamiento es similar [40].

4.1.1 Algoritmo *Lenstra Lenstra Lovász*

En la tabla 4-1 se comparan los valores de SNR obtenidos al simular el algoritmo LLL en los escenarios definidos en la tabla 3-1 con valores de trabajos referencia. Para el algoritmo LLL no se encontraron trabajos referencia que coincidieran con los escenarios 1, 2 y 3, por lo que los valores mostrados en la tabla correspondientes a estos escenarios fueron calculados con base a valores



referencia correspondientes al algoritmo ML de los trabajos [41] [7] [42] [43]. Para el escenario 4 se calcularon las diferencias teniendo en cuenta los resultados del trabajo [8].

Tabla 4-1. Diferencia de SNR para algoritmo de referencia y LLL simulado

BER	Escenario 1 [dB]	Escenario 2 [dB]	Escenario 3 [dB]	Escenario 4 [dB]
10^{-1}	2.76	8.17	8.48	9.83
10^{-2}	4.47	10.96	10.46	12.53
10^{-3}	5.16	12.01	11.86	13.15
10^{-4}	6.21	13.13	12.92	14.06
Media	4.65	11.06	10.93	12.39
Desv. Est.	1.448	2.125	1.919	1.82
C.V. (%)	31.1	19.2	17.5	14.6

Como se puede observar en la tabla 4-1 el algoritmo LLL presenta valores del coeficiente de variación menores al 50% para todos los escenarios, siendo el mayor valor 31.1%, por lo cual se determina que los resultados obtenidos son válidos y presentan un comportamiento similar a los resultados de los trabajos tomados como referencia[40].

4.1.2 Algoritmo *Seysen*

La validación de resultados del algoritmo *Seysen* se realizó comparándolos con resultados de trabajos relacionados con el algoritmo ML, pues no se encontró ningún trabajo que simulara este algoritmo en escenarios similares a los definidos en la tabla 3-1.

En la tabla 4-2 se muestran los coeficientes de variación obtenidos al realizar la comparación de valores de SNR obtenidos en la simulación con los de trabajos afines [8] [41] [42] [43].

Tabla 4-2. Diferencia de SNR para algoritmo de referencia y *Seysen* simulado

BER	Escenario 1 [dB]	Escenario 2 [dB]	Escenario 3 [dB]	Escenario 4 [dB]
10^{-1}	2.75	8.18	8.48	11.83
10^{-2}	4.46	10.95	10.4	14.65
10^{-3}	5.15	12.03	11.67	15.65
10^{-4}	6.18	13.13	12.33	16.47
Media	4.63	11.07	10.72	14.65
Desv. Est.	1.441	2.123	1.694	2.021
C.V. (%)	31.1	19.1	15.8	13.8



Los resultados del algoritmo de *Seysen* en los cuatro escenarios definidos son válidos ya que presentan valores del coeficiente de variación menores al 50% [40].

4.2 VERIFICACIÓN DE LOS PRINCIPIOS DE LA TEORÍA DE *LATTICES*

Al aplicar técnicas de reducción *lattice* en la detección de señales en sistemas MIMO, en el receptor del sistema se trabaja con una superposición lineal de puntos organizados en una matriz que debería ser ortogonal, para determinar si una matriz es singular u ortogonal se puede utilizar el índice de ortogonalidad [17].

4.2.1 Índice de Ortogonalidad

Según la definición de índice de ortogonalidad presentada en el capítulo dos, una matriz es ortogonal si tiene un índice de ortogonalidad igual a la unidad, si este índice es mayor a la unidad se dice que la matriz tiende a ser singular. Por lo tanto se espera que al aplicar a una matriz una técnica de reducción *lattice* se obtenga una nueva matriz cuyo índice de ortogonalidad tenga valores cercanos o iguales a la unidad.

En la tabla 4-3 se muestran los valores promedio del índice de ortogonalidad para las matrices del canal sin aplicar ninguna técnica de reducción *lattice*.

Tabla 4-3. Media y varianza del índice de ortogonalidad sin reducción *lattice*

Número de Antenas	Media	Varianza
2	1.9910	15.9114
4	6.1825	176.4245

4.2.1.1 Algoritmo LLL

Los valores de media y varianza del índice de ortogonalidad de 100000 matrices complejas, luego de aplicar el algoritmo de reducción *lattice* LLL se muestran en la tabla 4-4.

Tabla 4-4. Media y varianza del índice de ortogonalidad utilizando LLL

Número de Antenas	Media	Varianza
2	1.0681	0.0042
4	1.6006	0.0795



Los resultados obtenidos muestran que después de aplicar la técnica de reducción *lattice* a las matrices simuladas, el valor medio del índice de ortogonalidad tiene valores cercanos a la unidad lo cual indica que el proceso de reducción se realizó correctamente.

4.2.1.2 Algoritmo de *Seysen*

Los valores de media y varianza del índice de ortogonalidad de 100000 matrices complejas, luego de aplicar el algoritmo de reducción *lattice Seysen* se muestran en la tabla 4-5.

Tabla 4-5. Media y varianza del índice de ortogonalidad utilizando *Seysen*

Número de Antenas	Media	Varianza
2	1.0659	0.0034
4	1.5676	0.0545

Los resultados obtenidos muestran el valor medio del índice de ortogonalidad es cercano a la unidad, lo cual indica que el proceso de reducción *lattice* se realizó correctamente, los valores de varianza cercanos a cero muestran que los datos obtenidos en las simulaciones presentan una mínima dispersión.

5 ANÁLISIS DE RESULTADOS

En la primera sección de este capítulo se presentan los resultados obtenidos al simular los algoritmos de detección implementados. Para cada uno de los escenarios se muestran las curvas de desempeño correspondientes a un rango de valores de SNR entre 0 y 60 dB. Los resultados presentados son los valores promedio calculados a partir de los datos obtenidos al simular 10 veces 10000 tramas para cada valor de SNR, esto con el fin de obtener una colección de datos con una varianza mínima.

En la segunda sección se presentan los tiempos de reducción generados por los algoritmos *Lattice Reduction-Aided*, los cuales son un indicador de la complejidad computacional.

5.1 ANÁLISIS DE DESEMPEÑO

5.1.1 Escenario 1

Consiste en un sistema MIMO 2x2 con modulación 4QAM. Los resultados obtenidos al simular los algoritmos ML, ZF, LLL y *Seysen* se muestran en la figura 5.1.

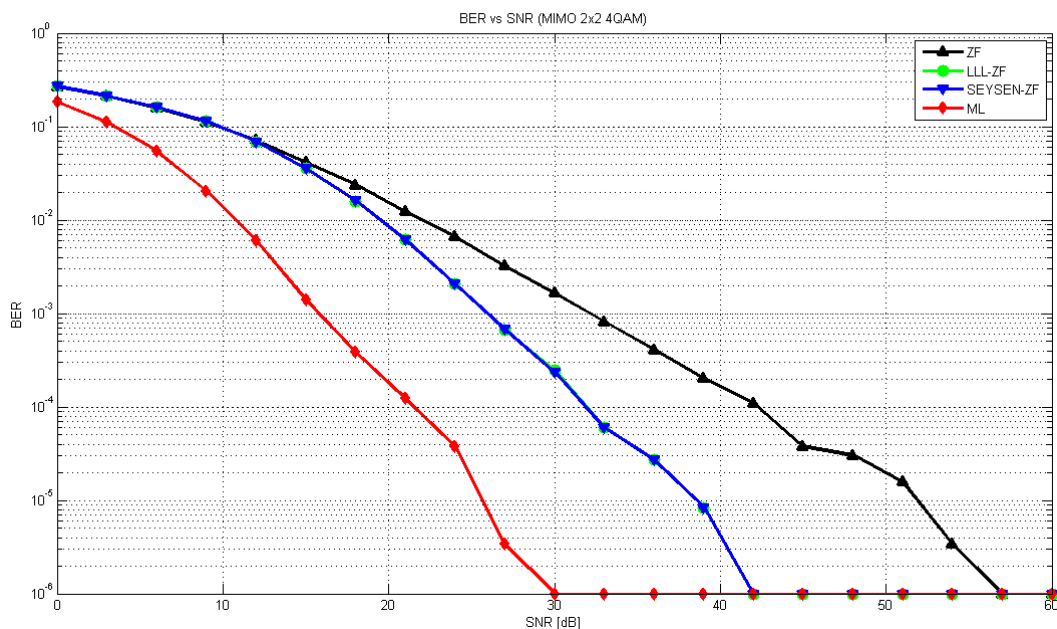


Figura 5.1. Curvas de desempeño para el escenario 1



En la figura 5.1 se puede apreciar que para valores de SNR inferiores a 12 dB no hay diferencia entre los algoritmos LRA (LLL y *Seysen*) y el algoritmo ZF. El comportamiento de los algoritmos LRA en este escenario es idéntico (las curvas están superpuestas), con curvas casi paralelas a la del ML a partir de los 15dB. El algoritmo que presenta mejor desempeño en este escenario es el ML, la curva de desempeño del algoritmo ZF se va alejando progresivamente a medida que aumentan los valores de SNR demostrando el peor desempeño, mientras que los algoritmos LRA presentan un desempeño mejor que el ZF y peor que el ML.

En la tabla 5-1 se muestran los valores de SNR asociados a determinados valores de BER para cada algoritmo simulado.

Tabla 5-1. Tabla de desempeño para el escenario 1

BER	SNR[dB]			
	ML	ZF	LLL	<i>Seysen</i>
10^{-1}	3.46	9.72	9.76	9.75
10^{-2}	10.76	22	19.47	19.46
10^{-3}	15.78	32.12	25.96	25.95
10^{-4}	21.51	42.27	31.91	31.88
10^{-5}	25.65	51.88	38.55	38.55
10^{-6}	30	57	42	42

De la tabla 5-1 Se puede observar que el algoritmo ML tiene los menores requerimientos de SNR para todos los valores de BER analizados. Los algoritmos LRA presentan mayores requerimientos de SNR que el algoritmo ML, para valores de BER de 10^{-1} la diferencia entre estos y el algoritmo ML es de aproximadamente 6 dB y a medida que se reducen los valores de BER esta diferencia va aumentando hasta llegar a un máximo de 12.9 dB para una BER de 10^{-5} . El algoritmo ZF tiene los mayores requerimientos de SNR para valores de BER menores a 10^{-1} , a partir de este valor los algoritmos LRA producen mejoras de hasta 15 dB con respecto al algoritmo ZF para valores de BER de 10^{-6} .

El comportamiento de los algoritmos LLL y *Seysen* es similar. Al comparar los algoritmos simulados, se puede concluir que los algoritmos LRA requieren una mayor SNR que el ML para proporcionar los mismos valores de BER, mientras que los requerimientos adicionales del ZF se van incrementando a medida que se reduce la BER, lo cual muestra que al aplicar técnicas de reducción *lattice* en sistemas MIMO se reducen los requerimientos de SNR en comparación con el algoritmo ZF.

5.1.2 Escenario 2

Es un sistema MIMO 2x2 con modulación 16QAM. Al simular los diferentes algoritmos se obtuvieron las curvas de desempeño mostradas en la figura 5.2.

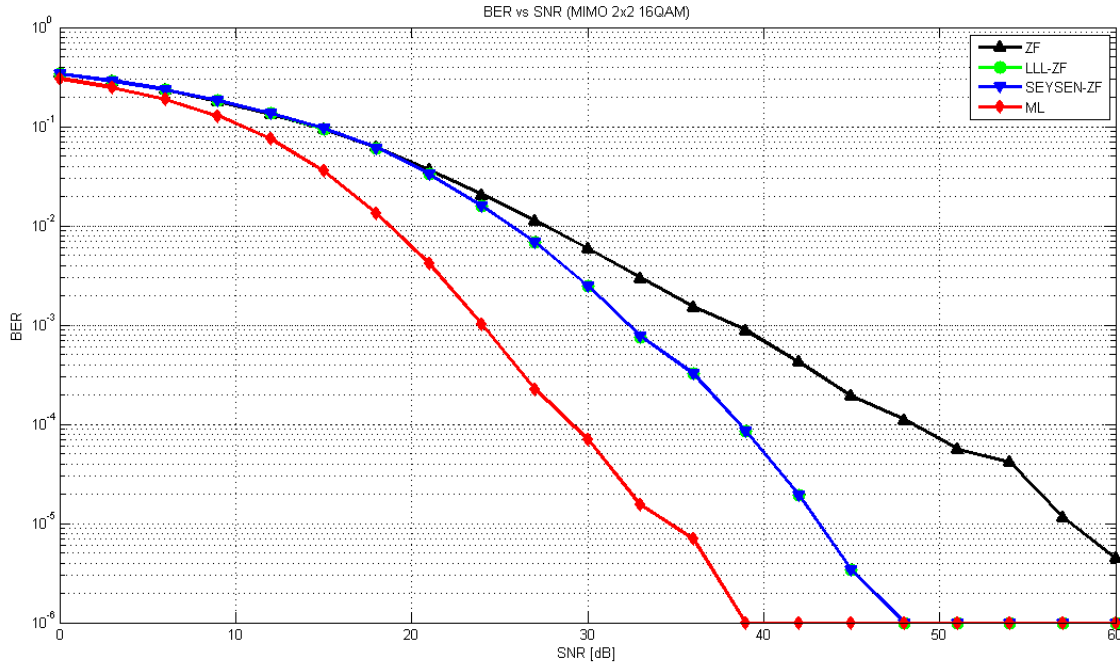


Figura 5.2. Curvas de desempeño para el escenario 2

Las curvas de la figura 5.2 muestran que el algoritmo ML presenta el mejor desempeño, seguido por los algoritmos LRA, que a partir de 17 dB de SNR presentan un desempeño mejor que el del algoritmo ZF. A medida que aumentan los valores de SNR la mejora en el desempeño de los algoritmos LRA sobre el ZF se incrementa, mientras que la diferencia entre el desempeño de los algoritmos LRA y el algoritmo ML se mantiene prácticamente constante a partir de los 27 dB.

Tabla 5-2. Tabla de desempeño para el escenario 2

BER	SNR[dB]			
	ML	ZF	LLL	Seysen
10^{-1}	10.36	14.4	14.57	14.58
10^{-2}	18.71	27.55	25.66	25.65
10^{-3}	24.01	38.29	32.31	32.33
10^{-4}	29.09	48.43	38.63	38.63
10^{-5}	34.63	57.45	43.14	43.14
10^{-6}	39	-	48	48

De la tabla 5-2 Se puede concluir que los menores requerimientos de SNR los presenta el algoritmo ML para todos los valores de BER analizados. Los algoritmos LRA tienen mayores



requerimientos de SNR que el algoritmo ML, pero menores que el ZF. Para valores de BER de 10^{-1} se da un caso excepcional en el que el algoritmo ZF requiere 0.18 dB menos de SNR que los algoritmos LRA, para valores de BER iguales a 10^{-2} la diferencia entre los algoritmos LRA y el algoritmo ML es de aproximadamente 4.2 dB, mientras que la diferencia entre los algoritmos ZF y ML es de 8.84 dB. A medida que se reducen los valores de BER la diferencia entre los algoritmos LRA y ML alcanza un valor máximo de 9.5 dB para una BER de 10^{-4} .

El algoritmo ZF tiene los mayores requerimientos de SNR con diferencias de hasta 14.3 dB sobre los algoritmos LRA para valores de BER de 10^{-5} . Estas diferencias muestran claramente que los algoritmos LRA presentan un desempeño más cercano al algoritmo ML que el algoritmo ZF.

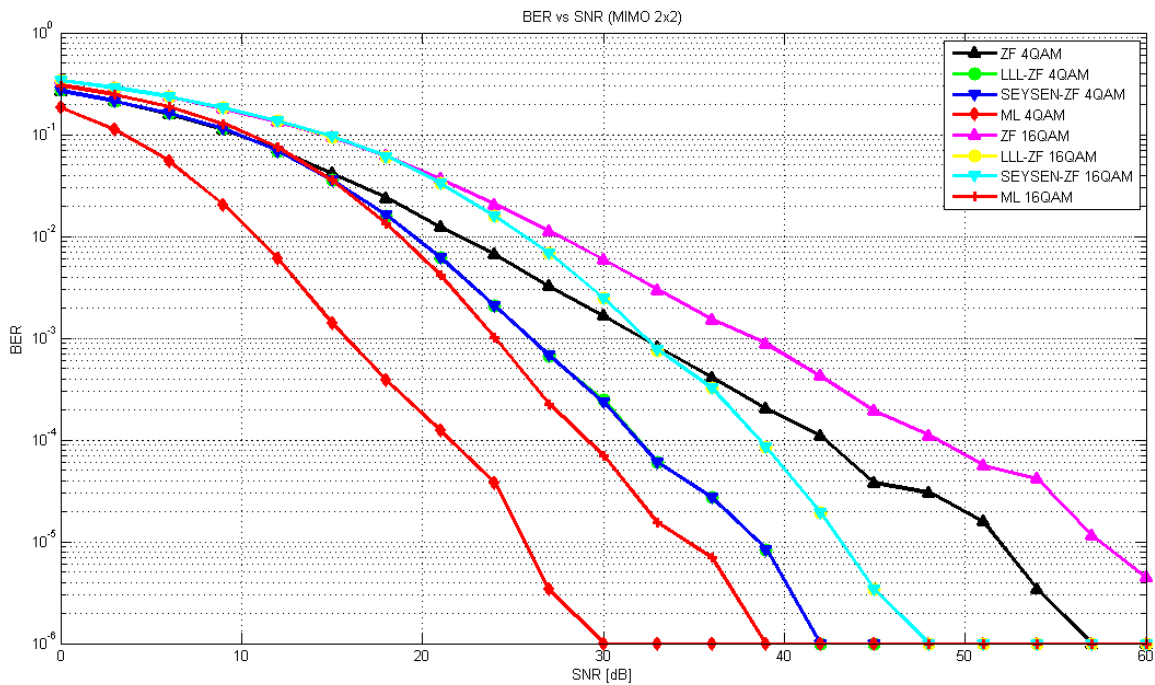


Figura 5.3. Curvas de desempeño de los algoritmos simulados para sistema MIMO 2x2

Las curvas de desempeño de la figura 5.3 muestran que al comparar los resultados obtenidos en los escenarios 1 y 2 se puede observar que al cambiar el orden de la modulación de 4QAM a 16QAM en un sistema MIMO 2x2 los requerimientos de SNR de los algoritmos LLL, *Seysen* y ZF aumentan en promedio 5 dB para los valores de BER estudiados y los del algoritmo ML en 8 dB, lo cual significa que el desempeño de los algoritmos de detección disminuye al aumentar el orden de la modulación.



5.1.3 Escenario 3

Es un sistema MIMO 4x4, modulación 4QAM y con algoritmos LRA, ML y ZF. Las curvas de desempeño correspondientes a los algoritmos simulados se muestran en la figura 5.4.

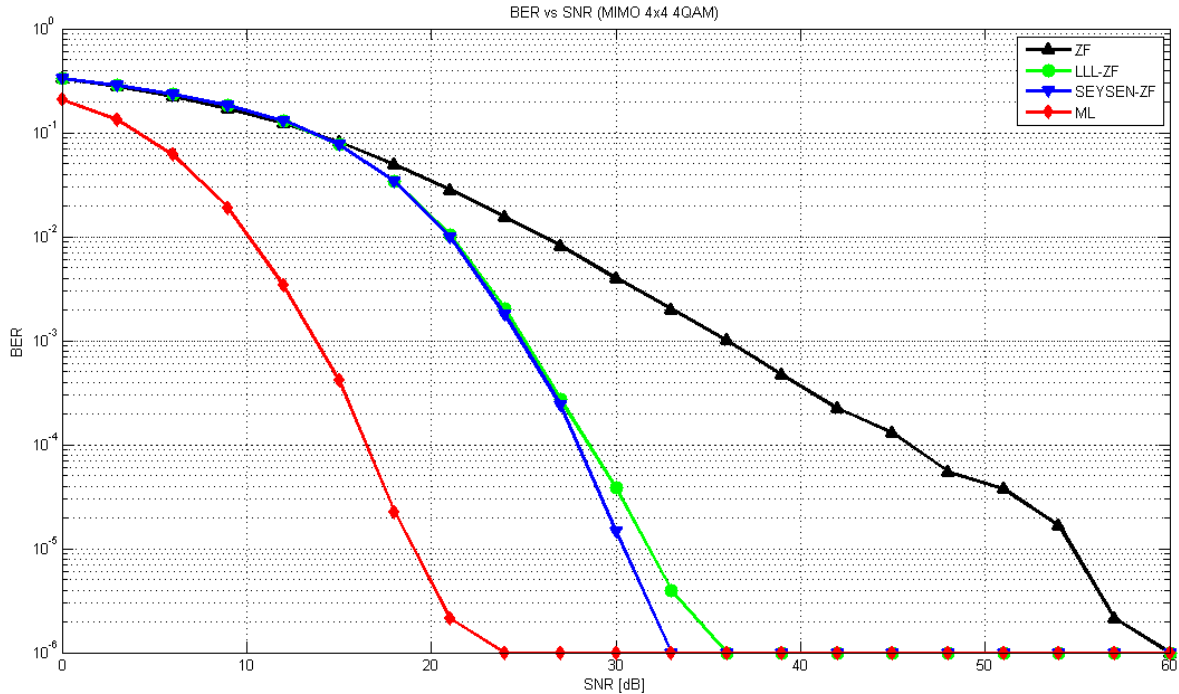


Figura 5.4. Curvas de desempeño para el escenario 3

Las curvas de desempeño de la figura 5.4 muestran que para valores de SNR mayores a 12 dB los algoritmos LRA tienen un mejor desempeño con respecto al algoritmo ZF, a partir de este valor su curva de desempeño es equidistante a la del algoritmo ML, mientras que la curva del ZF se aleja cada vez más, indicando que este algoritmo es el de peor desempeño. Aunque el algoritmo ML sigue siendo el de mejor desempeño, los algoritmos LRA presentan un mejor comportamiento que en los escenarios anteriores, lo cual se evidencia por la pronunciada pendiente de las curvas. También se puede observar que a partir de los 21 dB de SNR, el algoritmo *Seysen* muestra un desempeño mejor que el LLL con una diferencia que aumenta a medida que se incrementa la SNR, caso que no se había visto en los escenarios con sistemas MIMO 2x2.

La tabla 5-3 muestra valores detallados de los requerimientos de SNR para una BER específica según el algoritmo implementado.



Tabla 5-3. Tabla de desempeño para el escenario 3

BER	SNR[dB]			
	ML	ZF	LLL	<i>Seysen</i>
10^{-1}	4.08	13.49	13.48	13.48
10^{-2}	10.11	26.04	21.06	21
10^{-3}	13.74	36.03	25.06	24.87
10^{-4}	16.46	45.89	28.52	27.93
10^{-5}	19.04	54.75	31.78	30.42
10^{-6}	24	60	36	33

Según los valores de la tabla 5-3, el algoritmo ZF tiene los mayores requerimientos de SNR para todos los valores de BER, incrementándose a medida que disminuye la BER llegando al punto de requerir 60 dB para una BER de 10^{-6} . Las mejoras de los algoritmos LRA con relación al ZF son poco significativas para valores de BER de 10^{-1} , pero para valores menores a 10^{-2} estas mejoras se incrementan hasta llegar a un máximo de 27 dB de SNR para una BER de 10^{-6} . El algoritmo ML requiere en promedio 11 dB de SNR menos que los algoritmos LRA para los valores de BER analizados, haciendo que este sea el algoritmo con el mejor desempeño para el escenario planteado.

A partir de valores de BER menores a 10^{-4} se observa una diferencia significativa entre los requerimientos de SNR de los algoritmos LLL y *Seysen* siendo de 0.6 dB para una BER de 10^{-4} y llegando hasta 3 dB para 10^{-6} , por lo cual el algoritmo de *Seysen* tiene un mejor desempeño que el algoritmo LLL en este escenario.

Para el ZF los requerimientos adicionales de SNR con respecto al algoritmo ML ahora son más altos, alcanzando diferencias de 36dB, mientras que los requerimientos de SNR de los algoritmos LRA permanecen estables, por lo que se puede decir que las mejoras obtenidas a través de los algoritmos *lattice* son independientes de los requerimientos del algoritmo ZF.

5.1.4 Escenario 4

Corresponde a un sistema MIMO 4x4 con modulación 16QAM y con diferentes algoritmos de detección, en la figura 5.5 se muestran las curvas de desempeño obtenidas al simular los algoritmos ML, ZF, *Seysen* y LLL.

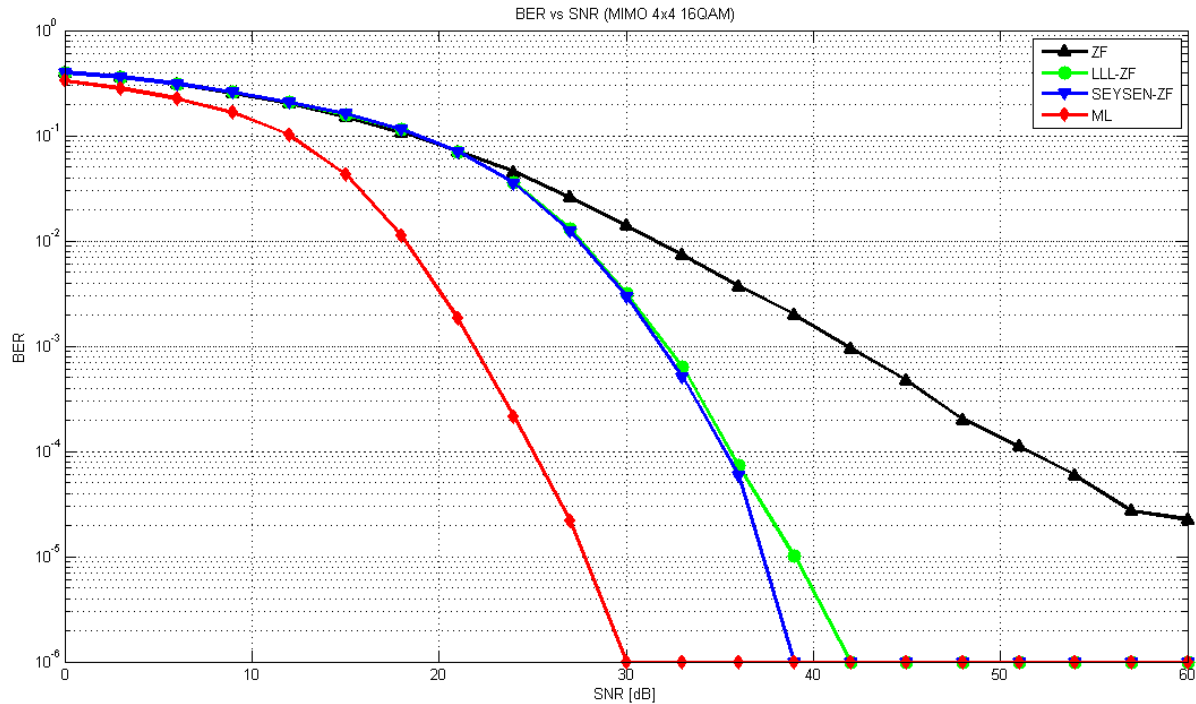


Figura 5.5. Curvas de desempeño para el escenario 4

En las curvas de la figura 5.5 se puede apreciar que para valores de SNR inferiores a 20 dB el algoritmo ZF presenta un desempeño levemente mejor que los algoritmos LRA, pero a medida que aumenta la SNR se encuentra un punto de cruce en 20 dB a partir del cual los algoritmos LRA presentan un mejor desempeño que el algoritmo ZF con una curva paralela a la del algoritmo ML.

Los algoritmos LRA LLL y *Seysen* presentan un desempeño igual hasta los 27 dB de SNR, valor a partir del cual el algoritmo *Seysen* presenta un mejor desempeño que el LLL por una diferencia que se hace más evidente a medida que se incrementa la SNR. El algoritmo ML presenta el mejor desempeño mientras que el algoritmo ZF el peor, demostrando que la utilización de los algoritmos LRA mejora el desempeño del sistema acercándolo al algoritmo ML.

En la tabla 5-4 se muestran valores de SNR requeridos para alcanzar ciertos niveles de BER según el algoritmo implementado.



Tabla 5-4. Tabla de desempeño para el escenario 4

BER	SNR[dB]			
	ML	ZF	LLL	<i>Seysen</i>
10^{-1}	12.05	18.45	18.83	18.83
10^{-2}	18.19	31.52	27.53	27.45
10^{-3}	21.84	41.78	32.15	31.85
10^{-4}	25	51.5	35.56	35.27
10^{-5}	27.76	-	39.01	37.31
10^{-6}	30	-	43	39

De acuerdo con la información suministrada en la tabla 5-4 los algoritmos LLL y *Seysen* presentan comportamientos distintos en cuanto a los requerimientos de SNR para determinados valores de BER. Aunque para una BER de 10^{-1} no hay diferencia entre el algoritmo *Seysen* y el LLL, a partir de una BER de 10^{-3} el algoritmo *Seysen* presenta un mejor desempeño con requerimientos de SNR 0.3 dB menores en comparación con el LLL. La diferencia máxima entre los requerimientos de SNR se da para una BER de 10^{-6} , valor para el cual el algoritmo de *Seysen* requiere 4 dB de SNR menos que el LLL.

El algoritmo ML presenta el mejor desempeño mostrando los menores requerimientos de SNR con una diferencia mínima de 6.78 dB para una BER de 10^{-1} en comparación con los algoritmos LRA, esta diferencia alcanza su máximo en comparación con el algoritmo *Seysen* para una BER de 10^{-4} , con una SNR menor en 10.27 dB y para el algoritmo LLL con una SNR menor en 13 dB para una BER de 10^{-6} .

Al comparar los valores obtenidos del algoritmo ZF con los de los algoritmos LRA es evidente que la aplicación de las técnicas de reducción *lattice* introduce una mejora importante. Los requerimientos de SNR de los algoritmos LRA son 4 dB menores para una BER de 10^{-2} y hasta 9 dB y 16 dB menores para valores de BER de 10^{-3} y 10^{-4} respectivamente.

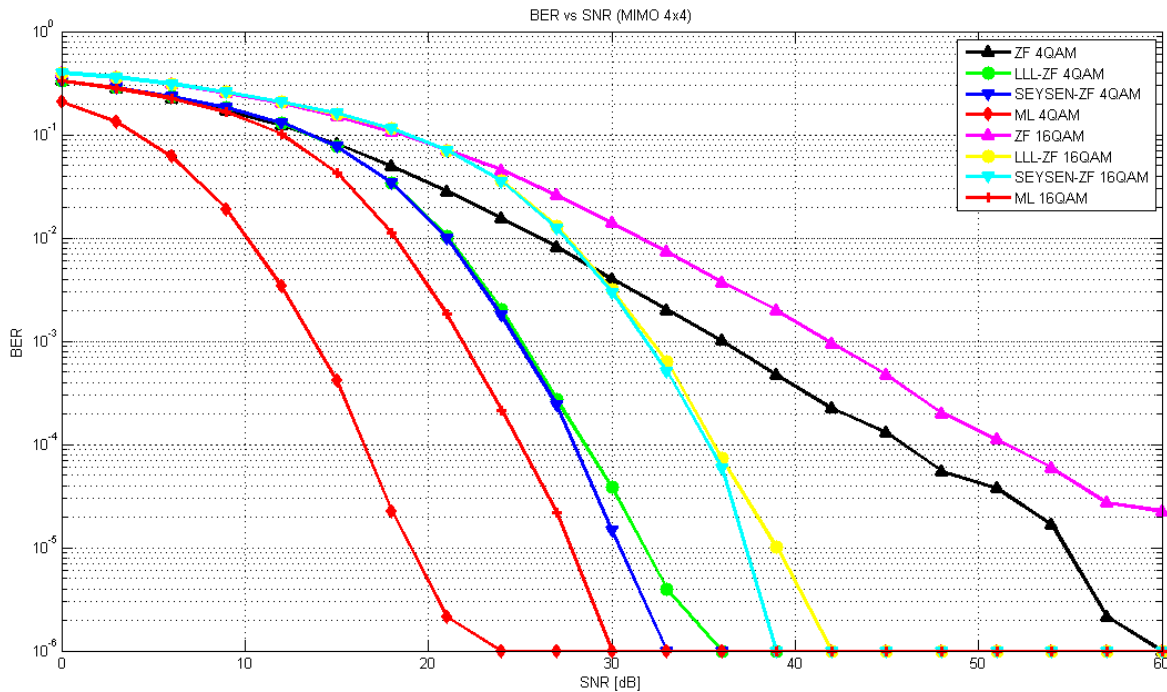


Figura 5.6. Curvas de desempeño de los algoritmos simulados para sistema MIMO 4x4

La figura 5.6 muestra que al comparar los resultados de los escenarios 3 y 4 se puede ver que el cambio en el orden de la modulación de 4QAM a 16QAM afecta principalmente al algoritmo ML ya que las diferencias en los requerimientos de SNR son aproximadamente 8 dB mayores en el escenario 4 que corresponde a una modulación de 16QAM. Los algoritmos LRA presentan mejor desempeño en el escenario 3 con modulación 4QAM, pues al aumentar el orden de la modulación a 16QAM se requieren en promedio 7 dB más que en el escenario 3 para alcanzar los valores de BER analizados, caso similar sucede con el algoritmo ZF para el cual la diferencia es de 6 dB.

Al analizar los resultados obtenidos en los cuatro escenarios simulados (ver anexo B) se puede determinar que el número de antenas del sistema afecta de forma diferente el desempeño de los algoritmos ML y LRA en comparación con el algoritmo ZF. En sistemas MIMO 2x2 el algoritmo ZF presenta un mejor desempeño que en sistemas MIMO 4x4, ya que sus requerimientos de SNR son en general 3 dB mayores. Al aumentar el número de antenas en un sistema MIMO los algoritmos ML y LRA presentan un mejor desempeño disminuyendo sus requerimientos de SNR a partir de una BER de 10^{-2} , valor para el cual el algoritmo ML requiere 0.6 dB menos en un sistema 4x4, para valores de 10^{-6} esta mejora es de hasta 9 dB. Los algoritmos LRA presentan un mejor desempeño a partir de valores de BER de 10^{-3} con requerimientos de SNR 0.4 dB menores en un sistema 4x4 hasta valores de SNR menores en 9 dB para el algoritmo *Seysen* y en 6 dB para el algoritmo *LLL* cuando la BER requerida es de 10^{-6} .



5.2 ANÁLISIS DE TIEMPOS DE REDUCCIÓN

Según los resultados de desempeño, los algoritmos LRA permiten alcanzar resultados más cercanos al detector ideal ML, pero no se puede establecer una comparación acertada sin tener en cuenta el costo de procesamiento de esta adición al sistema, por esto se analizan los tiempos de procesamiento promedio adicional que implica la aplicación de técnicas de reducción *lattice*, y el número de iteraciones requerido por cada algoritmo para realizar la reducción de la matriz base.

Para medir los tiempos de reducción no se tuvieron en cuenta los escenarios definidos en la tabla 3-1, ya que la modulación no influye en esta parte del procesamiento. El parámetro que se modificó fue el número de antenas, el cual determina la dimensión de la matriz del canal a reducir.

Los resultados presentados en las secciones siguientes corresponden a los tiempos e iteraciones promedio que se necesitaron al reducir 10000 matrices del canal 10 veces utilizando los algoritmos LLL y *Seysen*. Para estas pruebas se simuló por separado los procesos de reducción en el mismo equipo en el que se llevaron a cabo las pruebas de desempeño con el fin de tener un punto de referencia común.

5.2.1 Sistema MIMO 2X2

En la tabla 5-5 se muestran los valores correspondientes a tiempos de simulación y número de iteraciones promedio, empleados por cada algoritmo al reducir una matriz de canal en un sistema MIMO 2x2.

Tabla 5-5. Tiempos de reducción *lattice* e iteraciones en un sistema MIMO 2x2

Algoritmo	Iteraciones	Tiempo
LLL	1.6615	104.01 μ s
<i>Seysen</i>	1.0792	351.06 μ s

Según los datos presentados en la tabla 5-5, para sistemas MIMO 2x2 el algoritmo LLL realiza la reducción de la matriz en un tiempo mucho menor que el algoritmo *Seysen* aunque éste lo haga en menos iteraciones, esto se debe a que las iteraciones del algoritmo *Seysen* son más extensas y por lo tanto requieren de mayor tiempo de procesamiento.

5.2.2 Sistema MIMO 4X4

Los tiempos de simulación e iteraciones realizadas por los algoritmos *lattice* para reducir una matriz de canal en un sistema MIMO 4x4 se muestran en la tabla 5-6.



Tabla 5-6. Tiempos de reducción *lattice* e iteraciones en un sistema MIMO 4x4

Algoritmo	Iteraciones	Tiempo
LLL	9.16	341.72 μ s
<i>Seysen</i>	5.5	1.3 ms

La tabla 5-6 muestra que para sistemas MIMO 4x4 se mantiene la misma relación entre el número de iteraciones y los tiempos de reducción de los algoritmos *lattice*, siendo el más rápido el LLL a pesar de realizar un número mayor de iteraciones.

6 CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se presentan las conclusiones obtenidas a partir del trabajo grado realizado y los posibles trabajos futuros en el mismo campo de estudio.

6.1 CONCLUSIONES

- En sistemas MIMO donde se alcancen en el receptor valores de SNR mayores a 15 dB, los algoritmos LRA mejoran el comportamiento de un detector lineal de tal forma que su desempeño presenta un comportamiento similar al del algoritmo ML con requerimientos de SNR menores que los obtenidos con el algoritmo ZF.
- Al utilizar algoritmos LRA para la detección de señales en sistemas MIMO, se obtiene un mejor desempeño utilizando arreglos de 4 antenas, lo mismo sucede con el algoritmo ML mientras que con el algoritmo ZF el desempeño es mejor en sistemas MIMO 2x2 debido a su naturaleza lineal y su ineficiente manejo del ruido.
- La modulación afecta de forma general el desempeño de los algoritmos ML, LRA y ZF, ya que si se aumenta el orden de modulación el sistema es menos tolerante al ruido porque se utiliza una constelación con un mayor número de símbolos que aumenta la interferencia intersimbólica. Lo cual causa que los sistemas MIMO con modulación 4QAM tengan mejor desempeño que los sistemas MIMO con 16QAM.
- El número de iteraciones realizadas por un algoritmo para efectuar la reducción *lattice* de una matriz no es un indicador confiable de su complejidad. Algoritmos como el de *Seysen* realizan iteraciones complejas que requieren de gran capacidad de procesamiento, mientras que otros como el LLL se caracterizan por iteraciones simples, es por esto que se necesita conocer el tiempo utilizado por cada algoritmo al realizar la reducción para efectuar una comparación válida.
- Los algoritmos LRA presentan una mejora en el desempeño introduciendo pequeños tiempos adicionales de procesamiento por trama. En un sistema real esto podría presentar retardos considerables en una comunicación, por lo que el análisis del desempeño considerando únicamente la BER, no es un criterio confiable para elegir el algoritmo más apropiado en la etapa de detección de señales en un sistema MIMO.
- En sistemas MIMO 2x2 con modulaciones 4QAM o 16QAM, los algoritmos LRA LLL y *Seysen* presentan un desempeño prácticamente idéntico, pero al ser más rápido el LLL, se considera éste como el algoritmo más apropiado para implementar la detección de señales en un



sistema MIMO para obtener valores de BER menores sin aumentar los requerimientos computacionales.

- Con respecto a los algoritmos LRA en sistemas MIMO 4x4, el algoritmo *Seysen* presenta un mejor desempeño, pero los tiempos utilizados para la reducción son aproximadamente 3 veces mayores que los del LLL, por lo tanto su utilización implica un aumento considerable de la carga computacional para obtener leves mejoras sobre el LLL.
- Aunque el desempeño del algoritmo ML siempre será mejor, los algoritmos LRA son una alternativa adecuada para implementar la detección de señales en sistemas MIMO reales que cuenten con altos niveles de SNR en recepción, pues bajo estas condiciones tienen un desempeño similar al del algoritmo ML con tiempos de procesamiento mucho menores, especialmente el LLL.

6.2 TRABAJOS FUTUROS

- Implementar el algoritmo *Seysen* con métodos de selección de lambda como *Lazy* y *Greedy* [22] y comparar con la implementación utilizada en este trabajo de grado.
- Analizar el comportamiento del algoritmo LLL para la detección de señales en sistemas MIMO con diferentes valores para el parámetro de reducción delta y determinar su efecto en el desempeño del sistema.
- Realizar un análisis comparativo de los algoritmos LRA para la detección de señales en sistemas MIMO utilizando otros algoritmos de detección como el MMSE.
- Analizar el comportamiento del sistema con modelos de canal definidos por el estándar IEEE 802.11n.
- Analizar el desempeño de algoritmos LRA en un sistema MIMO sin suponer un conocimiento perfecto del canal en el receptor.

REFERENCIAS

- [1] E. Larsson, "MIMO Detection Methods: How They Work," *IEEE signal processing magazine*, vol. 26, pp. 91-95, Mayo 2009.
- [2] C. Sanchis, "Estudio Experimental de Algoritmos Espacio-Temporales para Sistemas Multiantena en Túneles," *Departamento de Tecnologías de la Información y las Comunicaciones*, Universidad Politécnica de Cartagena, 2011.
- [3] H. Yao y G. Wornell, "Lattice-Reduction-Aided Detectors for MIMO Communication Systems," Dept. EECS and Research Laboratory of Electronics MIT, Cambridge MA 02139, USA, 2002.
- [4] M. Rumney, "LTE and the Evolution to 4G Wireless: Design and Measurement Challenges," *California: Agilent Technologies*, pp. 43, 44 y 48, 2009.
- [5] T. Shimokawa y T. Fujino, "Iterative Lattice Reduction Aided MMSE List Detection in MIMO System," The University of Electro-Communications Chofugaoka 1-5-1, Chofu-shi, Tokyo, 182-8585, Japan.
- [6] X. Wang, et al, "Decision Feedback Aided Detection Based on Lattice Reduction in MIMO Systems," *Vehicular Technology Conference*, vol. 4, pp. 1708-1712, 2006.
- [7] K. Su, et al, "Coset-based lattice detection for MIMO systems," *Department of Electrical and Computer Engineering*, University of Toronto, 2007.
- [8] T. Fujino, et al, "A Lattice-Reduction Aided Detection by Combining LLL Algorithm and Gram-Schmidt Procedure," The University of Electro-Communications, Chofu-shi, Tokyo, 182-8585, Japan, 2009.
- [9] G. Medina y P. Caicedo, "Análisis de desempeño del nivel físico de un sistema MIMO punto a punto," Tesis de pregrado, Universidad del Cauca, Popayán, Colombia, 2010.
- [10] D. Gesbert, M. Shaffi, P. Smith y A. Naguib, "From theory to practice: An overview of MIMO space-time coded wireless systems," *IEEE journal on selected areas in communications*, vol. 21, pp. 281-302, Abril 2003.
- [11] M. Elo, "SISO to MIMO: Moving Communications from Single-Input Single-Output to Multiple-Input Multiple-Output," *Keithley*, no. 2885, pp. 899-907, Octubre 2007.
- [12] L. Bazdresch, "Complexité et performance des récepteurs MIMO," Doctorado, École Nationale Supérieure des Télécommunications, París, Francia, 2004.
- [13] V. Kühn, "Communications over MIMO channels: Applications to cdma and multiple antennas systems," John Wiley & Sons, pp. 277-303, 2006.



- [14] S. Haykin, *Sistemas de comunicación*: México: Limusa wiley, 2002.
- [15] K. Su, "Efficient maximum likelihood detection for communication over multiple input multiple output channels," Tesis de pregrado, Universidad de Cambridge, Cambridge, 2005.
- [16] D. Wübben, D. Seethaler, J. Jaldén y G. Matz, "A survey of lattice reduction techniques with applications to wireless communications," *IEEE signal processing magazine*, vol. 28, pp. 70-91, Mayo 2011.
- [17] W. Zhang, S. Qiao y Y. Wei, "Practical HKZ and Minkowski Lattice Reduction Algorithms," Departamento de computación y software universidad McMaster, Ontario, Canadá 2011.
- [18] H. Daudé, P. Flajolet y B. Vallée, "An average-case analysis of the Gaussian algorithm for Lattice Reduction," *Combinatorics, probability and computing*, vol. 6, pp. 397 - 433, Diciembre, 1997.
- [19] P. Nguyen y B. Vallée, "The LLL algorithm: Survey and applications," ed. Alemania: Springer, pp. 47 - 50, 1998.
- [20] C. Ling, "Approximate Lattice Decoding: Primal Versus Dual Basis Reduction," ed. King's College London, London, WC2R 2LS, UK: Department of Electronic Engineering, 2006.
- [21] C. Ling, "Towards characterizing the performance of approximate lattice decoding," Munich, Germany. Apr. 2006.
- [22] B. LaMacchia, "Basis Reduction Algorithms and subset sum problems," Tesis de maestría, Instituto de tecnología de Massachusetts, Massachusetts, Estados Unidos, 1991.
- [23] D. Seethaler, J. Jaldén y G. Matz, "Worst- and average-case complexity of LLL lattice reduction in MIMO wireless systems," *Acoustics, Speech and Signal Processing*, 2008.
- [24] W. Zhang, S. Qiao y Y. Wei, "Practical algorithms for constructing HKZ and Minkowski reduced bases," *Signal Processing*, Abril 20, 2011.
- [25] C. Ling, W. Mow y L. Gan, "Complex Lattice Reduction Algorithm for Low-Complexity MIMO Detection," *Global Telecommunications Conference, IEEE GLOBECOM '05*. Diciembre 2005.
- [26] C. Ling, et al, "Dual-Lattice Ordering and Partial Lattice Reduction for SIC-Based MIMO Detection," 2009.
- [27] A. Murray y S. Weller, "Enhanced Lattice-Reduction Aided detection for MIMO systems," *NEWCOM - ACoRN Joint Workshop, School of Electrical Engineering and Computer Science, University of Newcastle*, 2008.
- [28] C. Ling y N. Howgrave-Graham, "Effective LLL Reduction for Lattice Decoding," Junio, 2007.



- [29] H. Yao y. G. Wornell, "Lattice-Reduction-Aided Detectors for MIMO Communication Systems," *Dept. EECS and Research Laboratory of Electronics*, Noviembre, 2002.
- [30] D. Wubben, V. Kuhn, y K. Kammeyer, "On the Robustness of Lattice-Reduction Aided Detectors in Correlated MIMO Systems," *Department of Communications Engineering, University of Bremen*, 2004.
- [31] C. Windpassinger y. R. Fischer, "Optimum and Sub-Optimum Lattice-Reduction-Aided Detection and Precoding for MIMO Communications," *Lehrstuhl für Information supertragung, Universität Erlangen-Nürnberg*, 2003.
- [32] L. Bruderer, C. Senning y A. Burg, "Low-Complexity Seysen's Algorithm based Lattice Reduction-Aided MIMO Detection for Hardware Implementations," ed. *ETH Zurich, Switzerland: Integrated Systems Laboratory*, 2011.
- [33] K HongSun An, et al, "Lattice Reduction Aided Precoding for Multiuser MIMO using Seysen's Algorithm," ed. *Inha, Korea: Inha University*, 2009.
- [34] H. Daudé, P. Flajolet y B. Vallée, "An average-case analysis of the gaussian algorithm for lattice reduction," *Institut National De Recherche en Informatique et Automatique*, 1997.
- [35] M. Seysen, "SIMULTANEOUS REDUCTION OF A LATTICE BASIS AND ITS RECIPROCAL BASIS," *Combinatorica*, vol. 13, pp. 363-376, 1993.
- [36] B. La Macchia, "Basis Reduction Algorithms and subset sum problems," *Artificial Intelligence Laboratory and Department of electrical Engineering and computer cience, Massachusetts Institute of Technology, Massachusetts, EEUU*, 1991.
- [37] E. Astaiza, H. Bermudez, y P. Muñoz, *Simulación de Sistemas de Telecomunicaciones. Arte e Imagen*, 2007.
- [38] MathWorks Labs., "MATLAB, The Language of Technical Computing.," 2012.
- [39] M. Baudin. Noviembre, 2010, *Introduction to SCILAB*.
- [40] C. Muñoz y. T. Muñoz, "ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS ALGORITMOS FSD Y QRD-M," *Departamento de Telecomunicaciones, Universidad del Cauca, Popayán, Colombia*, 2012.
- [41] M. Chouayakh, A. Knopp y B. Lankl, "Low Complexity Two Stage Detection Scheme for MIMO Systems," *Institute for Communications Engineering, Munich University of the Bundeswehr, 85579 Neubiberg, Germany*, 2007.
- [42] S.-H. Moon, et al. 2010, *Statistical Precoder Design for Spatial Multiplexing Systems in Correlated MIMO Fading Channels. IEEE 71st Vehicular Technology Conf. Proc.*, 1-5.



- [43] I. Jelovčan y T. Javornik. May, 2007, Low Complexity MIMO Detection Algorithm. *Elektrotehniški vestnik* 74(4), 236-240.

ANEXO A

DIFERENCIA DE VALORES DE DESEMPEÑO

A continuación se muestran las tablas A 1 a A 8, que son tablas comparativas de valores SNR según BER de los algoritmos LRA para los cuatro escenarios a partir de las cuales se calcularon las diferencias utilizadas en el capítulo 4 para la validación de resultados.

Tabla A 1. Comparación LLL escenario 1 [\[42\]](#)

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	7	9.76	2.76
10^{-2}	15	19.47	4.47
10^{-3}	20.8	25.96	5.16
10^{-4}	25.7	31.91	6.21
		Media	4.65
		Desv. Est.	1.44893064
		C.V.	0.311597987

Tabla A 2. Comparación LLL escenario 2 [\[43\]](#)

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	6.4	14.57	8.17
10^{-2}	14.7	25.66	10.96
10^{-3}	20.3	32.31	12.01
10^{-4}	25.5	38.63	13.13
		Media	11.0675
		Desv. Est.	2.125188227
		C.V.	0.192020621



Tabla A 3. Comparación LLL escenario 3 [41]

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	5	13.48	8.48
10^{-2}	10.6	21.06	10.46
10^{-3}	13.2	25.06	11.86
10^{-4}	15.6	28.52	12.92
		Media	10.93
		Desv. Est.	1.919062271
		C.V.	0.175577518

Tabla A 4. Comparación LLL escenario 4[7]

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	9	18.83	9.83
10^{-2}	15	27.53	12.53
10^{-3}	19	32.15	13.15
10^{-4}	21.5	35.56	14.06
		Media	12.3925
		Desv. Est.	1.820226634
		C.V.	0.14688131

Tabla A 5. Comparación *Seysen* escenario 1 [42]

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	7	9.75	2.75
10^{-2}	15	19.46	4.46
10^{-3}	20.8	25.95	5.15
10^{-4}	25.7	31.88	6.18
		Media	4.635
		Desv. Est.	1.44176975
		C.V.	0.31106143



Tabla A 6. Comparación *Seysen* escenario 2 [43]

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	6.4	14.58	8.18
10^{-2}	14.7	25.65	10.95
10^{-3}	20.3	32.33	12.03
10^{-4}	25.5	38.63	13.13
		Media	11.0725
		Desv. Est.	2.12380751
		C.V.	0.19180921

Tabla A 7. Comparación *Seysen* escenario 3 [41]

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	5	13.48	8.48
10^{-2}	10.6	21	10.4
10^{-3}	13.2	24.87	11.67
10^{-4}	15.6	27.93	12.33
		Media	10.72
		Desv. Est.	1.69455992
		C.V.	0.15807462

Tabla A 8. Comparación *Seysen* escenario 4 [41]

BER	Referencia [dB]	Simulado [dB]	Diferencia [dB]
10^{-1}	7	18.83	11.83
10^{-2}	12.8	27.45	14.65
10^{-3}	16.2	31.85	15.65
10^{-4}	18.8	35.27	16.47
		Media	14.65
		Desv. Est.	2.02194626
		C.V.	0.13801681

ANEXO B

DIFERENCIA DE VALORES DE DESEMPEÑO ENTRE ESCENARIOS

Las tablas B 1 a B 8 contienen una comparación de las diferencias de requerimientos de SNR entre los diferentes algoritmos para cada escenario.

Tabla B 1. Tabla de diferencias de requerimientos de SNR para el escenario 1.

BER	SNR [dB]								
	ML	ZF	LLL	Seysen	ML - LLL	ML - Seysen	ZF - LLL	ZF - Seysen	ML - ZF
10^{-1}	3.46	9.72	9.76	9.75	-6.3	-6.29	-0.04	-0.03	-6.26
10^{-2}	10.76	22	19.47	19.46	-8.71	-8.7	2.53	2.54	-11.24
10^{-3}	15.78	32.12	25.96	25.95	-10.18	-10.17	6.16	6.17	-16.34
10^{-4}	21.51	42.27	31.91	31.88	-10.4	-10.37	10.36	10.39	-20.76
10^{-5}	25.65	51.88	38.55	38.55	-12.9	-12.9	13.33	13.33	-26.23
10^{-6}	30	57	42	42	-12	-12	15	15	-27



Tabla B 2. . Tabla de diferencias de requerimientos de SNR para el escenario 2.

BER	SNR [dB]								
	ML	ZF	LLL	Seysen	ML - LLL	ML - Seysen	ZF - LLL	ZF - Seysen	ML - ZF
10^{-1}	10.36	14.4	14.57	14.58	-4.21	-4.22	-0.17	-0.18	-4.04
10^{-2}	18.71	27.55	25.66	25.65	-6.95	-6.94	1.89	1.9	-8.84
10^{-3}	24.01	38.29	32.31	32.33	-8.3	-8.32	5.98	5.96	-14.28
10^{-4}	29.09	48.43	38.63	38.63	-9.54	-9.54	9.8	9.8	-19.34
10^{-5}	34.63	57.45	43.14	43.14	-8.51	-8.51	14.31	14.31	-22.82
10^{-6}	39	-	48	48	-9	-9	-	-	-

Tabla B 3. Tabla de diferencias de requerimientos de SNR para el escenario 3.

BER	SNR [dB]								
	ML	ZF	LLL	Seysen	ML - LLL	ML - Seysen	ZF - LLL	ZF - Seysen	ML - ZF
10^{-1}	4.08	13.49	13.48	13.48	-9.4	-9.4	0.01	0.01	0
10^{-2}	10.11	26.04	21.06	21	-10.95	-10.89	4.98	5.04	0.06
10^{-3}	13.74	36.03	25.06	24.87	-11.32	-11.13	10.97	11.16	0.19
10^{-4}	16.46	45.89	28.52	27.93	-12.06	-11.47	17.37	17.96	0.59
10^{-5}	19.04	54.75	31.78	30.42	-12.74	-11.38	22.97	24.33	1.36
10^{-6}	24	60	36	33	-12	-9	24	27	3



Tabla B 4. Tabla de diferencias de requerimientos de SNR para el escenario 3.

BER	SNR [dB]								
	ML	ZF	LLL	Seysen	ML - LLL	ML - Seysen	ZF - LLL	ZF - Seysen	ML - ZF
10^{-1}	12.05	18.45	18.83	18.83	-6.78	-6.78	-0.38	-0.38	0
10^{-2}	18.19	31.52	27.53	27.45	-9.34	-9.26	3.99	4.07	0.08
10^{-3}	21.84	41.78	32.15	31.85	-10.31	-10.01	9.63	9.93	0.3
10^{-4}	25	51.5	35.56	35.27	-10.56	-10.27	15.94	16.23	0.29
10^{-5}	27.76	-	39.01	37.31	-11.25	-9.55	-	-	1.7
10^{-6}	30	-	43	39	-13	-9	-	-	4

Tablas de diferencia de requerimientos de SNR entre los diferentes escenarios para los algoritmos ZF, ML y LRA.

Tabla B 5. Tabla de diferencias de requerimientos de SNR para los algoritmos LRA, ZF y ML para los escenarios 1 y 2.

BER	SNR [dB]											
	ML Esc 1	ZF Esc 1	LLL Esc 1	Seysen Esc 1	ML Esc 2	ZF Esc 2	LLL Esc 2	Seysen Esc 2	ML Esc 1 - Esc 2	ZF Esc 1 - Esc 2	LLL Esc 1 - Esc 2	Seysen Esc 1 - Esc 2
10^{-1}	3.46	9.72	9.76	9.75	10.36	14.4	14.57	14.58	-6.9	-4.68	-4.81	-4.83
10^{-2}	10.76	22	19.47	19.46	18.71	27.55	25.66	25.65	-7.95	-5.55	-6.19	-6.19
10^{-3}	15.78	32.12	25.96	25.95	24.01	38.29	32.31	32.33	-8.23	-6.17	-6.35	-6.38
10^{-4}	21.51	42.27	31.91	31.88	29.09	48.43	38.63	38.63	-7.58	-6.16	-6.72	-6.75
10^{-5}	25.65	51.88	38.55	38.55	34.63	57.45	43.14	43.14	-8.98	-5.57	-4.59	-4.59
10^{-6}	30	57	42	42	39	-	48	48	-9	-	-6	-6



Tabla B 6. Tabla de diferencias de requerimientos de SNR para los algoritmos LRA, ZF y ML para los escenarios 3 y 4.

BER	SNR [dB]											
	ML Esc 3	ZF Esc 3	LLL Esc 3	Seysen Esc 3	ML Esc 4	ZF Esc 4	LLL Esc 4	Seysen Esc 4	ML Esc 3 - Esc 4	ZF Esc 3 - Esc 4	LLL Esc 3 - Esc 4	Seysen Esc 3 - Esc 4
10^{-1}	4.08	13.49	13.48	13.48	12.05	18.45	18.83	18.83	-7.97	-4.96	-5.35	-5.35
10^{-2}	10.11	26.04	21.06	21	18.19	31.52	27.53	27.45	-8.08	-5.48	-6.47	-6.45
10^{-3}	13.74	36.03	25.06	24.87	21.84	41.78	32.15	31.85	-8.1	-5.75	-7.09	-6.98
10^{-4}	16.46	45.89	28.52	27.93	25	51.5	35.56	35.27	-8.54	-5.61	-7.04	-7.34
10^{-5}	19.04	54.75	31.78	30.42	27.76	-	39.01	37.31	-8.72	-	-7.23	-6.89
10^{-6}	24	60	36	33	30	-	43	39	-6	-	-7	-6

Tabla B 7. Tabla de diferencias de requerimientos de SNR para los algoritmos LRA, ZF y ML para los escenarios 1 y 3.

BER	SNR [dB]											
	ML Esc 1	ZF Esc 1	LLL Esc 1	Seysen Esc 1	ML Esc 3	ZF Esc 3	LLL Esc 3	Seysen Esc 3	ML Esc 1 - Esc 3	ZF Esc 1 - Esc 3	LLL Esc 1 - Esc 3	Seysen Esc 1 - Esc 3
10^{-1}	3.46	9.72	9.76	9.75	4.08	13.49	13.48	13.48	-0.62	-3.77	-3.72	-3.73
10^{-2}	10.76	22	19.47	19.46	10.11	26.04	21.06	21	0.65	-4.04	-1.59	-1.54
10^{-3}	15.78	32.12	25.96	25.95	13.74	36.03	25.06	24.87	2.04	-3.91	0.9	1.08
10^{-4}	21.51	42.27	31.91	31.88	16.46	45.89	28.52	27.93	5.05	-3.62	3.39	3.95
10^{-5}	25.65	51.88	38.55	38.55	19.04	54.75	31.78	30.42	6.61	-2.87	6.77	8.13
10^{-6}	30	57	42	42	24	60	36	33	6	-3	6	9



Tabla B 8. Tabla de diferencias de requerimientos de SNR para los algoritmos LRA, ZF y ML para los escenarios 2 y 4.

BER	SNR [dB]											
	ML Esc 2	ZF Esc 2	LLL Esc 2	Seysen Esc 2	ML Esc 4	ZF Esc 4	LLL Esc 4	Seysen Esc 4	ML Esc 2 - Esc 4	ZF Esc 2 - Esc 4	LLL Esc 2 - Esc 4	Seysen Esc 2 - Esc 4
10^{-1}	10.36	14.4	14.57	14.58	12.05	18.45	18.83	18.83	-1.69	-4.05	-4.26	-4.25
10^{-2}	18.71	27.55	25.66	25.65	18.19	31.52	27.53	27.45	0.52	-3.97	-1.87	-1.8
10^{-3}	24.01	38.29	32.31	32.33	21.84	41.78	32.15	31.85	2.17	-3.49	0.16	0.48
10^{-4}	29.09	48.43	38.63	38.63	25	51.5	35.56	35.27	4.09	-3.07	3.07	3.36
10^{-5}	34.63	57.45	43.14	43.14	27.76	-	39.01	37.31	6.87	-	4.13	5.83
10^{-6}	39	-	48	48	30	-	43	39	9	-	5	9



ANEXO C

CURVAS DE DESEMPEÑO POR ALGORITMOS Y MODULACIÓN

Las figuras C 1 a C 6 presentan curvas de desempeño agrupadas por algoritmos y por esquema de modulación.

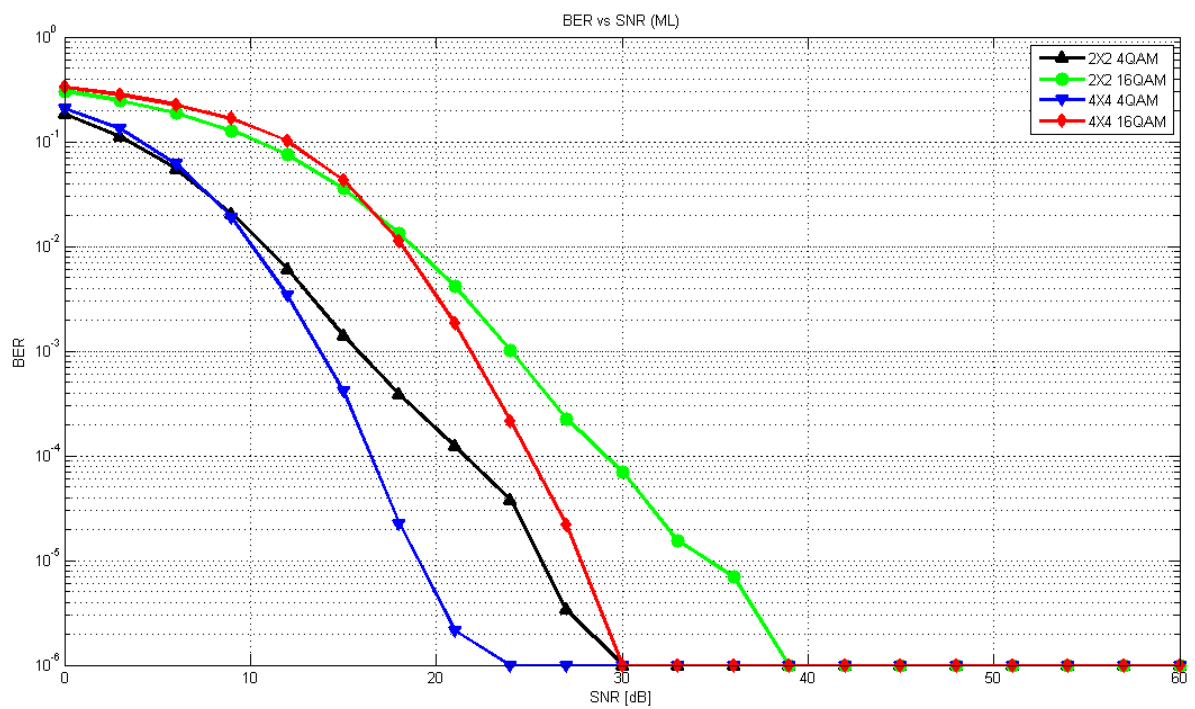


Figura C 1. Curvas de desempeño del algoritmo ML para los 4 escenarios simulados

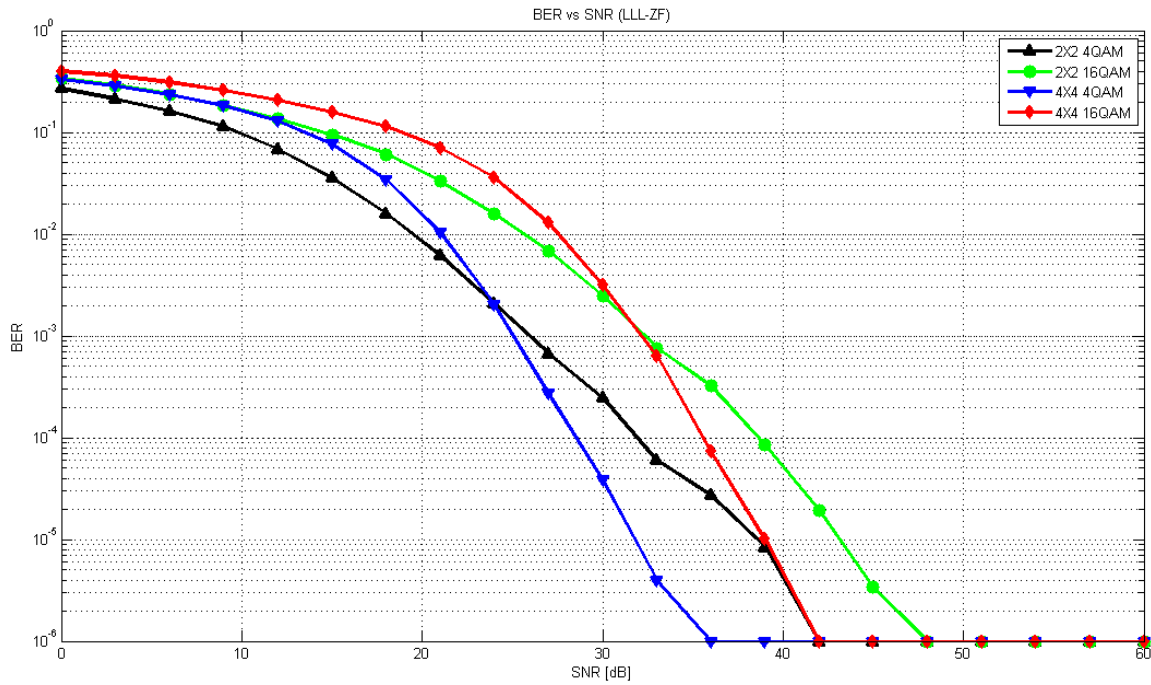


Figura C 2. Curvas de desempeño del algoritmo LLL para los 4 escenarios simulados

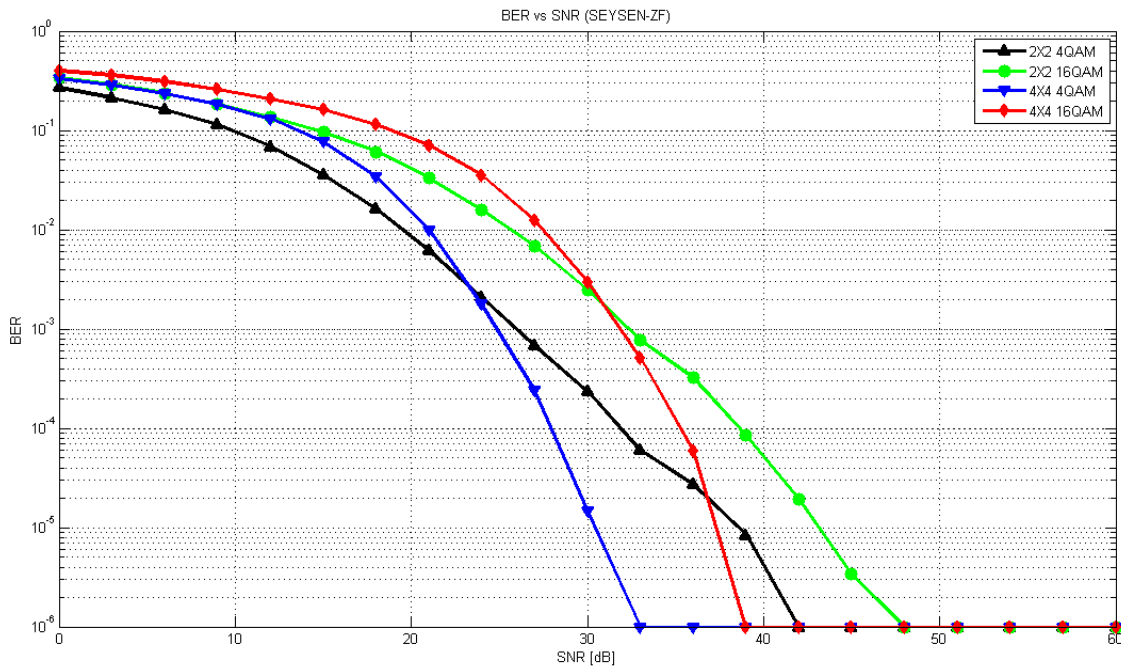


Figura C 3. Curvas de desempeño del algoritmo Seysen para los 4 escenarios simulados

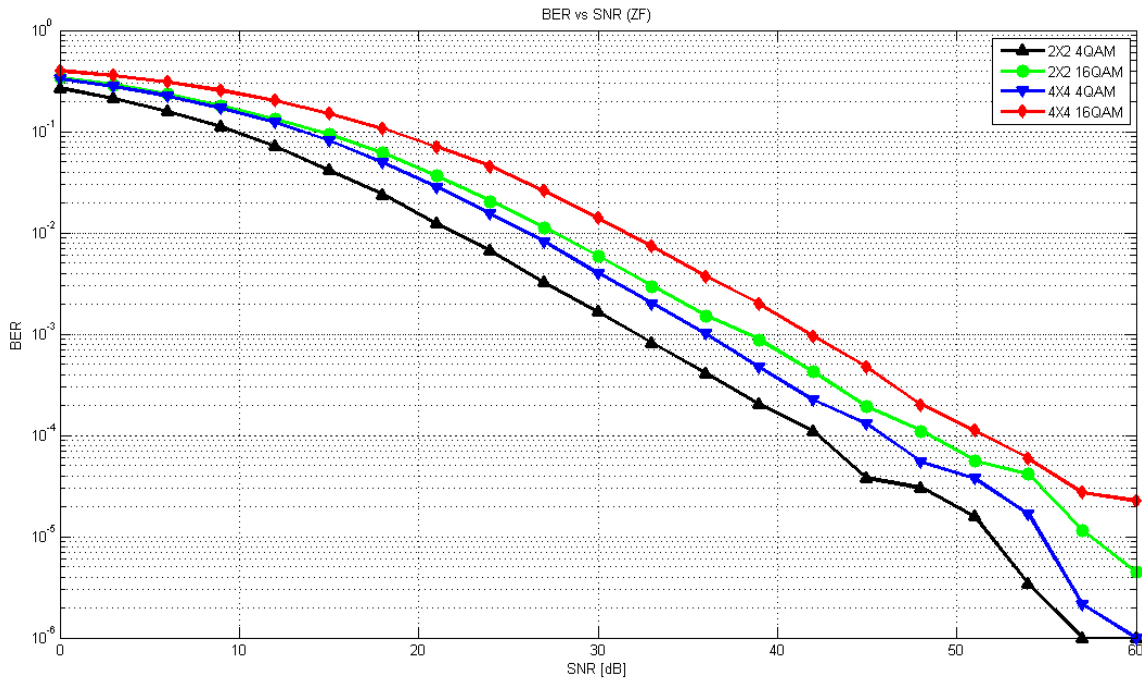


Figura C 4. Curvas de desempeño del algoritmo ZF para los 4 escenarios simulados

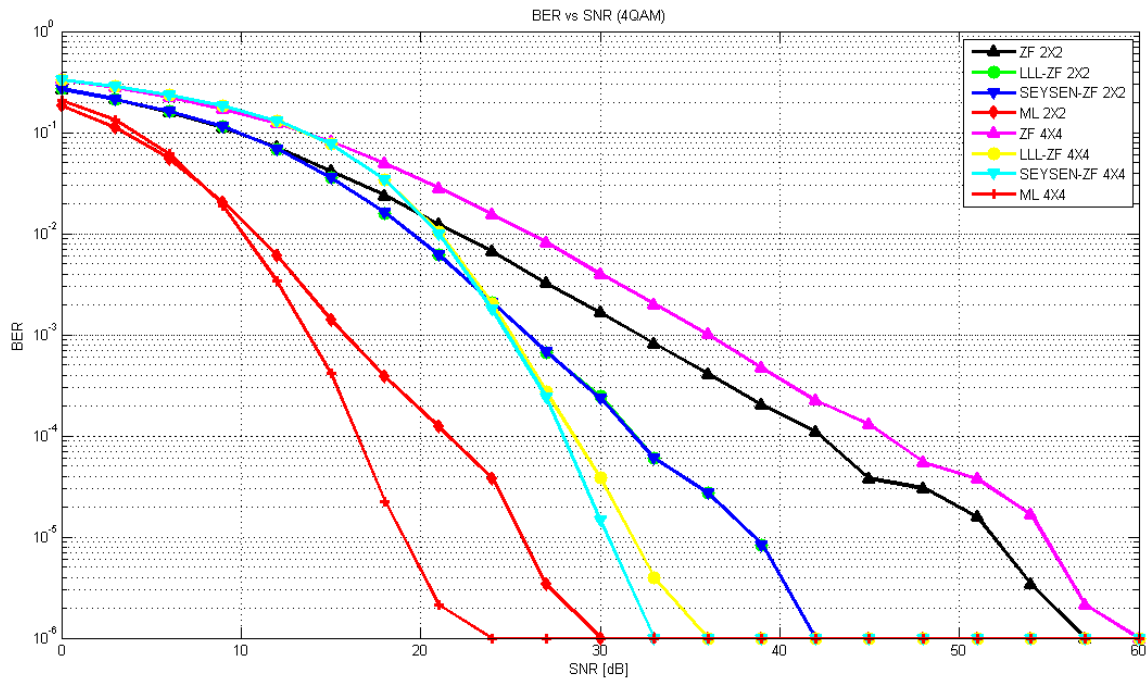


Figura C 5. Curvas de desempeño de los algoritmos simulados para modulación 4QAM

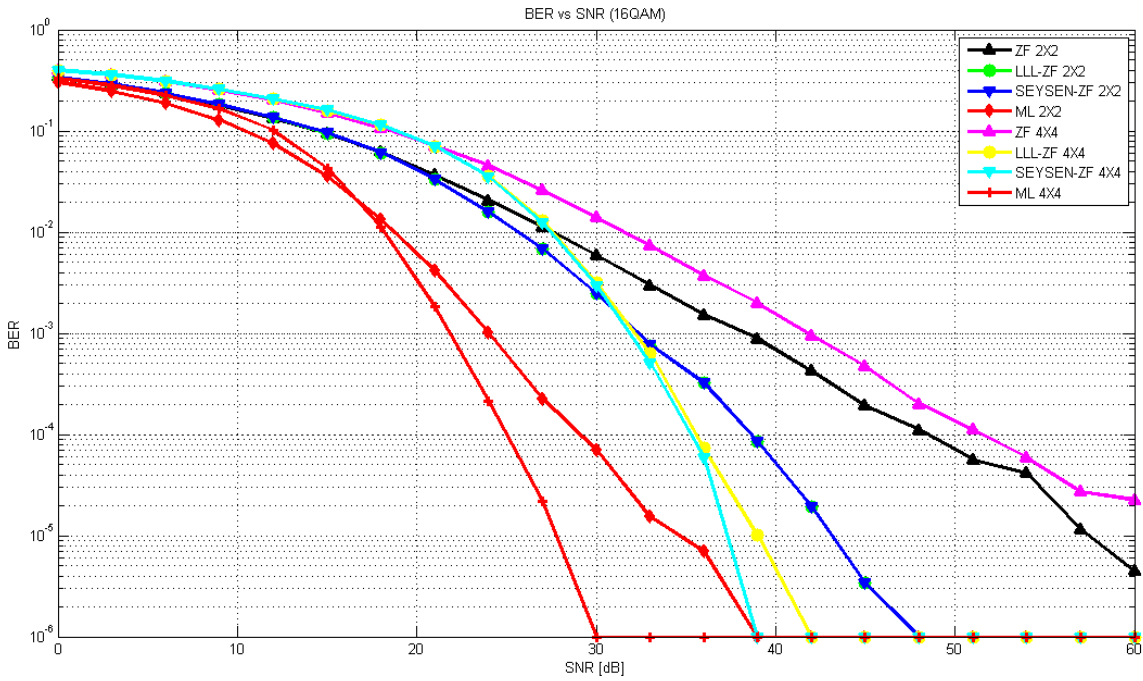


Figura C 6. Curvas de desempeño de los algoritmos simulados para modulación 16QAM