



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**ANÁLISIS DE LAS CODIFICACIONES DE CANAL: LDPC Y  
TURBO CÓDIGO, UTILIZANDO MODULACIONES NO  
UNIFORMES SOBRE CANALES AWGN CON  
DESVANECIMIENTO**

**MÓNICA JEANETTE FLORES VILLAFUERTE**

**Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo,  
presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH,  
como requisito parcial para la obtención del grado de:**

**MAGÍSTER EN SISTEMAS DE TELECOMUNICACIONES**

**RIOBAMBA – ECUADOR**

Noviembre 2021



## ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

### CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

**El Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, titulado ANÁLISIS DE LAS CODIFICACIONES DE CANAL: LDPC Y TURBO CÓDIGO, UTILIZANDO MODULACIONES NO UNIFORMES SOBRE CANALES AWGN CON DESVANECIMIENTO, de responsabilidad de la señora Mónica Jeanette Flores Villafuerte ha sido prolijamente revisado y se autoriza su presentación.**

Tribunal:

ING. LUIS EDUARDO HIDALGO ALMEIDA; PHD.  
**PRESIDENTE**

LUIS  
EDUARDO  
HIDALGO  
ALMEIDA

Firmado digitalmente por LUIS EDUARDO HIDALGO ALMEIDA  
Nombre de reconocimiento (DN): c=EC, o=BANCO CENTRAL DEL ECUADOR, ou=ENTIDAD DE CERTIFICACION DE INFORMACION-SCIRSE=QUITO, serialNumber=000045780, cn=LUIS EDUARDO HIDALGO ALMEIDA  
Fecha: 2021.11.22 15:50:09 -05'00'

ING. OLMEDO CIFUENTES GONZALO FERNANDO, PHD.  
**DIRECTOR**

 Firmado electrónicamente por:  
**GONZALO FERNANDO  
OLMEDO CIFUENTES**

ING. LARA CUEVA ROMAN ALCIDES, PHD.  
**MIEMBRO**

 Firmado electrónicamente por:  
**ROMAN  
ALCIDES LARA**

ING. PAZMIÑO ARMIJOS ALEXANDRA ORFELINA, M.Sc.  
**MIEMBRO**

 Firmado electrónicamente por:  
**ALEXANDRA  
ORFELINA PAZMINO  
ARMIJOS**

Riobamba, noviembre 2021

## DERECHOS INTELECTUALES

Yo, Mónica Jeanette Flores Villafuerte, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.



---

Mónica Jeanette Flores Villafuerte

No. Cédula: 180402033-5

## **DEDICATORIA**

A Dios por ser el pilar fundamental en mi hogar y por darme la sabiduría para concluir este proyecto.

A mi esposo por su apoyo incondicional en la realización de mi maestría.

A mis hijos Samantha y Guillermo por ser mi razón de vivir y superarme para brindarles un mejor futuro.

A mi tutor por toda la ayuda y apoyo que me ha brindado en la realización y desarrollo de este proyecto, que gracias a sus conocimientos ha sido un excelente guía.

A mis padres por creer siempre en mí y apoyarme en todos los momentos de mi vida.

## **AGRADECIMIENTO**

A todas las personas que en estos años de estudio me apoyaron incondicionalmente; a mis padres por estar conmigo apoyándome siempre en cada paso que doy, a mi esposo por sus palabras de aliento y paciencia durante todo este tiempo de estudio; y a mi tutor por su guía y apoyo durante la realización de este proyecto.

## TABLA DE CONTENIDO

<b>RESUMEN .....</b>	<b>XII</b>
<b>ABSTRACT.....</b>	<b>XIII</b>
<b>CAPÍTULO I.....</b>	<b>1</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1. Antecedentes .....	1
1.2. Situación problemática .....	2
1.2.1. Planteamiento del problema.....	2
1.2.2. Formulación del problema .....	3
1.3. Preguntas directrices o específicas de la investigación .....	3
1.4. Justificación de la investigación .....	4
1.5. Objetivo general de investigación .....	6
1.6. Objetivos específicos de investigación .....	6
1.7. Hipótesis.....	6
<b>CAPÍTULO II .....</b>	<b>7</b>
<b>MARCO DE REFERENCIA .....</b>	<b>7</b>
2.1. Ruido AWGN.....	7
2.2. Distribución Rayleigh.....	7
2.3. Desvanecimiento Rayleigh .....	7
2.4. Relación Señal a Ruido .....	9
2.5. Tasa de Error de Bit (BER) .....	9
2.6. Capacidad CM.....	9
2.7. Capacidad BICM.....	9
2.8. Constelaciones no uniformes.....	10
2.9. Codificación de canal .....	11
2.9.1. Códigos de bloque lineal .....	13
2.9.2. Códigos Convolutivos .....	13

2.9.3.	Codificación LDPC ( <i>Low Density Parity Check</i> ).....	14
2.9.3.1.	Corrección de errores utilizando chequeo de paridad .....	14
2.9.3.2.	Representación Gráfica .....	15
2.9.4.	Codificación turbo .....	16
2.9.4.1.	Código de producto .....	17
2.9.4.2.	Códigos convolucionales.....	17
2.10.	Límite de Shannon .....	18
2.11.	Modelo del Sistema de Comunicaciones .....	19
2.11.1.	Fuente de Información .....	19
2.11.2.	Codificación de Canal.....	20
2.11.3.	Modulación No Uniforme.....	20
2.11.4.	Canal AWGN con Desvanecimiento.....	20
2.11.5.	Demodulación.....	20
2.11.6.	Decodificador de Canal.....	20
2.11.7.	Recepción de la Información .....	20
2.12.	Codificación LDPC, turbo SOVA y LOG-MAP utilizando modulaciones no uniformes .....	21
2.12.1.	Límite CM para Constelaciones uniformes .....	21
2.12.2.	Límite BICM para constelaciones uniformes .....	21
2.12.3.	Constelaciones QAM no uniformes (NUC-QAM).....	24
2.12.4.	Optimización para desvanecimiento .....	26
2.13.	CODIFICACIÓN LDPC (Low Density Parity Check).....	28
2.13.1.	Codificación.....	29
2.13.1.2.	Matriz Generadora .....	30
2.13.1.3.	Palabra codificada.....	30
2.13.2.	Decodificación.....	30
2.14.	CODIFICACIÓN TURBO .....	34
2.14.1.	Codificación.....	34
2.14.1.1.	Códigos convolucionales.....	35

2.14.1.2. Intercalado.....	35
2.14.1.3. Perforación.....	36
2.14.2. Decodificación.....	37
2.14.2.1. Algoritmo MAP.....	38
2.14.2.2. Algoritmo de Log-MAP.....	40
2.14.2.3. SOVA (Algoritmo de Viterbi de salida suave).....	41
<b>CAPÍTULO III.....</b>	<b>44</b>
<b>METODOLOGÍA DE INVESTIGACIÓN .....</b>	<b>44</b>
3.1. Tipo y diseño de investigación .....	44
3.2. Métodos de investigación .....	44
3.3. Enfoque de la investigación.....	45
3.4. Alcance de la investigación .....	45
3.5. Unidad de análisis .....	45
3.6. Técnica de recolección de datos primarios y secundarios.....	45
3.7. Instrumentos de recolección de datos primarios y secundarios.....	45
3.8. Instrumentos para procesar datos recopilados .....	45
<b>CAPÍTULO IV .....</b>	<b>46</b>
<b>RESULTADOS Y ANÁLISIS .....</b>	<b>46</b>
4.1. Resultados para un canal AWGN con desvanecimiento al utilizar codificación LDPC y Constelaciones Uniformes versus No Uniformes.....	46
4.2. Resultados para un canal AWGN con desvanecimiento al utilizar codificación Turbo Log-Map y SOVA utilizando Constelaciones Uniformes versus No Uniformes.....	49
4.3. Resultados para un canal AWGN con desvanecimiento al utilizar codificación LDPC y Turbo con Constelaciones No Uniformes.....	54
<b>CONCLUSIONES.....</b>	<b>57</b>
<b>RECOMENDACIONES.....</b>	<b>58</b>
<b>GLOSARIO.....</b>	<b>59</b>
<b>BIBLIOGRAFÍA</b>	
<b>ANEXOS</b>	

## ÍNDICE DE TABLAS

<b>Tabla 1-4.</b> Métricas de desempeño, para 64QAM, codificación LDPC .....	46
<b>Tabla 2-4.</b> Métricas de desempeño, para 64QAM, codificación Turbo .....	49

## ÍNDICE DE FIGURAS

<b>Figura 1-2.</b> Capacidad BICM , UC vs límite de Shannon.....	10
<b>Figura 2-2.</b> Constelación 16-QAM uniforme .....	10
<b>Figura 3-2.</b> Constelación 64 NUC-QAM .....	11
<b>Figura 4-2.</b> Codificación de bloque lineal .....	13
<b>Figura 5-2.</b> Codificador convolucional de tasa $\frac{1}{2}$ .....	14
<b>Figura 6-2.</b> Ejemplo Matriz de Verificación de Paridad H (4x6).....	15
<b>Figura 7-2.</b> Representación de la matriz H con Grafo de Tanner .....	16
<b>Figura 8-2.</b> Codificación Turbo.....	16
<b>Figura 9-2.</b> Código de producto (TPC).....	17
<b>Figura 10-2.</b> Límite de Shannon.....	18
<b>Figura 11-2.</b> Diagrama de bloques del Sistema de Comunicaciones .....	19
<b>Figura 12-2.</b> Eje PAM 16 QAM.....	25
<b>Figura 13-2.</b> Capacidad BICM del parámetro $\alpha$ .....	25
<b>Figura 14-2.</b> Constelación 64 NUC-QAM, SNR=10dB .....	28
<b>Figura 15-2.</b> Diagrama de bloques de la codificación y decodificación LDPC .....	29
<b>Figura 16-2.</b> Codificador básico de turbo código .....	34
<b>Figura 17-2.</b> Codificador RSC con $R=1/2$ .....	35
<b>Figura 18-2.</b> Dispersión de los errores, entrelazado .....	36
<b>Figura 19-2.</b> Ejemplo de la codificación Turbo.....	36
<b>Figura 20-2.</b> Decodificador turbo código .....	37
<b>Figura 1-4.</b> Desempeño 64QAM UC vs NUC.....	47
<b>Figura 2-4.</b> Desempeño 256QAM UC vs NUC.....	47
<b>Figura 3-4.</b> Desempeño 1024QAM UC vs NUC.....	48
<b>Figura 4-4.</b> Desempeño 4096QAM UC vs NUC.....	48
<b>Figura 5-4.</b> Desempeño 64QAM UC vs NUC codificación Turbo .....	50
<b>Figura 6-4.</b> Desempeño 256QAM UC vs NUC codificación Turbo .....	50
<b>Figura 7-4.</b> Desempeño 1024QAM UC vs NUC codificación Turbo .....	51
<b>Figura 8-4.</b> Desempeño 4096QAM UC vs NUC codificación Turbo .....	51
<b>Figura 9-4.</b> Codificación SOVA.....	52
<b>Figura 10-4.</b> Codificación LOG-MAP .....	53
<b>Figura 11-4.</b> Codificación LDPC .....	53
<b>Figura 12-4.</b> Codificación LDPC y Turbo, $M=64$ .....	54
<b>Figura 13-4.</b> Codificación LDPC y Turbo, $M=256$ .....	55

<b>Figura 14-4.</b> Codificación LDPC y Turbo, $M=1024$ .....	55
<b>Figura 15-4.</b> Codificación LDPC y Turbo, $M=4096$ .....	56

## ÍNDICE DE ANEXOS

**ANEXO A.** SIMULADOR DEL DESEMPEÑO DE LAS MODULACIONES NO UNIFORMES UTILIZANDO CODIFICACIÓN LDPC Y TURBO CÓDIGOS SOBRE UN CANAL AWGN CON DESVANECIMIENTO1

**ANEXO B.** PARÁMETROS DE OPTIMIZACIÓN PARA AWGN CON DEAVANECIMIENTO15

**ANEXO C.** CONSTELACIONES PARA MODULACIONES NO UNIFORMES20

**ANEXO D.** MÉTRICAS DE DESEMPEÑO22

## RESUMEN

En este trabajo se presenta un análisis de los códigos de canal LDPC y Turbo, al utilizar modulaciones digitales con constelaciones no uniformes NUC-QAM de orden  $M=64, 256, 1024$  y  $4096$ , en canales con ruido aditivo blanco Gaussiano (AWGN) con desvanecimiento plano. Para estimar el comportamiento del flujo de bits durante la codificación de canal se utilizaron dos métricas: la relación señal a ruido (SNR), y la tasa de error de bit (BER). Se consideran tres posibles escenarios para el Canal de Comunicaciones. El primero caso corresponde a un canal de comunicación sin aplicar codificación de canal, el segundo caso se aplica la codificación de canal LDPC, y el tercer caso se aplica la codificación de canal Turbo. La simulación del canal se realiza con base al modelo estadístico de distribución de probabilidad Rayleigh. Los resultados para cada caso se obtienen a través de funciones programadas en el software Matlab. Los resultados del desempeño de la codificación de canal LDPC y Turbo se comparan con los resultados obtenidos sin aplicar codificación de canal, y se determina cuál de ellos se comporta mejor con un rango amplio de valores de SNR. Finalmente se obtiene que la mejor codificación es la LDPC ya que el BER se estabiliza en el orden de  $(10^{-6})$  mucho menor comparada con la codificación Turbo.

**PALABRAS CLAVE:** MODULACIONES, CONSTELACIONES NO UNIFORMES, CODIFICACIÓN LDPC, TURBO, OPTIMIZACIÓN.

LUIS  
ALBERTO  
CAMINOS  
VARGAS

Firmado digitalmente por LUIS  
ALBERTO CAMINOS VARGAS  
Nombre de reconocimiento (DN):  
c=EC, o=UNIVERSIDAD  
serialNumber=0602766974, cn=LUIS  
ALBERTO CAMINOS VARGAS  
Fecha: 2021.11.11 08:30:18 -05'00'



0110-DBRAI-UPT-IPEC-2021

## ABSTRACT

This research it presents an analysis of LDPC and Turbo channel codes, using digital modulations with non-uniform NUC-QAM constellations of order  $M=64, 256, 1024$  and  $4096$ , in channels with additive white Gaussian noise (AWGN) with flat fading. Two metrics were used to estimate the bitstream behavior during channel coding: the signal to noise ratio (SNR), and the bit error rate (BER). Three possible scenarios are considered for the Communications Channel. The first case corresponds to a communication channel that's applied without channel coding, the second case applies LDPC channel coding, and the third case it applied Turbo channel coding. The channel simulation is performed based on the Rayleigh probability distribution statistical model. The results for each case are obtained through functions programmed in Matlab software. The performance results of LDPC and Turbo channel coding are compared with the results obtained without applying channel coding, and it is determined which of them performs better with a wide range of SNR values. Finally, it is obtained that the best coding is LDPC as the BER stabilizes in the order of  $(10^{-6})$  much lower compared to Turbo coding.

**KEY WORDS:** Modulations, Non-uniform constellations, LDPC coding, Turbo, Optimization.

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1. Antecedentes

Existe una clara tendencia hacia los sistemas digitales de comunicación. Los servicios de telefonía celular, analógicos hasta hace algunos años atrás, hoy son todos de naturaleza digital, más compactos y de mejor calidad de comunicación (Ana Lorena Velásquez; Edwin Mudarra, 2009, pág. 15).

Lo mismo sucede con muchos otros sistemas de comunicación, entre los que podemos mencionar otros servicios de telefonía, servicios de transmisión de datos, de radio digital, de distribución de contenido vía satélite, la televisión, entre otros. Para que una señal, ya sea analógica o digital, pueda transmitirse eficientemente a través de un medio o canal de comunicación, es necesario contar con algún método de modulación (Marvin K. & Mohamed-Slim, 2000, pág. 3). Las modulaciones uniformes utilizadas hasta hace un par de años atrás, ya no son suficientes para transmitir servicios de televisión de alta definición. Debido a esta clara transición hacia los sistemas digitales de comunicación de alta definición, es necesario pensar en técnicas de modulación digital, que ofrecen una mayor capacidad para acarrear grandes cantidades de información con una excelente calidad (Belkacem Mouhouche; Hakju Lee; Carlos Barjau, 2015, pág. 1). En comparación con las técnicas de modulación tradicional (constelaciones uniformes), las modulaciones no uniformes (constelaciones no uniformes) proveen transmisiones de mejor calidad, compatibilidad con servicios digitales de datos de alta definición y mayor seguridad en la transmisión de información (Belkacem Mouhouche; Hakju Lee; Carlos Barjau, 2015, pág. 1).

En consecuencia, se ha hecho necesaria la aparición de distintos sistemas de transmisión y almacenamiento, y su mejora en eficacia y eficiencia. La codificación de canal juega un papel muy importante, ya que, como demostró Shannon en 1948, con una codificación apropiada se pueden reducir los errores producidos por un canal (líneas telefónicas, radioenlaces, enlaces vía satélite) o un soporte de almacenamiento (Escabosa, 1997, pág. 1).

Para garantizar una mejor calidad de transmisión, mayor seguridad y una recepción de información más confiable, resulta necesario utilizar mecanismos de corrección de errores. Los sistemas de corrección de errores aparecieron prácticamente con el nacimiento de las telecomunicaciones (Constantino Pérez Vega, 2003, pág. 2). En un medio de transmisión, sea cual sea este, siempre se producen errores en el canal, por lo que la existencia de herramientas que detecten y, sobre todo, corrijan dichos errores es algo esencial en cualquier sistema de comunicación.

La utilización de técnicas de codificación de canal conlleva una disminución del tiempo de recepción de contenido y provoca una reducción del tráfico en la red (Lava, 2010, pág. 3).

Existen varios tipos de codificación de canal, entre los más relevantes están: la codificación *Low Density Parity Check* (LDPC), codificación convolucional y Turbo Códigos (Rao, 2015, pág. 73).

La codificación (LDPC), son códigos bloques lineales sistemáticos basados en una matriz de comprobación de paridad utilizada en el proceso de codificación y decodificación (Lava, 2010).

Turbo Códigos, es una refinación de la estructura de codificación concatenada, ya que se hace al utilizar concatenación en paralelo, más un algoritmo iterativo de decodificación; se consigue un aprovechamiento casi total de la capacidad del canal (Vucetic, 2000, pág. 11).

## **1.2. Situación problemática**

### **1.2.1. Planteamiento del problema**

Los sistemas de comunicación son un factor importante en el desarrollo tecnológico de la humanidad, debido a que la comunicación a través de distintos medios ha conformado la llegada de diferentes servicios que juegan un papel central en nuestras actividades cotidianas (Belloch, 2012, pág. 5). Los consumidores de servicios digitales demandan volúmenes de datos cada vez mayores, para ello es necesario que las tasas de transmisión alcancen un nivel adecuado que permita satisfacer dicha necesidad.

Las Modulaciones no Uniformes (NUC por sus siglas en inglés *Non-Uniform Constellation*) surgen como una de las técnicas más innovadoras para abordar esos requisitos. Estas constelaciones ya se han implementado en sistemas de transmisión de próxima generación, como DVB-NGH (*Digital Video Broadcasting - Next Generation Handheld*) o ATSC 3.0 (*Advanced Television Systems Committee - Third Generation*) (Muela, 2017, pág. 3). Los NUC reducen la brecha entre las constelaciones de modulación de amplitud en cuadratura (QAM) uniformes y el límite teórico de Shannon. Con estas constelaciones, los símbolos se optimizan tanto en componentes en fase (I) como en cuadratura (Q) mediante la conformación geométrica de la señal, considerando una determinada relación señal/ruido (SNR) y el modelo de canal (Belkacem Mouhouche; Hakju Lee; Carlos Barjau, 2015, pág. 2).

Las próximas generaciones de sistemas de difusión terrestre y la demanda de servicios digitales con muy altas tasas de datos, exigen sistemas de transmisión espectralmente más eficientes. Es necesario considerar las limitaciones que se derivan al aumentar significativamente las tasas de datos, por ejemplo, en un canal con Desvanecimiento el cual provoca que los componentes tanto

de fase como cuadratura de una Constelación determinada lleguen al receptor leve o gravemente distorsionados. Este y otros efectos provocados por el canal, se encuentran actualmente en estudio (Josefina Castañeda Camacho; Domingo Lara Rodríguez, 2001, pág. 3).

Sin embargo, en la transmisión de información a través de un canal de comunicaciones se introducen errores en la señal debido a diversos factores tales como son ruido, desvanecimiento e interferencias (Rao, 2015, pág. 16).

Debido a todo esto, resulta necesario implementar códigos de canal que puedan detectar y corregir errores para recuperar la información original, a pesar de los problemas que se generan durante su transmisión (Rao, 2015, pág. 16).

En un canal de comunicaciones la información enviada está afectada por desvanecimientos lo que no permite alcanzar el nivel de señal adecuado para obtener una relación señal/ruido suficiente para recuperar la información sin errores. Con el fin de detectar y corregir los bits erróneos se aplican a la señal transmitida una serie de transformaciones.

Debido al incremento de las necesidades de intercambio de información en todos los sectores de la sociedad, los servicios de telecomunicación han aparecido y crecido para satisfacer tan alta demanda. En consecuencia, se ha hecho necesaria la aparición de distintos sistemas de transmisión y almacenamiento, y su mejora en eficacia y eficiencia (Belloch, 2012, pág. 2).

El objetivo de la codificación de canal es reducir la probabilidad de error o bien garantizar una relación señal/ruido necesario para garantizar una cierta tasa de error. Esta herramienta consiste en transformar secuencias binarias en secuencias mejores que incluyan redundancia estructurada: los bits redundantes pueden ser utilizados para detectar y corregir errores (Constantino Pérez Vega, 2003, pág. 3).

### **1.2.2. Formulación del problema**

¿Cómo afectan las codificaciones de canal LDPC y Turbo Código al utilizar modulaciones no uniformes; en la pérdida de información en el proceso de recepción sobre canales AWGN con desvanecimiento?

### **1.3. Preguntas directrices o específicas de la investigación**

- ¿Cómo trabajan las modulaciones no uniformes al utilizar codificación de canal de tipo LDPC y Turbo Código?

- ¿Dónde la pérdida de información es mayor en el proceso de transmisión sobre canales AWGN con desvanecimiento?
- ¿Cómo influye el coeficiente de la relación señal/ruido en la recepción de la información sobre canales AWGN con desvanecimiento?
- ¿La variación de la tasa de la tasa de código, de qué manera afecta a la modulación digital y a la codificación de canal en la transmisión de información?

#### **1.4. Justificación de la investigación**

En los últimos años se ha producido un auge en la utilización de redes inalámbricas en la industria de las comunicaciones. Han surgido una gran cantidad de tecnologías inalámbricas para transmitir altas tasas de datos en redes de acceso, como Wi-Fi, UMTS, WiMAX, LTE, así como también para sistemas de *broadcast* de televisión digital en HD y UHD, como ATSC, DVB-T, ISDB-T y DTMB (Stüber, 2002, pág. 3). El incremento de las tasas de transmisión con base a la eficiencia espectral de un sistema digital se concentra en mantener el ancho de banda disponible e incrementar el orden de modulación apoyado en códigos de canal que permitan detectar y corregir errores de transmisión (Rao, 2015, pág. 16). Las tecnologías antes mencionadas han conseguido llegar hasta niveles de modulación de 64-QAM y en el caso de DVB-T en su versión 2 de 256-QAM, al utilizar rotación en su constelación (Pisciotta, 2010, pág. 8).

Existe la propuesta de modulación no uniforme que alcanzan niveles de modulación superiores a 256-QAM, bajo el criterio de optimizar la capacidad del canal, asociado a códigos de canal que le permitan aproximarse al límite de Shannon. En la actualidad estas propuestas son base de los futuros sistemas *broadcast* en ATSC 3.0 para enlaces fijos con línea de vista (Lachlan Michael; David Gómez-Barquero, 2015, pág. 4).

El gobierno nacional del Ecuador plantea en el objetivo 4: Consolidar la sostenibilidad del sistema económico social y solidario, y afianzar la dolarización, y en el objetivo 5: Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria; se destaca que para lograr los objetivos de incrementar la productividad, agregar valor, innovar y ser más competitivo, se requiere investigación e innovación para la producción, transferencia tecnológica; vinculación del sector educativo y académico con los procesos de desarrollo; pertinencia productiva y laboral de la oferta académica, junto con la profesionalización de la población; mecanismos de protección de propiedad intelectual y de la

inversión en mecanización, industrialización e infraestructura productiva (Ecuador G. N., 2017, pág. 76).

El área de las telecomunicaciones es una de las áreas más amplias para investigar, ya que, al avanzar la tecnología a pasos agigantados, las telecomunicaciones avanzan de la misma manera, además cumple con los objetivos del Plan Nacional de Desarrollo 2017-2021-Toda una Vida (Ecuador G. N., 2017, pág. 76).

El Plan Nacional de Desarrollo (2017-2021) creado por el actual gobierno ecuatoriano, propone los ejes, objetivos, políticas y metas en base a una evaluación previa de los planes anteriores, tanto en gestión como en resultados. Se ha tomado en cuenta las oportunidades y capacidades generadas para el desarrollo social y el fortalecimiento del talento humano nacional, así como la instalación de infraestructura pública en los ámbitos logístico, operativos y de telecomunicaciones, y la consolidación de una matriz energética diversificada del país (Planificación, 2017, pág. 79). El sector de infraestructura ha sido uno de los principales beneficiarios de la política pública de inversión y ha servido para crear las bases del crecimiento económico del país. Sectores como transporte (modos y medios), electricidad, agua, minería, telecomunicaciones e hidrocarburos experimentaron mejoras sustanciales, si se comparan con la infraestructura existente en los años anteriores a 2007. (Planificación, 2017, pág. 79)

Por lo mencionado anteriormente la propuesta del Plan Nacional de Desarrollo en uno de sus lineamientos dice: “Impulso a la productividad y la competitividad sistémica a partir del potenciamiento de los roles y funcionalidades del territorio”, el mismo que propone “Incrementar el acceso a servicios públicos de telecomunicaciones y tecnologías de información, especialmente en el sector rural, la frontera, Amazonía y Galápagos” (Planificación, 2017, pág. 79). Por lo tanto, para mejorar el desarrollo tecnológico de los ecuatorianos, es necesario grupos técnicos que entiendan y configuren futuros sistemas de transmisión digital que permita la incidencia total de tecnología, mantenimiento y configuración de sistemas de comunicación.

La demanda de altas tasas y eficiencia espectral para transmisión de video en sistemas inalámbricos de comunicación de datos en áreas metropolitanas podrían utilizar las modulaciones no uniformes de alto nivel para cumplir con este objetivo, pero por su posición geográfica y su particularidad actual de portabilidad y movilidad se encuentran contaminadas por la presencia de desvanecimiento en el canal, lo que incrementa la cantidad de errores en la transmisión (Muela, 2017, pág. 5). La codificación de canal juega un papel muy importante, ya que, como demostró Shannon en 1948, con una codificación apropiada se pueden reducir los errores producidos por

un canal (líneas telefónicas, radioenlaces, enlaces vía satélite) o un soporte de almacenamiento (Escabosa, 1997, pág. 1).

Al trabajar con protocolos de control de conexión como TCP/IP, las técnicas de codificación de canal conllevan a una disminución del tiempo de recepción de un contenido y provoca una reducción del tráfico en la red, ya que evitan la solicitud de paquetes perdidos (Lava, 2010, pág. 3). Este proyecto se centra en el análisis y evaluación de la codificación Turbo y Low Density Parity Check (LDPC) en apoyo a las modulaciones no uniformes en canales AWGN con desvanecimiento, en función de la capacidad del canal.

### **1.5. Objetivo general de investigación**

Estudiar el desempeño de la codificación de canal al utilizar modulaciones no uniformes sobre canales AWGN con desvanecimiento.

### **1.6. Objetivos específicos de investigación**

- Conocer el comportamiento de las codificaciones de canal LDPC y Turbo sobre modulaciones no uniformes.
- Determinar si en canales AWGN con desvanecimiento existe menor pérdida de información al utilizar codificación de canal LDPC o Turbo Código.
- Analizar la relación señal/ruido en la recepción de información al utilizar codificación de canal y modulaciones no uniformes.
- Evaluar la probabilidad de error en la transmisión de información en canales AWGN con desvanecimiento al utilizar modulaciones no uniformes y codificación de canal.

### **1.7. Hipótesis**

Mediante el estudio y la evaluación de la codificación de canal LDPC y Turbo Código, permitirán incrementar el desempeño de las modulaciones no uniformes sobre canales AWGN con desvanecimiento, con lo que se consigue disminuir la probabilidad de error en base a la optimización del canal en relación al límite de Shannon.

## CAPÍTULO II

### MARCO DE REFERENCIA

#### 2.1. Ruido AWGN

Antes de entrar en detalle en lo referente a AWGN del inglés *Additive White Gaussian Noise*, es importante conocer la palabra ruido. El ruido es una señal no deseada que afecta el comportamiento de un sistema de comunicaciones, lo que provoca pérdidas o distorsión en la información (Xiao, 2008, pág. 3).

Se llama AWGN por las siguientes razones:

- Aditivo, ya que la señal recibida es igual a la suma de la señal transmitida más el ruido.
$$r(t) = s(t) + w,$$
donde,  $r(t)$  representa la señal en el receptor,  $s(t)$  es la señal en el transmisor y  $w$  es el ruido.
- Blanco, porque representa una potencia uniforme en toda la banda de frecuencias, es decir en todo el ancho de banda.
- Gaussiana, porque representa una distribución de probabilidad normal en el dominio del tiempo, donde el valor de la media es cero y varianza  $\sigma^2$ .

#### 2.2. Distribución Rayleigh

La distribución de Rayleigh es una función de distribución continua cuya función de densidad de probabilidad es:

$$R(\alpha, \sigma_r) = \frac{\alpha}{\sigma_r^2} e^{-\frac{\alpha^2}{2\sigma_r^2}}, \alpha > 0$$

Dónde  $\alpha$  representa la resultante de las dos componentes ortogonales de la función, que se comportan como variables aleatorias continuas  $X$  e  $Y$  (Haykin, Communication Systems, 2001, pág. 56). Cada una sigue una distribución normal independiente con desviación estándar  $\sigma_r$ :

$$\alpha = \sqrt{X^2 + Y^2} \rightarrow \begin{cases} X \rightarrow N(0, \sigma_r^2) \\ Y \rightarrow N(0, \sigma_r^2) \end{cases}$$

La variable  $\alpha$  sigue una Distribución Rayleigh.

#### 2.3. Desvanecimiento Rayleigh

El desvanecimiento es la variación que sufre la señal alrededor del nivel nominal en la entrada del receptor; el desvanecimiento es una desviación temporal de la energía de la señal a un lugar

no deseado. Es decir una señal puede llegar al receptor atenuada o degradada por efectos del canal como obstrucciones causadas por irregularidades presentes en la transmisión (Marvin K. & Mohamed-Slim, 2000, pág. 101).

La causa principal del desvanecimiento Rayleigh es la recepción de señales con múltiples trayectos; es decir en el receptor llega un gran número de ondas reflejadas y dispersas.

Cuando las señales llegan al receptor, la señal total es una combinación de todas las señales que han llegado al receptor a través de los diferentes caminos disponibles. Estas señales se sumarán todas juntas, como la fase de la señal es importante; al depender de las distintas longitudes de trayecto, las señales tendrán diferentes fases lo que provoca que unas señales se sumen o se resten entre sí; esto se conoce como desvanecimiento el cual actúa de forma multiplicativa con la señal:

$$r(t) = \sum_{i=1}^n \alpha s_n(t) + n(t) = \alpha s(t) + n(t)$$

Dónde  $r(t)$  representa la señal en el receptor,  $s(t)$  la señal en el transmisor,  $\alpha$  representa el desvanecimiento y  $n(t)$  es el ruido AWGN (Marvin K. & Mohamed-Slim, 2000, pág. 101).

La probabilidad de error de símbolo puede calcularse:

$$P_e = \int_{-\infty}^{+\infty} P(x|\alpha)p(\alpha)d\alpha$$

El desvanecimiento a gran escala representa la atenuación promedio de potencia de la señal o la pérdida de ruta debido al movimiento en grandes áreas, es decir cuando la distancia supera la longitud de onda (Sklar, 1999, pág. 6).

El desvanecimiento a pequeña escala se refiere a los cambios dramáticos en la amplitud y fase de la señal que se pueden experimentar como resultado de pequeños cambios en la distancia, es decir distancias menores a la longitud de onda (Sklar, 1999, pág. 6).

Desvanecimiento plano o Frecuencia no selectiva, ocurre cuando el ancho de banda de la señal es menor al ancho de banda del canal, y la dispersión del retraso es menor al periodo de la señal.

Desvanecimiento selectivo de Frecuencia, ocurre lo opuesto que el desvanecimiento plano, el ancho de banda de la señal es mayor al ancho de banda del canal, y la dispersión del retraso es mayor al periodo de la señal (Sklar, 1999, pág. 14).

## 2.4. Relación Señal a Ruido

La SNR es probablemente la medida de rendimiento más común y mejor de un sistema de comunicación digital (Alam, 2008, pág. 3). Es un término para medir la relación de potencia entre una señal (información significativa) y el ruido de fondo:

$$SNR = \frac{S[w]}{N[w]}$$

Dónde  $S [w]$  es la potencia de la señal en vatios y  $N [w]$  es la potencia del ruido en vatios, por lo que la relación señal a ruido es una relación adimensional.

Debido a que muchas señales tienen un rango dinámico muy amplio, las SNR generalmente se expresan en términos de la escala logarítmica de decibelios (Alam, 2008, pág. 3). En decibelios, la SNR es, por definición, 10 veces el logaritmo de la relación de potencia:

$$SNR[dB] = 10 \log_{10} SNR$$

## 2.5. Tasa de Error de Bit (BER)

La BER es la medida más fundamental del rendimiento del sistema. Es decir, es una medida de qué tan bien se transfieren los bits de extremo a extremo. Si bien este rendimiento se ve afectado por factores como la señal a ruido y la distorsión, en última instancia, la capacidad de recibir información sin errores define la calidad del enlace. La tasa de errores en los bits (BER) es la cantidad de bits recibidos por error dividido para la cantidad total de bits enviados (Alam, 2008, pág. 4).

Si la BER es más alta quiere decir que en el canal de comunicación se introducen demasiados errores lo que provoca que la señal llegue totalmente distorsionada; entre más pequeña sea la BER, la señal recibida será más parecida a la enviada desde el transmisor (Alam, 2008, pág. 4).

## 2.6. Capacidad CM

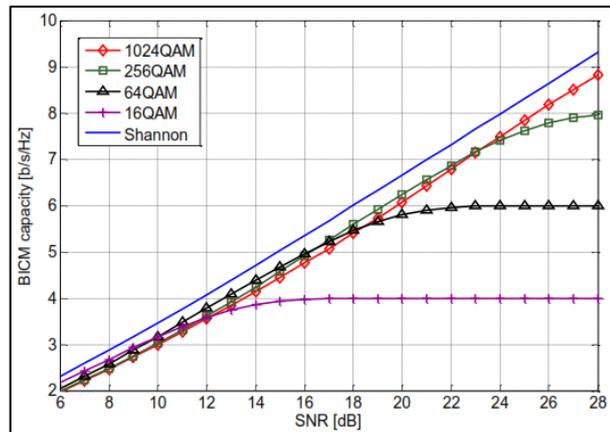
CM, viene del inglés *Coded Modulation*, que en español significa Modulación Codificada

## 2.7. Capacidad BICM

BICM, viene del inglés *Bit Interleaved Coded Modulation*, que en español quiere decir modulación codificada de bits entrelazados.

## 2.8. Constelaciones no uniformes

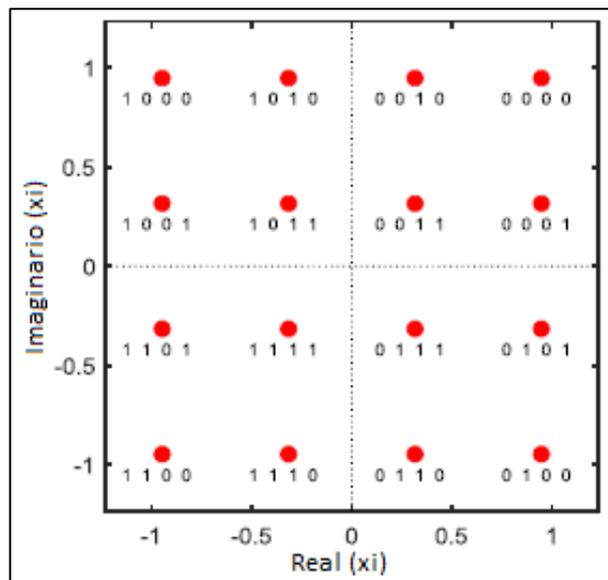
Se conoce que las constelaciones uniformes de QAM se aproximan muy estrechamente al límite de Shannon sin restricciones para la SNR baja, pero la separación se hace más evidente para relaciones de señal-ruido más altas (SNR), como se observa en la figura 1.2. (Zoellner & Loghin, 2013, pág. 1) .



**Figura 1-2.** Capacidad BICM , UC vs límite de Shannon

Fuente: (Zoellner & Loghin, 2013, pág. 3)

Con constelaciones de Modulación de amplitud en cuadratura (QAM) uniformes, los símbolos se espacian regularmente en el diagrama de constelación, es decir, con la misma distancia entre símbolos tanto en componentes en fase (I) como en cuadratura (Q) (Joan, David, & Manuel, 2016, pág. 1).

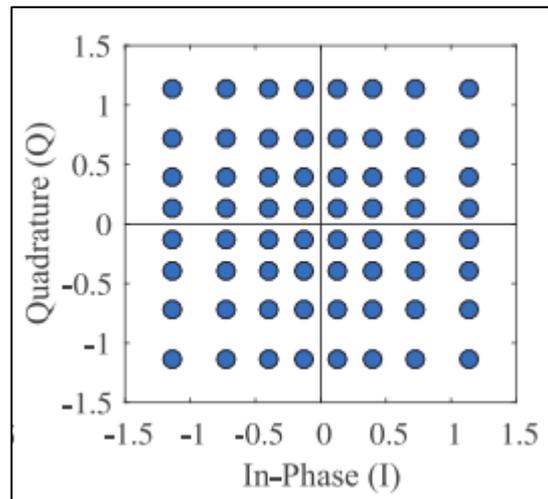


**Figura 2-2.** Constelación 16-QAM uniforme

Fuente: (Zoellner & Loghin, 2013, pág. 3)

La distribución de los símbolos de la constelación convencional muchas de las veces no es óptima, debido a las dos restricciones, la forma rectangular y niveles igualmente espaciados.

Para las constelaciones QAM de alto orden desde 64 QAM en adelante, se utilizan técnicas de conformación de constelaciones, es decir que las distancias entre símbolos son de diferente tamaño mismas que dependen del orden de la modulación, lo que beneficia para cerrar esta brecha, y da como resultado constelaciones con probabilidad de símbolo desigual o constelaciones no uniformes (NUC) (Zoellner & Loghin, 2013, pág. 4).



**Figura 3-2.** Constelación 64 NUC-QAM

Fuente: (Muela, 2017, pág. 68)

Las constelaciones no uniformes se han introducido en sistemas de televisión digital terrestre (DTT) de nueva generación para reducir la brecha entre la capacidad BICM (*Bit Interleaved Coding and Modulation*) de constelaciones uniformes y el límite Shannon. Además, proporcionan un mayor rendimiento en comparación con las constelaciones uniformes.

## 2.9. Codificación de canal

Las técnicas de codificación y entrelazado de canales se han reconocido durante mucho tiempo como una técnica eficaz para combatir los efectos nocivos del ruido, la interferencia, el desvanecimiento y otras degradaciones introducidas por el canal de transmisión (Cándido López García; Manuel Fernández Veiga, 2002, pág. 414).

El objetivo principal de la codificación de canal es introducir una redundancia en los datos a transmitir, para que en el receptor se detecte esta redundancia y se pueda corregir los errores inducidos por el canal por medio de la corrección de errores hacia adelante (Stüber, 2002, pág. 410). De esta manera se obtiene, una información lo más parecida a la información original, es decir con el menor número de errores existentes.

Las ventajas más relevantes de utilizar codificación de canal pueden ser:

- Operar con transmisión de baja potencia
- Transmite a largas distancias y altas tasas de datos
- Presenta mayor tolerancia a la interferencia

Lo que se desea cuando se realiza una transmisión de información es que esta experimente el menor número de cambios o errores, y que los bits que llegan al destino sean lo más parecido posible a los originados en la fuente (Cándido López García; Manuel Fernández Veiga, 2002, pág. 411).

Con la implementación de la codificación de canal actualmente es posible diseñar sistemas de comunicación más confiables y más rápidos.

Existen dos métodos de codificación de canal, Corrección de error hacia atrás BEC o solicitud de repetición automática (ARQ) y Corrección de error de avance (FEC) (Haykin, Communication Systems, 2001, pág. 626).

➤ **BEC (Backward error correction) o ARQ (automatic-repeat request)**

La filosofía subyacente de ARQ es bastante diferente de la de FEC. ARQ usa redundancia con el propósito de detectar errores. Tras la detección de un error en una palabra de código transmitida, el receptor solicita una retransmisión de la palabra de código dañada, que necesita el uso de una ruta de retorno (es decir, un canal de realimentación) (Haykin, Communication Systems, 2001, pág. 626).

ARQ puede usarse solo en enlaces *half-duplex* o *full-duplex*. En un enlace *half-duplex*, la transmisión de datos a través del enlace puede realizarse en ambas direcciones pero no simultáneamente; mientras que en un enlace *full-duplex* es posible que la transmisión de datos proceda sobre el enlace en ambas direcciones simultáneamente (Haykin, Communication Systems, 2001, pág. 627).

➤ **FEC (*Forward error correction*)**

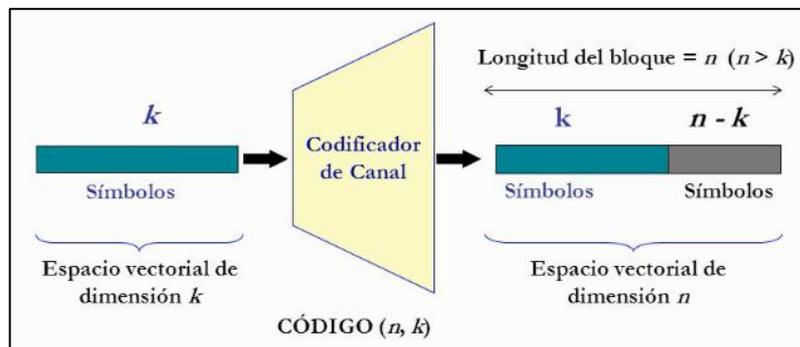
Se basa en el uso controlado de la redundancia en la palabra de código transmitida tanto para la detección como para la corrección del error introducidos durante la transmisión sobre un canal con ruido. Independientemente de si la decodificación de la palabra de código recibida es exitosa, no se realiza ningún procesamiento adicional en el receptor. En consecuencia, las técnicas de codificación de canal adecuadas para FEC requieren solo un enlace de una vía entre el transmisor y el receptor; sin la necesidad de retransmitir la información (Cándido López García; Manuel Fernández Veiga, 2002, pág. 257).

Los códigos de bloque lineal y códigos convolucionales son códigos correctores de errores de avance (FEC), los mismos que se detallan a continuación. Además, los turbo códigos y los códigos LDPC se detallan en el siguiente capítulo, ya que estos son el objetivo de estudio y análisis del presente proyecto.

### 2.9.1. Códigos de bloque lineal

Se dice que un código es lineal si se pueden agregar a los bits de código bits de redundancia para producir una tercera palabra de código.

Considérese entonces un código de bloque lineal  $(n, k)$ , en el que  $k$  bits de los  $n$  bits de código son siempre idénticos a la secuencia de mensaje a transmitir. Los  $n-k$  bits en la parte restante se calculan a partir de los bits de mensajes, estos se denominan bits de control de paridad o bits de paridad (Cándido López García; Manuel Fernández Veiga, 2002, pág. 434).



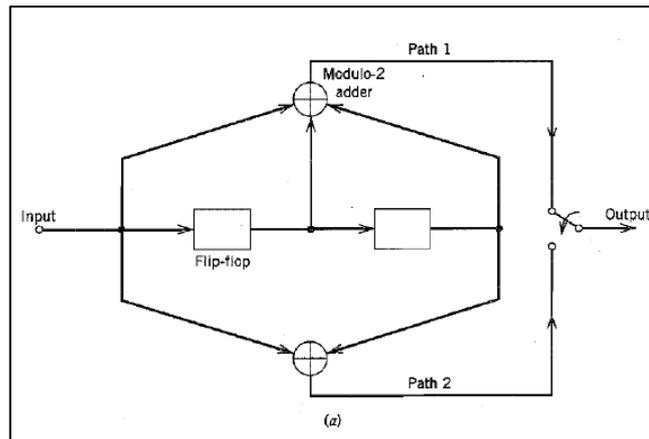
**Figura 4-2.** Codificación de bloque lineal

Fuente: (Belen, 2014, pág. 39)

### 2.9.2. Códigos Convolucionales

Los códigos convolucionales creados como una alternativa a los códigos de bloque. Los códigos convolucionales difieren de los códigos de bloque en que el codificador contiene memoria, y la salida del codificador no solo depende de las entradas, sino también de algún número de entradas previas (Constantino Pérez Vega, 2003, pág. 7).

La codificación convolucional opera sobre la secuencia de bits a la entrada del codificador o en bloques pequeños de bits. En los códigos convolucionales intervienen tres parámetros,  $(n, k, m)$ ; donde  $n$  es el número de bits a la salida del codificador,  $k$  el número de bits de información a la entrada de éste y  $m$ , el número de registros de memoria; la tasa de código se obtiene de la relación  $k/n$  (Cándido López García; Manuel Fernández Veiga, 2002, pág. 413).



**Figura 5-2.** Codificador convolucional de tasa  $\frac{1}{2}$

Fuente: (Haykin, *Communication Systems*, 2001)

### 2.9.3. Codificación LDPC (*Low Density Parity Check*)

Los códigos de verificación de paridad de baja densidad son códigos especificados por una matriz que contiene principalmente ceros (0) y solo un pequeño número de unos (1).

#### 2.9.3.1. Corrección de errores utilizando chequeo de paridad

Shannon demostró que es posible transmitir información de manera arbitrariamente confiable sobre cualquier canal no confiable, siempre que la velocidad de transmisión de información sea menor que la capacidad del canal. Por lo tanto, la capacidad del canal establece el límite de la cantidad de información que podemos transmitir a través de un canal (Rao, 2015, pág. 76).

La afirmación de Shannon puede realizarse mediante una técnica denominada corrección de errores directa. La idea básica es incorporar bits redundantes o bits de verificación, esto se conoce como palabra de código. Si los bits de control se introducen de una "manera apropiada" para hacer que cada palabra de código sea suficientemente distinta entre sí, el receptor será capaz de determinar la palabra de código más probable que se ha transmitido. La capacidad del canal determinará la cantidad exacta de redundancia que debe ser incorporada por el codificador para poder corregir los errores impuestos por el canal (Rao, 2015, pág. 76).

El esquema de codificación más simple posible es el código de verificación de paridad único (SPC). El SPC implica la adición de un único bit al mensaje, cuyo valor depende de los bits del mensaje. En un código de paridad par, el bit adicional agregado a cada mensaje asegura un número par de 1s en cada palabra de código; es decir si el número de unos es par el bit de paridad es 0, caso contrario es 1 (Rao, 2015, pág. 75).

Por ejemplo: para la letra *S*, que en binario es 1010011 a la que se le agregará un bit de paridad como el octavo bit. La cadena para *S* como ya tiene un número par de unos, por lo tanto, el bit de

paridad es un 0, y la palabra de código para  $S$  es 10100110. Más formalmente, para el código ASCII de 7 bits, definimos una palabra código  $c$  que tiene la siguiente estructura:

$$c = [c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8],$$

Dónde cada  $c_i$  es un 0 o un 1, y cada palabra de código satisface la restricción

$$c_1 \oplus c_2 \oplus c_3 \oplus c_4 \oplus c_5 \oplus c_6 \oplus c_7 \oplus c_8 = 0 \quad (1)$$

A la ecuación (1) se la denomina ecuación de verificación de paridad, en la que el símbolo  $\oplus$  representa la suma del módulo 2.

Para fines de codificación, entonces las ecuaciones representadas por las matrices de chequeo siempre se pueden resolver para dar los dígitos de verificación como sumas explícitas de dígitos de información.

### 2.9.3.2. Representación Gráfica

La matriz de verificación de paridad  $H$  puede ser representada gráficamente mediante los llamados “Grafo de Tanner”, estos ayudan a entender de una mejor manera el algoritmo de decodificación.

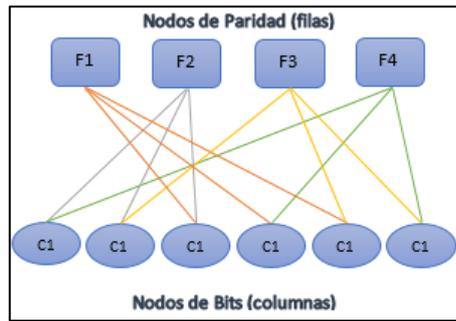
En el grafo de Tanner se identifican los nodos de bits (variables) y los nodos de paridad (verificación), los que están en función de los valores de la matriz de verificación de paridad. Los nodos de bits representan el número de columnas de  $H$ , mientras que los nodos de paridad representan el número de filas de  $H$ . En la figura (6.2) se observa un ejemplo de matriz  $H$  con su respectivo grafo de Tanner en la figura (7.2) (Belen, 2014, pág. 70).

$$H = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

**Figura 6-2.** Ejemplo Matriz de Verificación de Paridad  $H$  (4x6)

Fuente: (Belen, 2014, pág. 70)

En la figura anterior se observa la matriz  $H$  de dimensiones (4x6), en la que 4 representan el número de filas que se denotan por  $M$ , y el 6 representan el número de columnas que se denota por  $N$ .

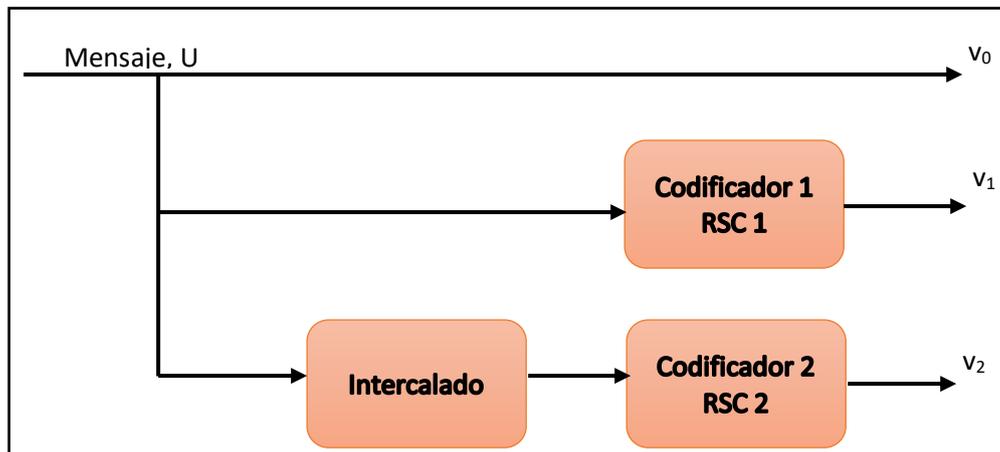


**Figura 7-2.** Representación de la matriz H con Grafo de Tanner

Fuente: (Belen, 2014, pág. 72)

#### 2.9.4. Codificación turbo

Turbo códigos se pueden codificar de diferentes maneras, pero se puede resumir esta codificación como se muestra en la figura 8-2.



**Figura 8-2.** Codificación Turbo

Fuente: Realizado por Mónica F.

Aquí el esquema comienza con la información, o mensaje (llamado U) que se enviará. U se envía a través del codificador 1 y U también está intercalado, que a menudo se denota por  $\pi$ , y el resultado es, U', se envía a través del codificador 2. El codificador 1 y el codificador 2 son normalmente idénticos, la razón por la que su salida es diferente depende totalmente del entrelazado. La información, u o  $v_0$ , es junto con el resultado del codificador 1,  $v_1$ , y el resultado del codificador 2,  $v_2$ , concatenados. Por consiguiente algunos codificadores perforan el resultado y la palabra de código puede entonces ser enviado a través de un canal (Vucetic, 2000, pág. 10).

Como se mencionó anteriormente, los turbo códigos se pueden codificar de diferentes maneras.

### 2.9.4.1. Código de producto

Los códigos de producto Turbo (TPC) normalmente se crean a partir de bloques de palabras de códigos más pequeños.

Un TPC codificará un bloque de palabras de código, al calcular los bits de paridad fila por fila y al añadir bits de paridad a cada fila, también llamada codificación horizontal. Luego, los bits de paridad se calculan columna por columna y luego se anexan a cada columna, que se denomina codificación vertical. Estos dos bloques de bits de paridad se utilizan luego como entrada para la paridad en la paridad, es decir, bits de paridad que dependen de bits de paridad vertical y horizontal. Un pequeño ejemplo se da en la figura (9-2), donde se puede observar los bits de información y los de paridad (Utby, 2006, pág. 42).



**Figura 9-2.** Código de producto (TPC)

**Fuente:** Realizado por Mónica F.

### 2.9.4.2. Códigos convolucionales

Normalmente, en los códigos turbo, se utilizan dos códigos recursivos de convolución sistemática (RSC) (Vucetic, 2000, pág. 11). Estos dos códigos se concatenan y luego se envían a través de un canal.

Una observación importante en RSC es que, dado que son recursivas, el codificador no volverá al estado de cero al añadir una cola de cero. En cambio, el contexto actual del codificador debe adjuntarse en la cola.

## 2.10. Límite de Shannon

El objetivo de un sistema de comunicación es llevar una señal desde una fuente de interés a través de un canal ruidoso a un usuario en el otro extremo del canal. Este mensaje se entrega al usuario de manera eficiente y confiable (Utby, 2006, pág. 41).

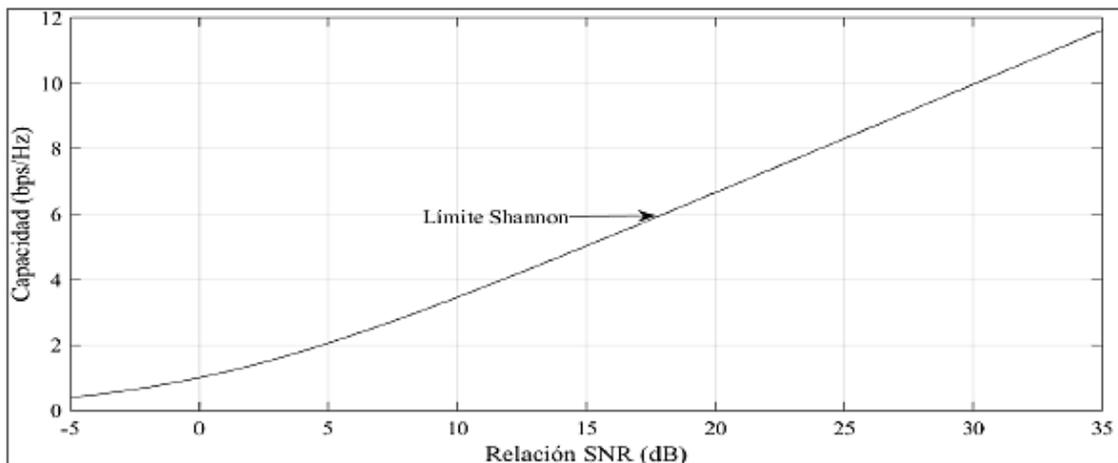
En un sistema de comunicación digital, la confiabilidad se expresa generalmente en términos de tasa de error de bit (BER) o probabilidad de error de bit, medidos a la salida del receptor. Cuanta más pequeña es la BER, más confiable es el sistema de comunicación. Entonces, ¿es posible diseñar un sistema de comunicación que opere con cero BER incluso a través de un canal con ruido? (Haykin, Communication systems, 2001, pág. 20).

La respuesta a esta interrogante es el conocido teorema de Shannon, que se llama el teorema de la capacidad de información; el cual describe una ecuación para calcular este límite,

$$C = B \log_2(1 + SNR) \quad [b/s]$$

Dónde  $B$  denota el ancho de banda del canal,  $SNR$  denota la relación señal a ruido recibida,  $C$  es la capacidad de información del canal.

Por lo antes expuesto la capacidad de información del canal se define como la velocidad máxima a la que la información puede transmitirse a través del canal, sin errores; se mide en bits por segundo (b/s).



**Figura 10-2.** Límite de Shannon

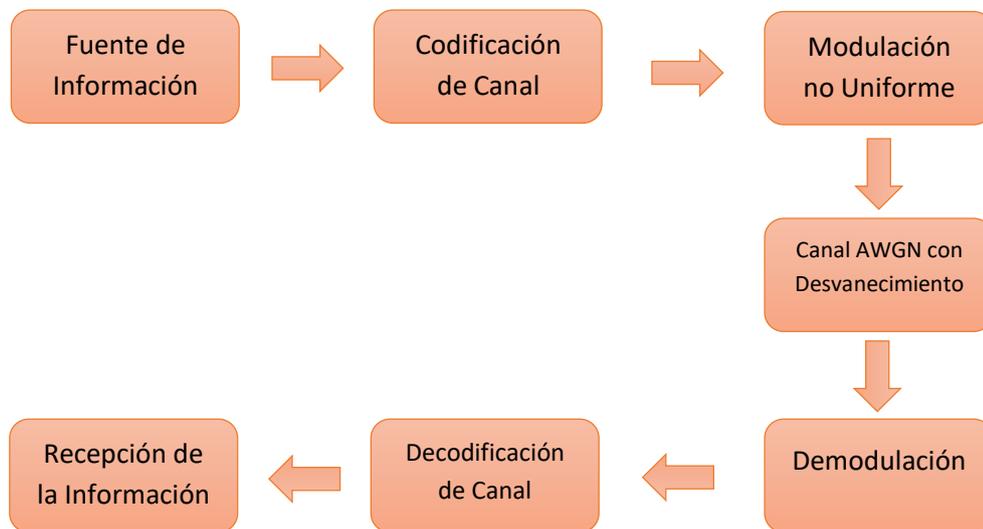
**Fuente:** (Cándido López García; Manuel Fernández Veiga, 2002, pág. 21)

El teorema de capacidad de información nos dice que un mensaje puede transmitirse a través de un sistema de comunicación sin errores, incluso cuando el canal es ruidoso, siempre y cuando la velocidad real de señalización en bits por segundo, a la que se transmiten los datos a través del

canal, sea menor que la capacidad de información  $C$ . Sin embargo, es necesario considerar procesos adicionales que sufre la información para adaptarse al canal de comunicaciones, que introducen retardos y errores significativos, lo que dificulta el acercarse al límite de Shannon (Cándido López García; Manuel Fernández Veiga, 2002, pág. 21).

## 2.11. Modelo del Sistema de Comunicaciones

En un sistema de comunicación existen varias etapas que realizan diferentes tareas como recolección de la información, codificación, digitalización y conversiones para adaptarlas al canal de comunicación; luego, en el receptor se realizan procesos inversos para poder recuperar la información original (Avíles, 2017, pág. 12).



**Figura 11-2.** Diagrama de bloques del Sistema de Comunicaciones

**Fuente:** Realizado por Mónica F.

Se puede observar en la figura 11-2 un diagrama de bloques con las etapas que tiene un sistema de comunicación para el análisis de este proyecto; los mismos que se describen a continuación.

### 2.11.1. Fuente de Información

Corresponden a los datos que se van a transmitir, pueden ser: un archivo de texto, de audio, de video, o una cadena de caracteres. En el presente proyecto la fuente de información está representada por una cadena aleatoria de bits.

### **2.11.2. Codificación de Canal**

La codificación de canal introduce redundancia a los datos, con esto se protege la información de errores que puedan introducirse en la transmisión. Para el presente proyecto se utilizan las codificaciones de canal LDPC y Turbo Códigos.

### **2.11.3. Modulación No Uniforme**

En la etapa de modulación se mapea los bits de información sobre constelaciones no uniformes para poder ser transmitidos por el canal de comunicación.

En esta etapa se utilizan como ya se explicó anteriormente las modulaciones no uniformes que actualmente están en investigación.

### **2.11.4. Canal AWGN con Desvanecimiento**

Esta etapa representa el canal de comunicación, al mismo que se le añade ruido gaussiano y desvanecimiento, para simular un canal de comunicación real.

### **2.11.5. Demodulación**

La demodulación consiste en mapear la información recibida en el receptor; es decir se realiza el proceso inverso al de la modulación.

### **2.11.6. Decodificador de Canal**

En la decodificación de canal se utiliza la redundancia introducida en el transmisor para corregir el mayor número de errores que tiene la información después de ser demodulada.

Al igual que en la etapa de codificación se utilizan algoritmos de decodificación para los códigos LDPC y Turbo.

### **2.11.7. Recepción de la Información**

Es quien recibe la información, sea cual sea el mensaje que se envió. Representa la recuperación del mensaje enviado desde el transmisor al pasar por el canal de comunicación y llegar al receptor. En esta etapa se realiza una comparación entre la señal original que salió desde el transmisor y la señal que llegó al receptor; para cuantizar el número de errores que introdujo el canal de comunicación.

De esta manera poder concluir si el canal de comunicación resulta óptimo o no para la transmisión de información.

## 2.12. Codificación LDPC, turbo SOVA y LOG-MAP utilizando modulaciones no uniformes

### 2.12.1. Límite CM para Constelaciones uniformes

CM, viene del inglés *Coded Modulation*, que en español significa Modulación Codificada, Se supone que se transmite un grupo de  $M$  símbolos, cada uno con su respectiva ubicación  $X = x_i$ , de una constelación determinada, donde  $i = 1, 2, \dots, M$ . Estos símbolos pasan por un Canal AWGN que los distorsiona introduciéndoles ruido, estos lleguen al receptor, en forma de una señal continua  $Y$  (Stott, 2013, pág. 8). Entonces, el cálculo de la Información mutua  $I(X, Y)$ , entre la señal enviada y la señal recibida, debe escribirse de la siguiente forma:

$$I(X; Y) = \int_Y \sum_{i=1}^M p(x_i, y) \log_2 \frac{p(x_i, y)}{P(x_i)p(y)} dy \quad (2.1)$$

Normalmente supondríamos que todas las  $M$  posiciones de constelación ocurren con igual probabilidad, entonces  $p(x_i) = 1/M$ , se tiene que la probabilidad condicional es igual a:

$$p(x_i, y) = \frac{p(y|x_i)}{M} \quad (2.2)$$

Por lo tanto, para encontrar la capacidad CM se reemplaza la ecuación (2.2) en la ecuación (2.1) y se tiene:

$$I(X; Y) = \int_Y \sum_{i=1}^M p(x_i, y) (\log_2 p(x_i, y) - \log_2 P(x_i)p(y)) dy$$

$$I(X; Y) = \int_Y \sum_{i=1}^M \frac{p(y|x_i)}{M} \left( \log_2 p(y|x_i) - \log_2 \sum_{k=1}^M \frac{p(y|x_k)}{M} \right) dy \quad (2.3)$$

La Capacidad CM encontrada en base a la ecuación (2.3), es en unidades de bit/símbolo. Los símbolos se transmiten con la tasa de Nyquist ( $F_s = 2F_{max}$ ), por lo tanto este resultado es numéricamente igual a la Capacidad normalizada a bit/s/Hz (Stott, 2013, pág. 10).

### 2.12.2. Límite BICM para constelaciones uniformes

BICM, viene del inglés *Bit Interleaved Coded Modulation*, que en español quiere decir modulación codificada de bits entrelazados, a diferencia de la capacidad CM el análisis se lo realiza a nivel de bit (Stott, 2013, pág. 13). Al igual que en la capacidad CM, se supone la transmisión de un grupo de  $M$  símbolos asociados a un grupo de ubicaciones  $X = x_i$ , que pasan por un canal

AWGN. Para calcular la Capacidad BICM también se recurre a la ecuación de la Información Mutua, pero ahora aplicándola bit a bit, por separado. Luego se suman todos los resultados y así se obtiene la Capacidad total BICM.

El cálculo de la Capacidad BICM puede aplicarse a un flujo de bits entrelazados, codificados y luego mapeados (Stott, 2013, pág. 13).

Si se requiere calcular la Capacidad BICM para un grupo de símbolos mapeados con una Constelación M-QAM, se extrae uno de sus ejes PAM, el cual contiene únicamente  $M_{PAM} = \sqrt{M}$  símbolos, cada uno con  $k = \log_2 M_{PAM}$  niveles de bit. Estos niveles van en orden descendente, desde el más significativo (MSB), hasta el menos significativo (LSB). En cada nivel de bit se puede diferenciar dos conjuntos de símbolos (0 y 1), ambos con  $m = M_{PAM}/2$  elementos:

Un conjunto  $x_0 = x_k$ , con las ubicaciones que contienen un cero, donde  $k = 1, 2, \dots, m$ .

Un conjunto  $x_1 = x_q$ , con las ubicaciones que contienen un uno, donde  $q = m + 1, m + 2, \dots, M_{PAM}$ .

En base a la ecuación (3-1), la Capacidad BICM puede calcularse así:

Basándose en la ecuación de la información mutua (2.1), la capacidad BICM puede expresarse así:

$$C_{BICM} = \int_Y p(x_0, y) \log_2 \frac{p(x_0, y)}{p(x_0)p(y)} dy + \int_Y p(x_1, y) \log_2 \frac{p(x_1, y)}{p(x_1)p(y)} dy \quad (2.4)$$

Al suponer que se transmite el mismo número de unos y ceros, de modo que  $p(x_0) = p(x_1) = 1/2$ , entonces la probabilidad condicional del cero y uno son:

$$p(x_0, y) = p(y|x_0)P(x_0)$$

$$p(x_0, y) = \frac{p(y|x_0)}{2} \quad (2.5)$$

$$p(x_1, y) = p(y|x_1)P(x_1)$$

$$p(x_1, y) = \frac{p(y|x_1)}{2} \quad (2.6)$$

Al reemplazar en la ecuación (2.4), se tiene:

$$\begin{aligned}
C_{BICM} &= \int_{-\infty}^{+\infty} \frac{p(y|x_0)}{2} \log_2 \frac{\frac{p(y|x_0)}{2}}{\frac{1}{2}p(y)} dy + \int_{-\infty}^{+\infty} \frac{p(y|x_1)}{2} \log_2 \frac{\frac{p(y|x_1)}{2}}{\frac{1}{2}p(y)} dy \\
C_{BICM} &= \int_{-\infty}^{+\infty} \frac{p(y|x_0)}{2} \log_2 \frac{p(y|x_0)}{p(y)} dy + \int_{-\infty}^{+\infty} \frac{p(y|x_1)}{2} \log_2 \frac{p(y|x_1)}{p(y)} dy \\
C_{BICM} &= \int_{-\infty}^{+\infty} \left( \frac{p(y|x_0)}{2} \log_2 \frac{p(y|x_0)}{p(y)} + \frac{p(y|x_1)}{2} \log_2 \frac{p(y|x_1)}{p(y)} \right) dy \\
C_{BICM} &= \int_{-\infty}^{+\infty} \left[ \frac{p(y|x_0)}{2} (\log_2 p(y|x_0) - \log_2 p(y)) + \frac{p(y|x_1)}{2} (\log_2 p(y|x_1) - \log_2 p(y)) \right] dy \\
C_{BICM} &= \int_{-\infty}^{+\infty} \left[ \frac{p(y|x_0) \log_2 p(y|x_0) + p(y|x_1) \log_2 p(y|x_1)}{2} \right. \\
&\quad \left. - \left( \frac{p(y|x_0)}{2} + \frac{p(y|x_1)}{2} \right) \log_2 p(y) \right] dy \quad (2.7)
\end{aligned}$$

Si se sabe que:

$$p(y) = \left( \frac{p(y|x_0)}{2} + \frac{p(y|x_1)}{2} \right) \quad (2.8)$$

Entonces, si reemplaza (2.8) en (2.7), se tiene:

$$\begin{aligned}
C_{BICM} &= \int_{-\infty}^{+\infty} \left[ \frac{p(y|x_0) \log_2 p(y|x_0) + p(y|x_1) \log_2 p(y|x_1)}{2} \right. \\
&\quad \left. - p(y) \log_2 p(y) \right] dy \quad (2.9)
\end{aligned}$$

Ahora, si se asume que los otros bits de la constelación también son equiprobables, puede expresarse como:

$$p(y|x_0) = \frac{\sum_{k=1}^m p(y|x_k)}{m} \quad (2.10)$$

$$p(y|x_1) = \frac{\sum_{q=m+1}^{M_{PAM}} p(y|x_q)}{m} \quad (2.11)$$

En base a la ecuación (2.10),  $p(y)$  puede expresarse como:

$$p(y) = \left( \frac{\sum_{k=1}^m p(y|x_k)}{2} + \frac{\sum_{q=m+1}^{M_{PAM}} p(y|x_q)}{2} \right)$$

$$p(y) = \left( \frac{\sum_{k=1}^m p(y|x_k)}{2m} + \frac{\sum_{q=m+1}^{M_{PAM}} p(y|x_q)}{2m} \right) \quad (2.12)$$

Ahora, al suponer que el canal AWGN añade ruido con una varianza  $\sigma^2$ , a cada eje, las ecuaciones (2.10) y (2.11) pueden expresarse, así:

$$p(y|x_0) = \frac{\sum_{k=1}^m \frac{e^{-\frac{(y-x_k)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}}{m} \quad (2.13)$$

$$p(y|x_1) = \frac{\sum_{q=m+1}^{M_{PAM}} \frac{e^{-\frac{(y-x_q)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}}}{m} \quad (2.14)$$

El valor de la varianza, se puede calcular en función de la relación Señal-Ruido:

$$\frac{E_b}{N_0} = SNR(dB) - 10 \log_{10} 2$$

$$\sigma = \sqrt{\frac{\overline{E_s}}{2 \frac{\overline{E_s}}{N_0}}}$$

$$\sigma = \sqrt{\frac{\overline{E_s}}{2k \frac{E_b}{N_0}}} \quad (2.15)$$

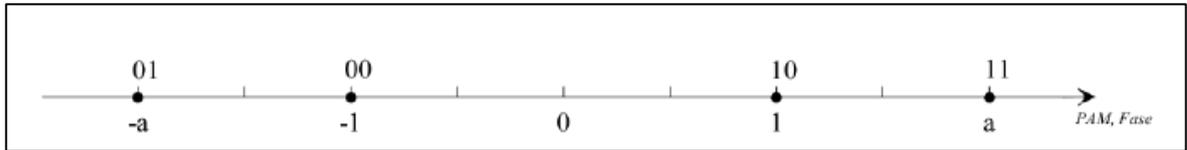
Una vez que se tiene la Capacidad BICM de un eje PAM, solo hay que duplicar su valor, para así conseguir la Capacidad BICM total de la Constelación M-QAM.

### 2.12.3. Constelaciones QAM no uniformes (NUC-QAM)

En las Constelaciones QAM uniformes, los símbolos están distribuidos uniformemente en el plano; esto se debe a que la distancia entre un símbolo y otros más cercanos se mantiene constante en cualquier punto de la Constelación. Esto no sucede con las constelaciones no uniformes, ya

que si se calcula la capacidad BICM a cada posición de la constelación y la que presente el mayor valor será la más óptima (Avíles, 2017, pág. 56).

Por ejemplo, si se toma un eje PAM de 16 QAM se tiene el parámetro de  $a$

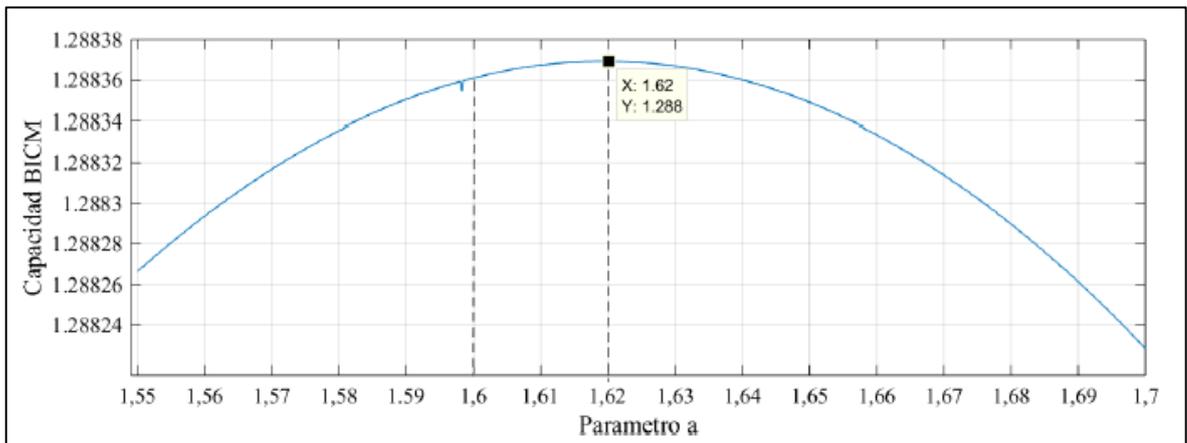


**Figura 12-2.** Eje PAM 16 QAM

Fuente: (Avíles, 2017, pág. 57)

Como se observa en la figura anterior, como la modulación es 16 QAM se tiene solo un parámetro para evaluar que es  $a$ , por lo que se debería evaluar  $a$  entre un amplio rango de valores, por lo general la solución más óptima es tomar valores que se encuentren relativamente cerca a los convencionales.

Por lo que si  $a = 3$  en la modulación uniforme 16 QAM;  $a$  debería evaluarse entre  $1 \leq a \leq 3.05$ , con pasos muy pequeños (0,01). Al calcular la capacidad BICM se debe elegir el valor de  $a$  que tenga mayor capacidad, como muestra la figura 13-2.



**Figura 13-2.** Capacidad BICM del parámetro  $a$

Fuente: (Muela, 2017)

Como se observa en la figura anterior, se tiene que el valor más óptimo es  $a = 1.62$ ; además se puede decir que existe un solo punto óptimo donde la capacidad es máxima.

Hay que tener en cuenta que el número parámetros a optimizar depende del orden de modulación, es decir:

Para 64 NUC-QAM:  $X = [-d3, -d2, -d1, -1, 1, d1, d2, d3]$

Para 256 NUC-QAM:  $X = [-d7, -d6, -d5, \dots, -d2, -d1, -1, 1, d1, d2, \dots, d5, d6, d7]$

Para 1024 NUC -QAM:  $X = [-d15, -d14, \dots, -d3, -d2, -d1, -1, 1, d1, d2, d3, \dots, d14, d15]$

Para 4096 NUC -QAM:  $X = [-d31, -d30, \dots, -d3, -d2, -d1, -1, 1, d1, d2, d3, \dots, d30, d31]$

Entre más alta sea el orden de la modulación, más parámetros se tienen que optimizar, lo que conlleva a utilizar bastante tiempo para encontrar los puntos de constelación adecuados.

#### 2.12.4. Optimización para desvanecimiento

Hasta el momento se ha analizado el canal de comunicación solo con ruido AWGN, pero ¿cómo se comporta el canal si se añade desvanecimiento?, ¿cómo optimizar las constelaciones?

La respuesta a estas preguntas se encuentra en la ecuación de la probabilidad de error condicionada para un **canal AWGN con desvanecimiento**, por lo tanto las probabilidades de  $p(y|x_0)$  y  $p(y|x_1)$ , están dadas por (Avíles, 2017, pág. 65):

$$p(y|x_0) = \frac{\sum_{k=1}^m \frac{\alpha e^{-\frac{(y-x_k)^2}{2\sigma^2}} \frac{\alpha^2}{e^{2\sigma_r^2}}}{\sigma \sigma_r^2 \sqrt{2\pi}}}{m} \quad (2.16)$$

$$p(y|x_1) = \frac{\sum_{q=m+1}^{M_{PAM}} \frac{e^{-\frac{(y-x_q)^2}{2\sigma^2}} \frac{\alpha^2}{e^{2\sigma_r^2}}}{\sigma \sigma_r^2 \sqrt{2\pi}}}{m} \quad (2.17)$$

Dónde  $\sigma = \sigma_r = \sqrt{1/2}$ , por lo tanto el cálculo de la capacidad BICM queda de la siguiente manera:

$$C_{BICM} = \int_0^\infty \int_{-\infty}^{+\infty} \left[ \frac{p(y|x_0) \log_2 p(y|x_0) + p(y|x_1) \log_2 p(y|x_1)}{2} - p(y) \log_2 p(y) \right] dy \quad (2.13)$$

En la tabla 3-1 se muestran las ubicaciones optimizadas para 64 NUC-QAM, para distintos valores de SNR sobre un canal AWGN con desvanecimiento.

Se debe tener en cuenta que en la tabla 1-2 se encuentran solo las distancias positivas de un eje PAM, para transformarla en QAM, las mismas distancias se multiplican por -1, y además se las ubica en cuadratura y fase.

**Tabla 1-2.** Ubicaciones optimizadas para canales AWGN con Desvanecimiento 64 NUC-QAM

64 NUC-QAM			
SNR(dB)	AWGN Desvanecimiento		
	<i>d1</i>	<i>d2</i>	<i>d3</i>
7	1,2092	3,0799	4,4154
8	1,3451	3,298	4,9297
9	1,5409	3,5826	5,5069
10	1,8112	3,9386	6,1594
11	2,1208	4,3237	6,8108
12	2,3945	4,6577	7,3475
13	2,6067	4,9074	7,7177
14	2,756	5,0773	7,9488
15	2,8505	5,1674	8,0398
16	2,912	5,2201	8,068
17	2,9496	5,2393	8,0538
18	2,9751	5,2491	8,0217
19	2,9907	5,2493	7,9849
20	3,0032	5,2489	7,9528
21	3,0055	5,2365	7,9035
22	3,0126	5,2375	7,8862
23	3,0124	5,2247	7,8443
24	3,0136	5,2182	7,8194

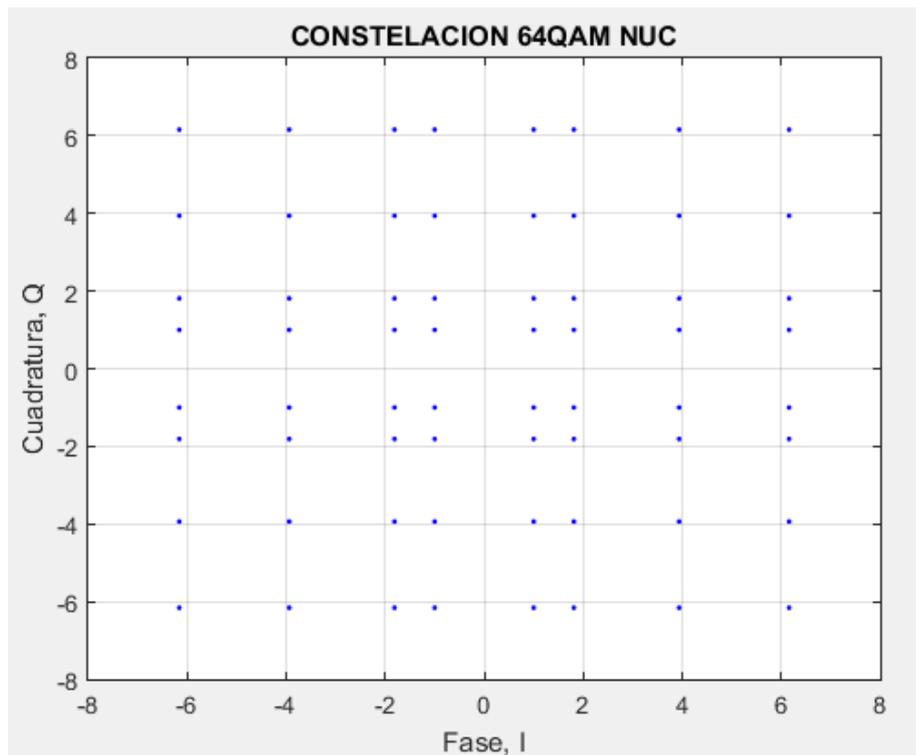
Fuente: Realizado por Mónica F.

En base a estas distancias se realiza la gráfica de la constelación no uniforme, la misma que es diferente para cada una de las SNR.

Las ubicaciones optimizadas para las modulaciones 256, 1024 y 4086 NUC-QAM se encuentran en el Anexo B.

En la figura 2.3 se observa la constelación no uniforme correspondiente a 64 NUC-QAM con una  $SNR = 10 \text{ dB}$ ; además se puede ver que las distancias entre los símbolos son diferentes.

Se pueden identificar claramente tres parámetros: el punto ideal de la constelación, los bits modulados y los bits recibidos sobre un canal AWGN con desvanecimiento.



**Figura 14-2.** Constelación 64 NUC-QAM, SNR=10dB

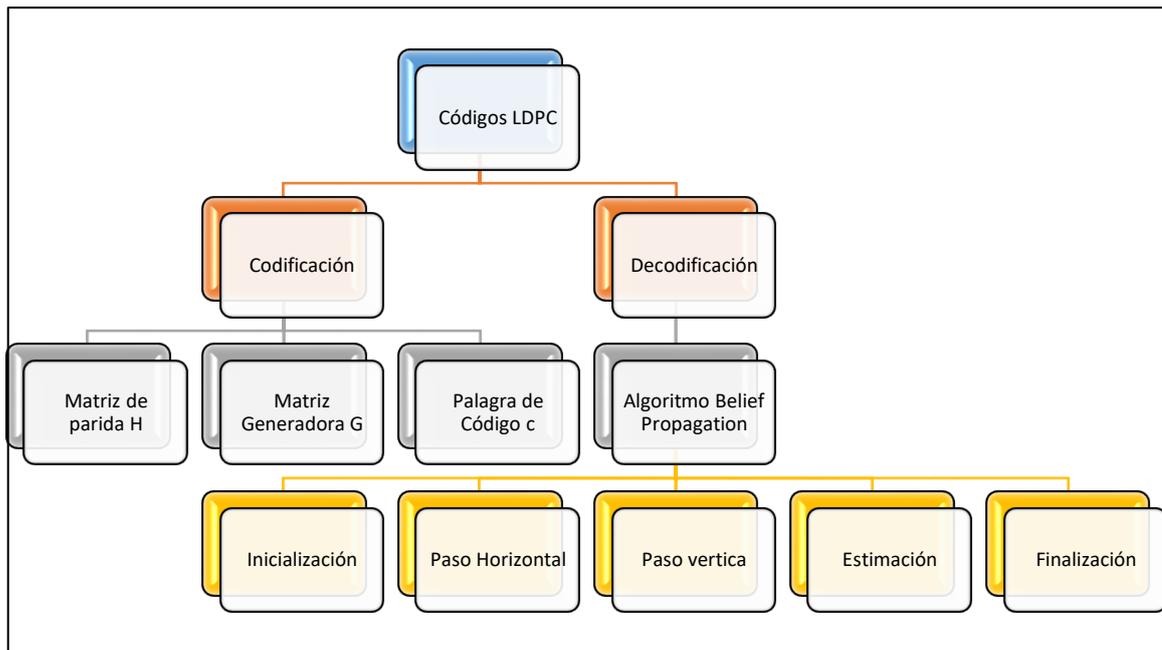
Fuente: Realizado por Mónica F.

Como las constelaciones dependen del orden de la modulación y de la SNR, en el Anexo II se encuentran las constelaciones correspondientes a 256, 1024 y 4096 NUC-QAM con diferentes SNR.

### 2.13. CODIFICACIÓN LDPC (Low Density Parity Check)

Los códigos de verificación de paridad de baja densidad son códigos especificados por una matriz que contiene principalmente ceros (0) y solo un pequeño número de unos (1) (Gallager, 1962, pág. 7).

Al construir códigos LDPC se divide el análisis en dos etapas: en el transmisor la etapa de codificación y en el receptor la etapa de decodificación.



**Figura 15-2.** Diagrama de bloques de la codificación y decodificación LDPC

Fuente: Realizado por Mónica F.

### 2.13.1. Codificación

#### 2.13.1.1. Matriz de Verificación de Paridad (H)

La matriz de verificación de paridad, es una matriz compuesta por unos (1) y ceros (0) lógicos; la densidad de unos lógico debe ser baja, es decir la cantidad de unos lógicos en filas y columnas deben ser muchos menores de los valores  $M$  y  $N$ , es decir de la dimensión de la matriz  $H$  (Gallager, 1962, pág. 12).

La matriz de verificación de paridad  $H$  es la base para la creación de los códigos LDPC, será la base para la creación de la matriz Generadora y es la matriz usada para la decodificación en la recepción.

Para el desarrollo de este proyecto se tiene un numero fijos de unos lógicos por filas y un número fijo de unos lógicos por columnas que cumplen la condición:

$$W_r M = W_c N \quad (2.18)$$

Dónde  $W_r$  representa el número de unos lógicos por filas y  $W_c$  representa el número de unos lógicos por columnas.

Además, se utilizará la Matriz de Gallager, propuesta por Robert Gallager en 1962.

Para construir una matriz de Gallager se deben seguir los siguientes pasos:

- Las filas de  $H$  se dividen en grupos de  $Wc$  con  $M/Wc$  filas en cada grupo.
- En el primer grupo  $M/Wc$  de filas, se colocan  $Wr$  unos lógicos consecutivos ordenados de izquierda a derecha por las columnas.
- Para los demás grupos de  $M/Wc$  filas se realiza una permutación de las columnas del primer grupo (Rao, 2015, pág. 281)

### 2.13.1.2. Matriz Generadora

La matriz Generadora ( $G$ ) se obtiene a partir de la matriz de verificación de paridad ( $H$ ) por medio de eliminación Gaussiana; que consiste en realizar cambios en las posiciones de filas o columnas y operaciones en Galois Field GF(2) para conseguirla.

*Galois Field* GF(2) quiere decir Campo de Galois de dos dimensiones  $\{0,1\}$  con las operaciones de suma y producto, usuales en esos anillos. Una de las propiedades que debe satisfacer un anillo para que sea campo es que cualquier elemento distinto de cero tiene un inverso; cosa que sucede ya que el inverso del 0 es el 1 (Gallager, 1962, pág. 15)

### 2.13.1.3. Palabra codificada

Una vez que se tiene la matriz de verificación de paridad ( $H$ ) y la matriz Generadora ( $G$ ). Para obtener la palabra código ( $c$ ), los datos ( $m$ ) se multiplican con la matriz generadora ( $G$ ).

$$c = m \times G, \quad (2.19)$$

Dónde  $c$  representa el mensaje codificado

Una vez el mensaje codificado pasa por las etapas de modulación, canal AWGN con desvanecimiento, llega al receptor, se demodula y se realiza la decodificación.

## 2.13.2. Decodificación

A los algoritmos utilizados en el proceso de decodificación de códigos LDPC se los conocen como “Algoritmos de Paso de Mensaje”, porque sus operaciones pueden ser explicadas como el paso de mensajes a través de los bordes en los grafos de Tanner.

En este proyecto se utilizará el algoritmo “*Belief Propagation*”, (BP o Probabilidad de Propagación), ya que es un método ampliamente utilizado para solucionar problemas de inferencia en modelos gráficos. Se aplica básicamente en dos situaciones: (1) cuando el objetivo es obtener distribuciones marginales para algunas de las variables presentes en el problema, y (2) cuando, dado un modelo, se busca el estado global más probable para un determinado problema. Estos dos casos se conocen habitualmente con el nombre de algoritmo *sum-product* y *max-product* respectivamente, los mismos que más adelante se explicaran. Se ha demostrado en

diferentes aplicaciones, tales como LDPC, turbo códigos, procesamiento de imágenes, u optimización, es muy eficiente (Rao, 2015, pág. 286)

Además el algoritmo BP es de decisión suave basado en “*Message Passing*” (Rao, 2015, pág. 286).

### ***Message Passing* (Paso de mensaje)**

Para cada iteración del algoritmo se realizan los siguientes pasos:

1. Cada nodo de bit envía un mensaje a cada nodo de paridad conectado.
2. Cada nodo de paridad envía un mensaje a cada nodo de bit conectado.
3. Para cada bit de la palabra de código recibida, se calcula la probabilidad a posteriori que el bit tome en el valor “1 lógico”, con todos los valores pico de la señal y debe cumplir con todas las condiciones de paridad.

### ***Belief Propagation***

Como se explicó anteriormente el algoritmo de “*Belief Propagation*” es una derivación del algoritmo de “*Message Passing*”, en el cual los mensajes que cada nodo procesa son probabilidades.

#### **2.13.2.1. Inicialización**

Para  $i = 0, 1, \dots, N - 1$ , se fija  $q_{i,j}(x)$  basado en  $Y_i = y_i$  al asumir que las probabilidades a priori de los  $W_i$  son  $\{+1, -1\}$ , y los posibles valores de  $x = \{0, 1\}$  debido a que se maneja en GF(2). Para cada valor de  $i, j$  donde  $H_{i,j} = 1$ , las variables  $q_{i,j}(1)$  y  $q_{i,j}(0)$  son inicializados a los valores  $p_j(1)$  y  $p_j(0)$ .

$$q_{i,j}(0) = p_j(0) = \frac{1}{1 + e^{\frac{2y_i}{\sigma^2}}} \quad (2.20)$$

$$q_{i,j}(1) = p_j(1) = \frac{1}{1 + e^{\frac{-2y_i}{\sigma^2}}} \quad (2.21)$$

#### **2.13.2.2. Paso horizontal**

Para la actualización de información y envío de mensajes entre nodos se define:

$$\delta q_{i,j} = q_{i,j}(0) - q_{i,j}(1) \quad (2.22)$$

$$\delta r_{i,j} = \prod_{j' \in N(i)/j} \delta q_{i,j'} \quad (2.23)$$

Dónde  $N(i)$  representa el conjunto de subíndices de los nodos de bits que tienen conexión con el nodo de paridad, mientras que  $N(i)/j$  representa la exclusión del nodo  $j$  de ese conjunto. Obtenido el valor  $\delta r_{i,j}$  se puede obtener los valores:

$$r_{i,j}(0) = \frac{1}{2}(1 + \delta r_{i,j}) \quad (2.24)$$

$$r_{i,j}(1) = \frac{1}{2}(1 - \delta r_{i,j}) \quad (2.25)$$

### 2.13.2.3. Paso Vertical

Se estima el valor de  $q_{i,j}$  para valor elemento  $i,j$

$$q_{i,j}(x) = \alpha_{i,j} p_j(x) \prod_{j' \in \frac{M(j)}{i}} r_{i',j}(x), \quad (2.26)$$

Dónde  $M(i)$  representa el conjunto de subíndices de los nodos de paridad que tienen conexión con el nodo de bit, mientras que  $M(j)/i$  representa la exclusión del nodo  $i$  de ese conjunto. La constante  $\alpha_{i,j}$  debe cumplir la condición de:

$$q_{i,j}(0) + q_{i,j}(1) = 1. \quad (2.27)$$

Se agrupa los términos para obtener:

$$c_{i,j}(x) = p_j(x) \prod_{j' \in \frac{M(j)}{i}} r_{i',j}(x) \quad (2.28)$$

Al reemplazar la ecuación (2.24) en la ecuación (2.22) se tiene que:

$$q_{i,j}(x) = \alpha_{i,j} C_{i,j}(x) \quad (2.29)$$

Debido a que se debe cumplir la ecuación (2.27), se reemplaza la ecuación (2.29) en la ecuación (2.23) y se obtiene el valor que  $\alpha_{i,j}$  debe tomar:

$$\alpha_{i,j} C_{i,j}(0) + \alpha_{i,j} C_{i,j}(1) = 1 \quad (2.30)$$

$$\alpha_{i,j} = \frac{1}{C_{i,j}(0) + C_{i,j}(1)} \quad (2.31)$$

Finalmente al reemplazar la ecuación (2.31) en ecuación (2.29) se obtiene:

$$q_{i,j}(x) = \frac{C_{i,j}(x)}{C_{i,j}(0) + C_{i,j}(1)} \quad (2.32)$$

#### 2.13.2.4. Estimación

Posteriormente, para cada  $j$  se realiza el cálculo de las probabilidades pseudo a posteriori  $q_j(0)$  y  $q_j(1)$ , así:

$$q_j(x) = \beta_j p_j(x) \prod_{i \in M(i)} r_{i,j}(x) \quad (2.33)$$

De la ecuación (2.19), se descompone el factor por:

$$q_j(x) = \beta_j p_j(x) \left( \prod_{i' \in M(j)/i} r_{i',j}(x) \right) r_{i,j}(x) \quad (2.34)$$

Al reemplazar la ecuación (2.28) en ecuación (2.34):

$$q_j(x) = \beta_j C_{i,j}(x) r_{i,j} \quad (2.35)$$

De igual manera la constante  $\beta_j$  debe cumplir con la ecuación (2.27):

$$\beta_j C_{i,j}(0) r_{i,j}(0) + \beta_j C_{i,j}(1) r_{i,j}(1) = 1 \quad (2.36)$$

$$\beta_j = \frac{1}{C_{i,j}(0) r_{i,j}(0) + C_{i,j}(1) r_{i,j}(1)} \quad (2.37)$$

Al reemplazar la constante en la ecuación (2.34) se tiene:

$$q_j(x) = \frac{C_{i,j}(x) r_{i,j}(x)}{C_{i,j}(0) r_{i,j}(0) + C_{i,j}(1) r_{i,j}(1)} \quad (2.38)$$

Estos valores son usados dentro de la iteración para obtener el valor tentativo para cada símbolo.

$$c_{e_i} = \begin{cases} q_j(1) > q_j(0) & x = 1 \\ \text{caso contrario} & x = 0 \end{cases} \quad (2.39)$$

### 2.13.2.5. Finalización

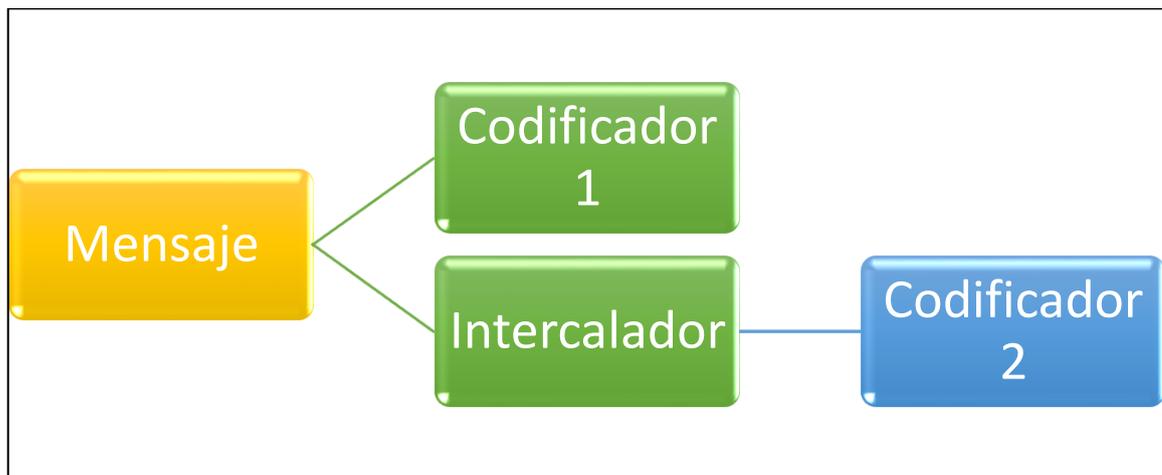
Todo lo anteriormente descrito se repetirá recursivamente en cada iteración; al final de cada iteración se realiza las siguientes comprobaciones, que en el caso que se cumplan, el algoritmo termina:

- Si  $H \times c_e = 0$ .
- Si se cumple el máximo de iteraciones propuestas. Esto depende de la configuración del algoritmo (Rao, 2015, pág. 295).

## 2.14. CODIFICACIÓN TURBO

### 2.14.1. Codificación

El esquema de codificación empieza con la información o mensaje que se enviará. El mensaje se envía a través de un codificador 1 y también se lo intercala para ser enviado por un codificador 2. El codificador 1 y 2 son normalmente idénticos, la razón por la que su salida es diferente depende totalmente del intercalado (Vucetic, 2000, pág. 10).



**Figura 16-2.** Codificador básico de turbo código

**Fuente:** Realizado por Mónica F.

El mensaje, junto con el resultado del codificador 1 y el resultado del codificador 2, son concatenados antes de ser transmitidos.

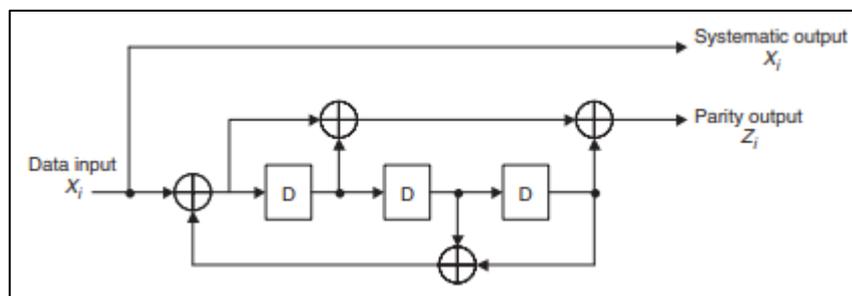
Como se mencionó anteriormente, los turbo códigos se pueden codificar de diferentes maneras. Uno de los métodos más utilizados se llama codificación convolucional que se describe a continuación.

### 2.14.1.1. Códigos convolucionales

Normalmente, en los códigos turbo, se utilizan dos códigos recursivos de convolución sistemática (RSC) (Berrou, Glavieux, & Thitimajshima, 1993). Estos dos códigos se concatenan y luego se envían a través de un canal.

Los códigos RSC son muy similares en naturaleza a los códigos convolucionales convencionales, también llamados códigos convolucionales no sistemáticos (NSC), que se usan comúnmente para sistemas celulares 2G (Vucetic, 2000, pág. 14).

En codificador turbo las palabras de código deben poder dividirse en datos separados y campos de paridad, lo que queremos es un codificador sistemático, uno cuya entrada aparezca en la salida; una palabra de código sistemática se puede dividir en datos y componentes de paridad, como se observa en la figura 6-3.



**Figura 17-2.** Codificador RSC con  $R=1/2$

Fuente: (Valenti, Matthew C.; Jian Sun, 2004, pág. 380)

Una observación importante en RSC es que, dado que son recursivos, el codificador no volverá al estado de cero al añadir una cola de cero. En cambio, el contexto actual del codificador debe adjuntarse en la cola.

### 2.14.1.2. Intercalado

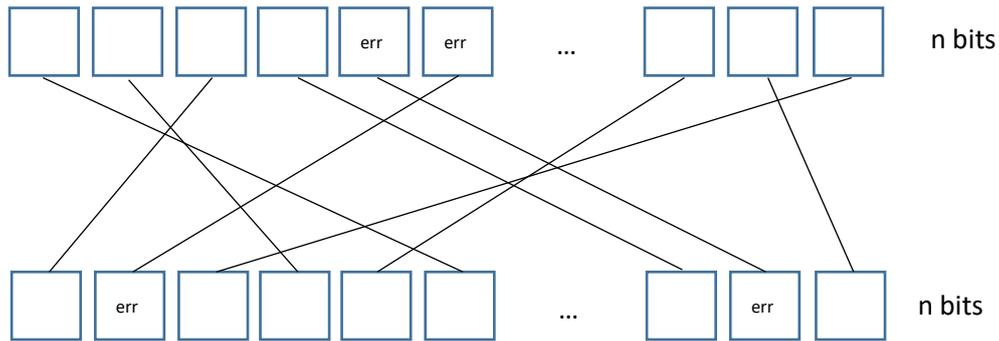
Su función es reorganizar el orden de los bits de datos de una manera prescrita, pero irregular. Aunque el mismo conjunto de bits de datos está presente en la salida del intercalador, el orden de estos bits se ha cambiado.

Sin el intercalador, los dos codificadores RSC recibirían los datos en el mismo orden, por lo tanto, al suponer que los codificadores RSC fueran idénticos, sus resultados serían los mismos; si esto sucede el código generado no sería muy poderoso (Vucetic, 2000, pág. 30).

Al usar el intercalador los datos se reorganizan de modo que el segundo codificador los reciba en un orden diferente. Por lo tanto, la salida del segundo codificador será diferente a la salida del primer codificador, ya que el código es más complejo.

El objetivo principal del entrelazado es proteger los datos de los errores de ráfaga. Esto se puede explicar al observar el paso de intercalación como una permutación temporal de bits.

Si se producen errores en  $n$  bits consecutivos en un segmento de código sin intercalar, los errores se distribuirán en todo el bloque en el segmento de código intercalado, como se ilustra en la figura 7-3. Entonces, lo que realmente hace el intercalador es aumentar la distancia mínima de Hamming al código concatenado (Mazumdar, Chaturvedi, & Banerjee, 2005, pág. 1) esto permite evitar que los errores se presenten consecutivamente en la palabra de código.



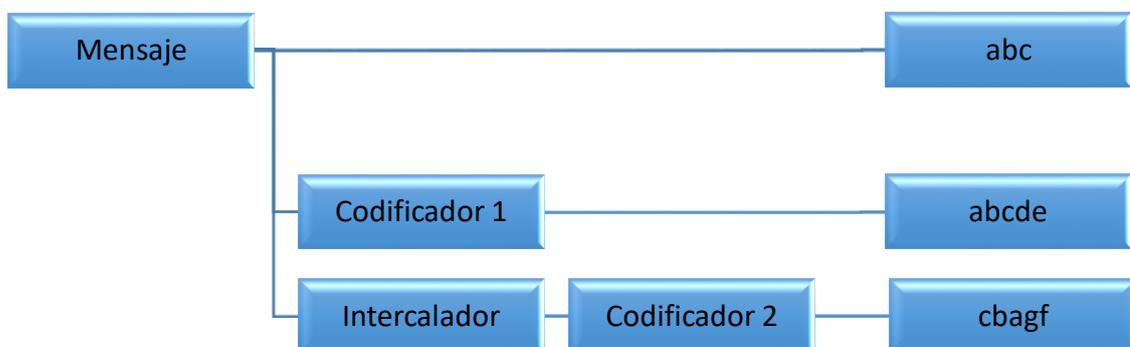
**Figura 18-2.** Dispersión de los errores, entrelazado

Fuente: (Mazumdar, Chaturvedi, & Banerjee, 2005, pág. 1)

### 2.14.1.3. Perforación

La perforación (*puncturing* en inglés) se utiliza principalmente para optimizar el ancho de banda, al eliminar algunos de los bits de paridad después de la codificación. Esto es, ambos codificadores pueden querer enviar la información. Esta repetición es ineficiente por lo que un codificador puede perforar sus bits de información. Por lo tanto, se ignoran los bits de información en cualquiera de los codificadores, y se aumenta la velocidad del código.

Para entender de una mejor manera cómo funciona el perforado tenemos por ejemplo (Vucetic, 2000, pág. 41),



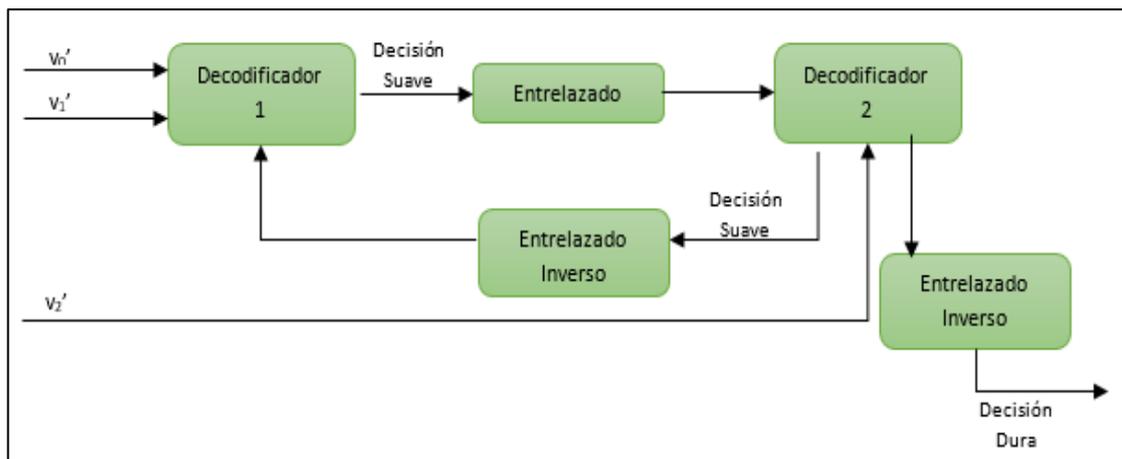
**Figura 19-2.** Ejemplo de la codificación Turbo

Fuente: Realizado por Mónica F.

Se tiene las tres salidas del decodificador, tal como se muestra en la figura 8.3, aquí se observa " abc " que se envía tres veces, lo que supone un desperdicio de energía y tiempo. Por lo tanto, se debe utilizar la perforación para no enviarlo tres veces sino solo una, y el resultado puede ser " abcdefg”.

### 2.14.2. Decodificación

En resumen, los esquemas de decodificación de los Turbo códigos pueden describirse en la figura 9.3.



**Figura 20-2.** Decodificador turbo código

Fuente: (Vucetic, 2000, pág. 41)

En la figura 20-2, los datos del canal  $v$  se derivan en diferentes partes,  $v_0$  es la parte del mensaje,  $v_1$  es la información del primer codificador y  $v_2$  es la información del segundo codificador.

Los dos decodificadores son normalmente similares, sin embargo, dado que los datos codificados han sido intercalados, los datos entre los dos codificadores tienen que desentrelazarse e intercalarse repetidamente para coincidir con su codificador asociado (Vucetic, 2000, pág. 83).

En consecuencia, la salida del primer decodificador se envía al segundo decodificador, que luego utiliza estos datos para estimar la palabra de código. Esta información se envía de vuelta al primer codificador. Este método iterativo normalmente se ejecuta al menos seis veces antes de que la estimación sea tan buena para que el decodificador haya encontrado la palabra de código correcta. Sin embargo, este método también puede experimentar un desajuste sino se encuentra la palabra de código en el número de iteraciones propuesto.

Los resultados que se envían entre los decodificadores se denominan decisiones blandas, y son cálculos entre números reales. Estos números pueden ir de  $\{-\infty, +\infty\}$ . Una decisión suave sería que los números negativos probablemente se interpretaran como un 0, y una decisión suave

para los números positivos será 1. Pero esto es incierto, ya que, si la decisión suave es 0, el número puede ser 0 o 1 y se requieren iteraciones adicionales. Cuando se toma la decisión final el resultado se llama una decisión difícil. También debe tenerse en cuenta que, si los datos enviados desde el codificador se perforaron, el receptor debe tener esto en cuenta.

Hay varios algoritmos utilizados para decodificar un código turbo. Para motivos de esta investigación se describen dos algoritmos diferentes, a saber, el Log-MAP y el algoritmo SOVA.

### 2.14.2.1. Algoritmo MAP

El Algoritmo probabilidad máxima a posteriori (MAP), es capaz de producir la probabilidad a posteriori (APP) para cada bit decodificado, el cual encuentra el bit de información más probable que se haya enviado dada la secuencia codificada (Valenti, Matthew C.; Jian Sun, 2004, pág. 390).

El decodificador MAP recibe una secuencia con los datos transmitidos  $r$  y las relaciones de verosimilitud logarítmica a priori de los bits de información  $L_a(u_l)$ ,  $l = 0, 1, \dots, h - 1$ ;  $h$  es la longitud del bloque de la secuencia  $u$ . El algoritmo calcula los LLR a posteriori (LLR de MAP) para cada bit de información

$$L(u_l) \equiv \ln \left[ \frac{P(u_l = +1|r)}{P(u_l = -1|r)} \right] \quad (2.40)$$

La salida del decodificador está dada por decisiones difíciles de los valores estimados resultantes  $\hat{u}$ , donde

$$\hat{u}_l = \begin{cases} +1 & \text{si } L(u_l) > 0 \\ -1 & \text{si } L(u_l) < 0 \end{cases}, l = 0, 1, \dots, h - 1 \quad (2.41)$$

La ecuación (2.36) se puede reformular después de varios pasos así

$$L(u_l) \equiv \ln \left[ \frac{\sum_{(s',s) \in S_l^+} p(s_l = s', s_{l+1} = s, r)}{\sum_{(s',s) \in S_l^-} p(s_l = s', s_{l+1} = s, r)} \right] \quad (2.42)$$

Dónde  $p$  es la función de densidad de probabilidad y  $s$  es algún estado en el enrejado. El conjunto  $S_l^+$  denota el conjunto de todos los pares de estados  $s_l = s', s_{l+1} = s$  en el enrejado, que corresponde al bit de entrada  $u_l = +1$  en el tiempo  $l$ , y el conjunto  $S_j$  será todos los pares correspondientes con  $u_l = -1$ . Además, la función de densidad de probabilidad se puede evaluar recursivamente mediante

$$p(s', s, r) = p(s', s, r_{t < l}, r_l, r_{t > l}) \quad (2.43)$$

Dónde  $r_l$  representa la porción de la secuencia recibida en el tiempo  $l$ , y  $r_{t<l}$  y  $r_{t>l}$  representa la parte recibida, antes y después del tiempo  $l$ , respectivamente. Luego, después de usar la regla de Bayes se tiene

$$\begin{aligned}
p(s', s, r) &= p(r_{t>l}|s', s, r_{t<l}, r_l)p(s', s, r_{t<l}, r_l) \\
&= p(r_{t>l}|s', s, r_{t<l}, r_l)p(s, r_l|s', r_{t<l})p(s', r_{t<l}) \quad (2.44) \\
&= p(r_{t>l}|s)p(s, r_l|s')p(s', r_{t<l})
\end{aligned}$$

La última igualdad se divide en tres partes, de la siguiente manera

$$\alpha_l(s') \equiv p(s', r_{t<l}) \quad (2.45)$$

$$\eta(s', s) \equiv p(s, r_l|s') \quad (2.46)$$

$$\beta_{l+1}(s) \equiv p(r_{t>l}|s) \quad (2.47)$$

La ecuación (2.44) puede ser reescrita como

$$p(s', s, r) = \beta_{l+1}(s)\eta(s', s)\alpha_l(s') \quad (2.48)$$

La métrica hacia adelante  $\alpha_{l+1}(s)$  se puede derivar de la ecuación (2.44), y esta recursión hacia adelante se define como

$$\alpha_{l+1}(s) = \sum_{s' \in \sigma_l} \eta(s', s)\alpha_l(s') \quad (2.49)$$

Dónde  $\sigma_l$  es el conjunto de todos los estados en el tiempo  $l$ , y la métrica hacia atrás para  $\beta_l(s')$  se puede definir de manera similar como

$$\beta_l(s') = \sum_{s \in \sigma_l} \eta(s', s)\beta_{l+s}(s) \quad (2.50)$$

Luego la métrica de rama  $\eta(s', s)$  se define como

$$\eta(s', s) = P(u_l)p(r_l|v_l) \quad (2.51)$$

Que después de algunas modificaciones puede escribirse como

$$\eta(s', s) = P(u_l)e^{-\frac{E_s}{N_0}\|r_l - v_l\|^2} \quad (2.52)$$

Dónde  $\|r_l - v_l\|^2$  es la distancia euclidiana entre el  $r_l$  recibido y el  $v_l$  transmitido.  $\frac{E_s}{N_0}$  es la relación de señal a ruido (SNR). Las condiciones iniciales para  $\alpha_0(s)$  y  $\beta_k(s)$ , donde  $K = h + m$  es la longitud de la secuencia de entrada  $u$ , se define por

$$\alpha_0(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases} \quad (2.53)$$

Y

$$\beta_k(s) = \begin{cases} 1, & s = 0 \\ 0, & s \neq 0 \end{cases} \quad (2.54)$$

Finalmente, el algoritmo se dará.

En resumen, los pasos para crear un algoritmo MAP son:

1. Establezca las métricas hacia adelante y hacia atrás de acuerdo con la ecuación (2.53) y (2.54), respectivamente.
2. Usa la ecuación (2.51) para calcular las métricas de rama  $\gamma_l^*(s', s)$ ,  $l = 0, 1, \dots, K - 1$ .
3. Usa la ecuación (2.49) para calcular las métricas de avance  $\alpha_{l+1}^*(s)$ ,  $l = 0, 1, \dots, K - 1$ .
4. Usa la ecuación (2.50) para calcular las métricas hacia atrás  $\beta_l^*(s')$ ,  $l = K - 1, K - 2, \dots, 0$ .
5. Al usar la ecuación (2.42), calcule las relaciones de probabilidad logarítmica APP  $L(u_l)$  para  $l = 0, 1, \dots, h - 1$ .
6. Finalmente, use la ecuación (2.41) para calcular las decisiones difíciles  $\hat{u}_l$ ,  $l = 0, 1, \dots, h - 1$ .

#### 2.14.2.2. Algoritmo de Log-MAP

Probabilidad logarítmica máxima *A-posteriori* (Log-MAP), es una versión modificada del algoritmo de Máxima Probabilidad *A-Posteriori* (MAP). La decodificación Log-MAP se usa generalmente con codificadores sistemáticos ya que la primera estimación del valor real  $u_l$  es la información sistemática recibida  $v_0$  (Valenti, Matthew C.; Jian Sun, 2004, pág. 390).

En el algoritmo MAP las LLR a posteriori se calculan con la ecuación (2.42), donde las probabilidades se reemplazan con la ecuación (2.48) junto con las ecuaciones (2.49), (2.50) y (2.51).

La simplificación de estas ecuaciones (2.48), (2.49), (2.50) y (2.51), se realiza transfiriéndolas al dominio de aritmética de registros y al utilizar la aproximación

$$\ln\left(\sum_i e^{x_i}\right) = \max_i(x_i) \quad (2.55)$$

$\max_i(x_i)$  Es el valor máximo de  $x_i$ . Entonces,  $A_l(s)$ ,  $B_l(s)$  y  $\Gamma_l(s', s)$  son, después de usar las definiciones dadas en las ecuaciones (2.49), (2.50) y (2.51), definidas por

$$A_{l+1}(s) \triangleq \ln(\alpha_{l+1}(s)) \approx \max_{s'} (A_l(s') + \Gamma_l(s', s)) \quad (2.56)$$

$$B_l(s') \triangleq \ln(\beta_l(s')) \approx \max_s (B_{l+1}(s) + \Gamma_l(s', s)) \quad (2.57)$$

$$\Gamma_l(s', s) \triangleq \ln(\gamma_l(s', s)) = \frac{u_l L_a(u_l)}{2} + \frac{L_c r_l \cdot v_l}{2} \quad (2.58)$$

Dónde  $L_a$  son los valores de verosimilitud logarítmica a priori de los bits de información enviados,  $v_l$  es el bit enviado,  $r_l$  es el bit recibido y  $L_c$  es el factor de fiabilidad del canal.

Dado que la aproximación en la ecuación (2.55) produce una salida suave inferior en comparación con el algoritmo MAP, el algoritmo Log-MAP usa el logaritmo jacobiano para solucionar este problema.

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}) \quad (2.59)$$

Cuando la ecuación (2.56) y (2.57) usan la ecuación (2.59) en lugar de la ecuación (2.55), el algoritmo se llama Log-MAP.

### 2.14.2.3. SOVA (Algoritmo de Viterbi de salida suave)

La decodificación suave de Viterbi, o SOVA para abreviar, es una técnica de decodificación que es similar al algoritmo de Viterbi.

#### 2.14.2.3.1. Algoritmo de Viterbi

El algoritmo de Viterbi fue introducido en 1967 por Andrew J. Viterbi (Jr., 1973). Es un algoritmo de decodificación para códigos convolucionales que explota el enrejado de decodificación (Hagenauer & Hoehner, 1989, pág. 2).

Una palabra recibida  $r$  puede diferir de la palabra de código enviada  $v$ . Esta diferencia se denomina el vector de error, denotado  $E$ . Cada camino a través del enrejado representa una palabra de código recibida, y la función de probabilidad logarítmica,  $\log(r|v)$ , se llama la métrica asociada con la ruta de la palabra de código enviada, que a menudo se denota como  $M(r|v)$ .

El algoritmo es bastante simple y contiene 3 pasos:

1. Comience en el estado más a la izquierda del enrejado y establezca el tiempo  $t = i$ . Calcule y guarde la ruta y la métrica de los estados actuales.
2. Para cada estado en el tiempo,  $t = +1$ , calcule la métrica de todas las rutas entrantes agregando la métrica del estado anterior con la métrica para la ruta de conexión entre los dos nodos. Para cada estado, compare todas las métricas que ingresan al estado y seleccione la ruta con la métrica más grande. Almacene esta ruta junto con su métrica y elimine todas las demás rutas.
3. Si  $t$  es menor que la longitud de la secuencia de información, repita el paso 2, Si no, deténgase y devuelva la ruta del estado actual.

El algoritmo de Viterbi no puede ser utilizado directamente en los códigos turbo ya que la salida del algoritmo está basada en decisiones difíciles. Sin embargo, después de algunos ajustes el algoritmo utiliza una entrada suave y devuelve una salida suave, este algoritmo se llama SOVA.

Hay dos diferencias entre el algoritmo de Viterbi y el SOVA (Rao, 2015, pág. 162). En primer lugar, las métricas de ruta se modifican para utilizar información a priori cuando se decide la ruta a través del enrejado que es más probable. En segundo lugar, la salida de software tiene información de fiabilidad sobre la salida decodificada. Esta información de fiabilidad es la relación de verosimilitud logarítmica a posteriori. Este tipo de decodificadores también se denominan decodificadores *Soft-In Soft- Out* (SISO) (Rao, 2015, pág. 162).

Considere un enrejado cualquiera, cada ruta a través de este enrejado representa diferentes palabras de código. Todos estos caminos se pueden dividir en estados que están conectados por un borde a uno o dos otros estados en la misma ruta. Cada uno de estos estados tiene una métrica que denota la "probabilidad" de la ruta de supervivencia que pasa por su estado. Esta métrica  $M(s_l)$  depende de la métrica del estado anterior  $M(s)$  en la ruta y la métrica para el borde,  $\gamma$  entre ellas, como en el algoritmo original de Viterbi. La métrica está definida por

$$M(s_l) \triangleq M(s'_{l-1}) + \ln(\gamma_l(s', s)) \quad (2.60)$$

En un enrejado binario, todos los estados tendrán dos bordes entrantes. Como el enrejado comienza y termina en el estado de cero, los estados iniciales tendrán uno o ninguno de los bordes entrantes, ya que el resto de los estados no se pueden alcanzar y, por lo tanto, se ignoran. De todos modos, cuando dos rutas entran en un estado, sus métricas se calculan, y luego se comparan, y se selecciona la métrica más grande  $M(s_l)$ , mientras que la otra  $M(\hat{s}_l)$  se descarta. La diferencia entre estas dos métricas es

$$\Delta_l = M(s_l) - M(\hat{s}_l) \geq 0 \quad (2.61)$$

Dónde  $\Delta_l$  es la relación log-verosimilitud de la métrica seleccionada que es la decisión correcta.

Se muestra en (Hagenauer & Hoehner, 1989, pág. 2) que la relación log-verosimilitud del bit de información  $u_l$  dado el bit  $r_l$  recibido se puede aproximar por

$$L(u_l|r_l) \approx u_l \min_{i=l, \dots, l+\delta} (\Delta_i) \quad (2.62)$$

Dónde  $\delta$  es una cantidad de estados después de  $l$ .

El algoritmo SOVA sigue los mismos pasos que el algoritmo de Viterbi. Sin embargo, la entrada y salida blandas se calculan mediante las ecuaciones anteriores para encontrar la ruta más probable. Cuando se ha encontrado la ruta más probable, las decisiones difíciles se calculan con la ecuación (2.41).

Además, se debe tener en cuenta que tanto las modulaciones no uniformes, como las codificaciones LDPC y Turbo se las realiza a nivel de bit; es decir se analizan paquetes de bit para su transmisión t recepción.

## CAPÍTULO III

### METODOLOGÍA DE INVESTIGACIÓN

#### 3.1. Tipo y diseño de investigación

El presente trabajo es una investigación experimental y documental por lo que se basa en estudios realizados y se aplica criterios técnicos para el diseño de la propuesta con parámetros debidamente sustentados de acuerdo a la investigación a realizarse y a la bibliografía encontrada, al ser un estudio transversal.

#### 3.2. Métodos de investigación

**Método Deductivo**, para la comprensión de conceptos, definiciones, órdenes, tipos y parámetros, que involucran el conocimiento de los tipos de codificación de canal sobre canales AWGN con desvanecimiento al utilizar constelaciones no uniformes y su aplicación en la transmisión de información.

**Método Inductivo**, mediante el estudio bibliográfico y la utilización de parámetros adecuados para la evaluación de la codificación de canal sobre canales AWGN con desvanecimiento al utilizar constelaciones no uniformes que permitan alcanzar los objetivos planteados de esta investigación.

**Análisis**, al permitir clasificar las ideas para sintetizarlas, describirlas, concretarlas y así poder evaluar la codificación de canal y su aplicación en la transmisión de información.

La tasa de código que se utilizó es de  $1/2$ , que dependerá del desarrollo de la investigación; además se realizarán pruebas con *Puncturing* en las codificaciones del canal.

La evaluación de la codificación de canal en el desempeño de modulaciones no uniformes sobre canales AWGN con desvanecimiento se lo realiza mediante el cálculo de la Tasa de Error de Bit obtenida en el receptor, en función de la relación señal/ruido.

Para el análisis de desempeño se comparan los resultados obtenidos para las Constelaciones No Uniformes como para las Constelaciones Uniformes; con codificación de canal LDPC y Turbo Códigos,

**Síntesis**, enfocar los análisis para establecer las conclusiones de la investigación.

### **3.3. Enfoque de la investigación**

Se aplica un enfoque cualitativo de los datos obtenidos, al realizar pruebas de diseño con diferentes tipos de códigos de canal, al utilizar herramientas matemáticas para encontrar los parámetros de tasa de transmisión, relación señal/ruido, adecuados, para la transmisión de información en canales AWGN con desvanecimiento al utilizar modulaciones no uniformes.

Las características del objeto de investigación deben enfocarse dentro del ámbito de las telecomunicaciones. En la presente investigación se evalúa la codificación de canal en el desempeño de modulaciones no uniformes sobre canales AWGN con desvanecimiento.

### **3.4. Alcance de la investigación**

Por tratarse de la evaluación de la codificación de canal en el desempeño de modulaciones no uniformes sobre canales AWGN con desvanecimiento se espera tener un alcance descriptivo de acuerdo a los datos encontrados y obtenidos en la simulación.

### **3.5. Unidad de análisis**

Tipos de códigos de canal: se obtienen datos del material bibliográfico

Orden de las Modulaciones no uniformes: se obtienen datos del material bibliográfico.

Canales AWGN con desvanecimiento: se obtienen datos del material bibliográfico

### **3.6. Técnica de recolección de datos primarios y secundarios**

Lectura científica y observación

Fuentes primarias obtenidas de los datos bibliográficos.

Recolección de datos del sistema de simulación.

### **3.7. Instrumentos de recolección de datos primarios y secundarios**

Hojas de cálculo y gráficas.

### **3.8. Instrumentos para procesar datos recopilados**

Computador y programa de simulación.

## CAPÍTULO IV

### RESULTADOS Y ANÁLISIS

Ahora, con la herramienta de simulación desarrollada en Matlab, la cual se muestra en el Anexo I, es posible poner a prueba el desempeño de las codificaciones M-NUCQAM. Para esto se utiliza una medida de errores conocida como la Tasa de Error de Bit (Bit Error Rate), la cual expresa el porcentaje de bits errados; al comparar el flujo de bits a transmitir con el flujo de bits recibidos en el receptor.

Las diferentes simulaciones se realizaron al utilizar una cadena de bits aleatorios de 100000 símbolos.

#### 4.1. Resultados para un canal AWGN con desvanecimiento al utilizar codificación LDPC y Constelaciones Uniformes versus No Uniformes

En la tabla 1-4., se muestra la métrica de desempeño de un canal AWGN con desvanecimiento, para una modulación de 64 QAM, con codificación LDPC tanto con UC y NUC; entre un rango de valores de SNR (17 a 20 dB). Al utilizar constelaciones no uniformes (NUC) se obtienen un BER mucho más pequeño que al utilizar la constelación uniforme; es decir una mejora de un 90% aproximadamente al utilizar constelaciones no uniformes.

**Tabla 1-4.** Métricas de desempeño, para 64QAM, codificación LDPC

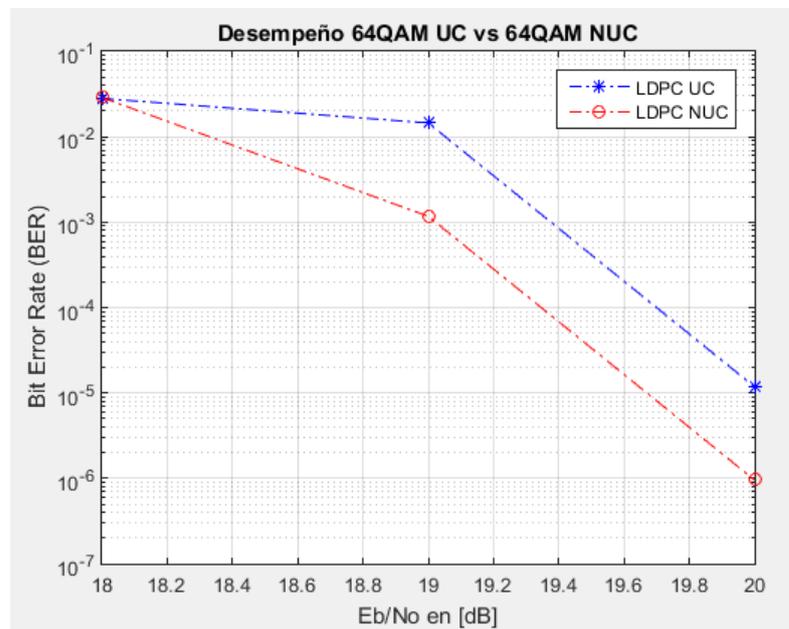
ber		
SNR (dB)	UC LDPC	NUC LDPC
17	0,04203667	0,034868333
18	0,02811167	0,028806667
19	0,01451000	0,001170000
20	0,00001180	9,55167E-07

Fuente: Realizado por Mónica F.

Las Métricas de desempeño, en un canal AWGN con desvanecimiento correspondiente a las constelaciones uniformes y no uniformes de orden 256QAM, 1024QAM y 4096QAM, se encuentran en el Anexo IV.

Para visualizar de manera más clara los datos de la tabla 5.1., se gráfica la SNR en función del BER.

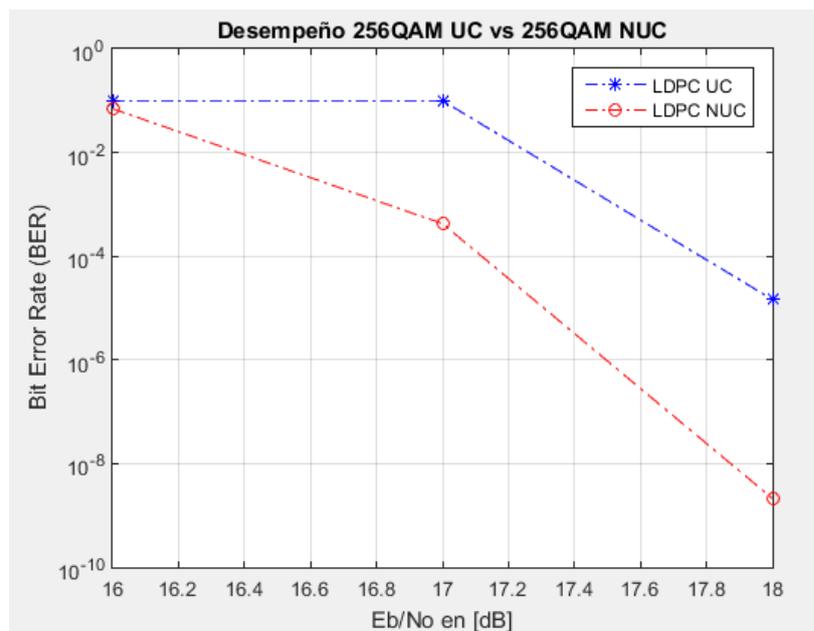
En la figura 1-4, se muestran los resultados del desempeño, para 64QAM en la cual se observa que la curva al utilizar NUC tiene menos errores que la UC, hay una mejora significativa de 19 a 20 dBs ya que el BER baja considerablemente y se tiene mayores ganancias



**Figura 1-4.** Desempeño 64QAM UC vs NUC

Fuente: Realizado por Mónica F.

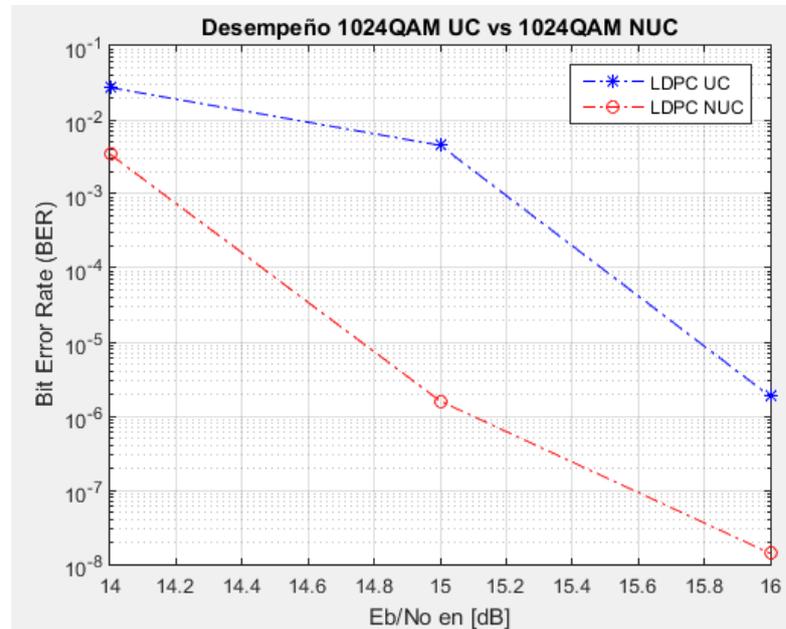
En la figura 2-4, se muestran los resultados del desempeño para un orden de 256 QAM. Como se observa al utilizar constelaciones no uniformes se tiene una caída de la curva más pronunciada, es decir menor número de errores en el receptor; además se tiene una mejora del 99% con relación a las constelaciones uniformes.



**Figura 2-4.** Desempeño 256QAM UC vs NUC

Fuente: Realizado por Mónica F.

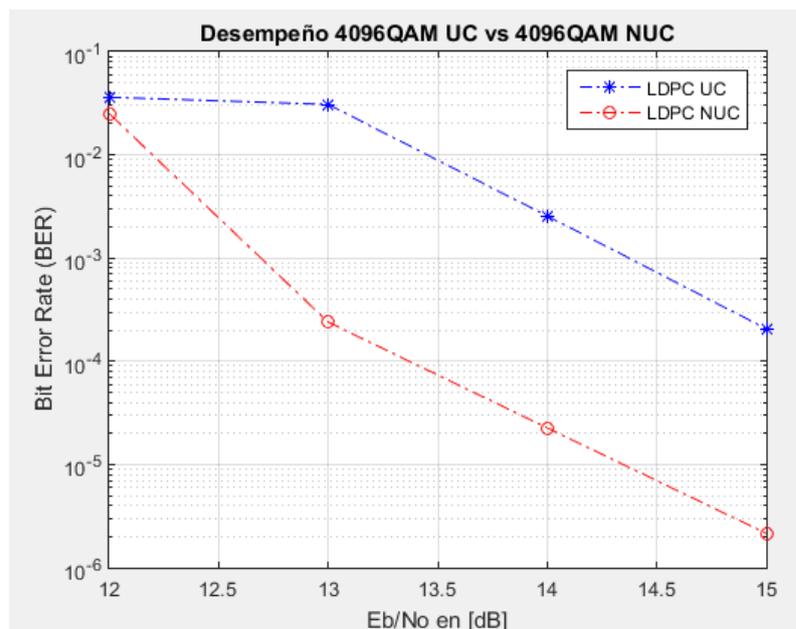
En la figura 3-4, se muestran los resultados de una modulación 1024QAM, en la cual se observa que la curva LDPC NUC tiene una caída de aproximadamente un 99% con relación a la curva de LDPC UC, se analizó con estos valores de SNR ya que se tiene una caída considerable del error de bit.



**Figura 3-4.** Desempeño 1024QAM UC vs NUC

Fuente: Realizado por Mónica F.

En la figura 4-4, se muestran los resultados del desempeño para 4096QAM, en la que se observa una mejora del 81% de la LDPC NUC con la LDPC UC, es decir una reducción considerable del BER



**Figura 4-4.** Desempeño 4096QAM UC vs NUC

Fuente: Realizado por Mónica F.

#### 4.2. Resultados para un canal AWGN con desvanecimiento al utilizar codificación Turbo Log-Map y SOVA utilizando Constelaciones Uniformes versus No Uniformes

En la tabla 2-4., se muestra la métrica de desempeño de un canal AWGN con desvanecimiento, para una modulación de 64 QAM, con codificación Turbo Log\_Map y SOVA tanto con UC y NUC; entre un rango de valores de SNR (8 a 12 dB). Al utilizar constelaciones no uniformes (NUC) se obtienen un BER mucho más pequeño que al utilizar la constelación uniforme; es decir una mejora de 70% al utilizar codificación Log-Map y de 80% con una codificación SOVA.

**Tabla 2-4.** Métricas de desempeño, para 64QAM, codificación Turbo

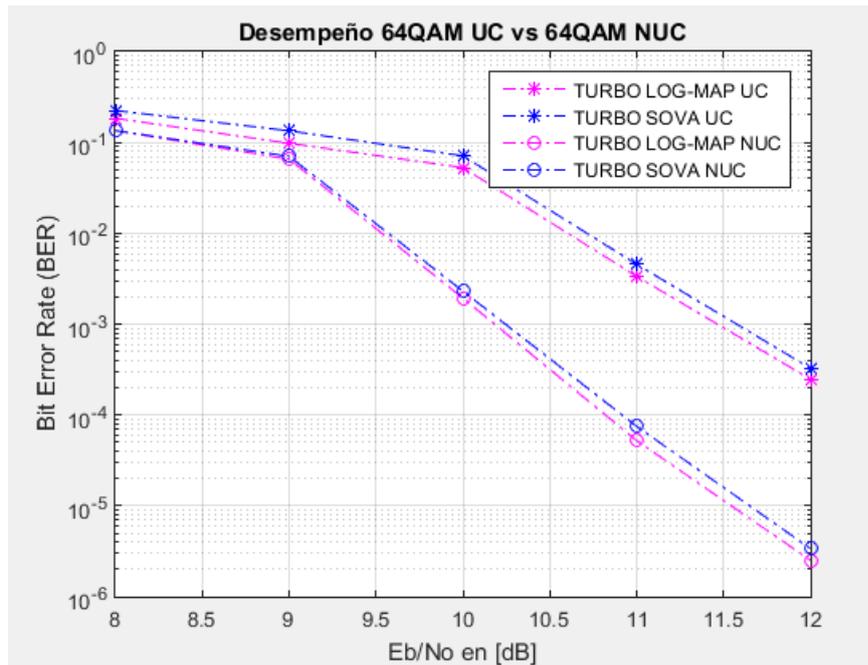
BER				
SNR dB	UC		NUC	
	Turbo Log-Map	Turbo Sova	Turbo Log-Map	Turbo Sova
8	0,18410000	0,22441667	0,13464667	0,13506333
9	0,09781667	0,13418333	0,06524833	0,07015000
10	0,05280000	0,07105000	0,00192967	0,00230433
11	0,00334500	0,00454167	5,2067E-05	7,4883E-05
12	0,00024733	0,00032583	2,5083E-06	3,3717E-06

Fuente: Realizado por Mónica F.

Las Métricas de desempeño, en un canal AWGN con desvanecimiento correspondiente a las constelaciones uniformes y no uniformes de orden 256QAM, 1024QAM y 4096QAM, se encuentran en el Anexo V.

Para visualizar de manera más clara los datos de la tabla 5.2., se gráfica la SNR en función del BER.

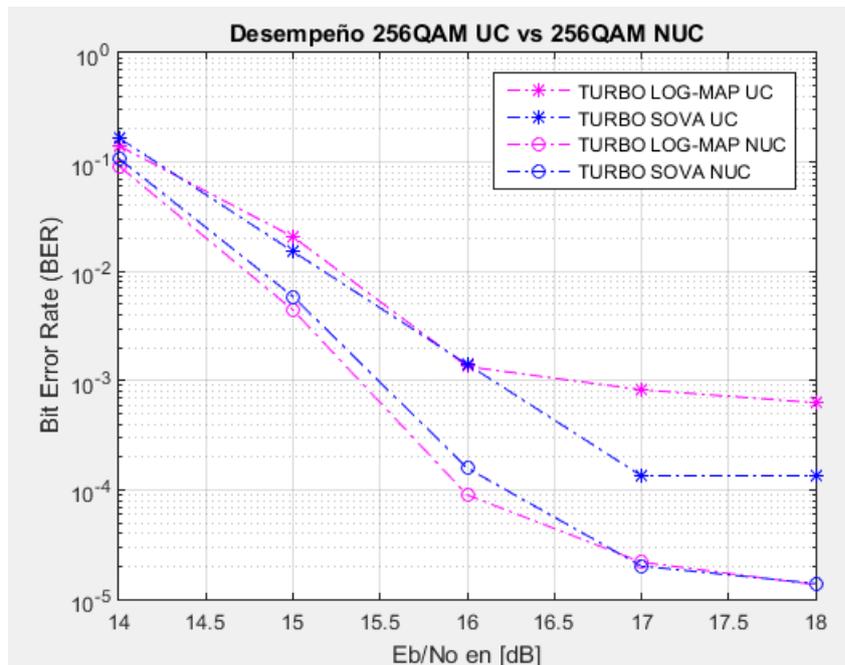
En la figura 5-4, se muestran los resultados del desempeño, para 64QAM en la cual se observa que tanto al utilizar la codificación Log-Map o Sova de tiene una disminución del BER, aproximadamente un 75% con relación a las constelaciones uniformes, además la curva tiene a bajar mientras se sube el SNR.



**Figura 5-4.** Desempeño 64QAM UC vs NUC codificación Turbo

Fuente: Realizado por Mónica F.

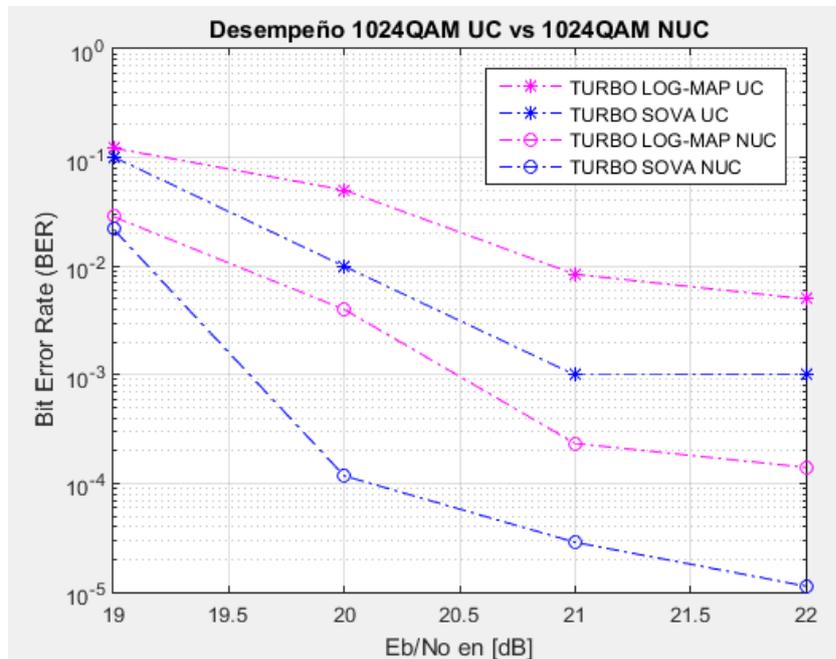
En la figura 6-4, se muestran los resultados del desempeño para un orden de 256 QAM. Aquí también se observa que al utilizar constelaciones no uniformes se tiene una disminución de la probabilidad de error, además se puede notar que las dos codificaciones Turbo se comportan de una manera similar. Aproximadamente se tiene una disminución del 85% de errores en la transmisión.



**Figura 6-4.** Desempeño 256QAM UC vs NUC codificación Turbo

Fuente: Realizado por Mónica F.

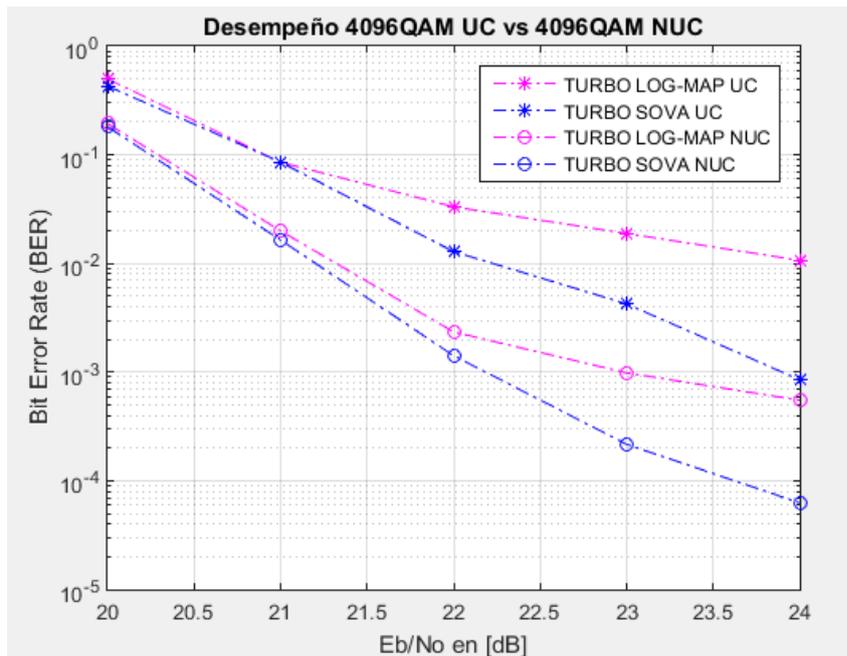
En la figura 7-4, se muestran los resultados de una modulación 1024QAM, en la cual se observa que las curvas con constelaciones no uniformes presentan una mayor ganancia, aunque la caída no es pronunciada si baja el BER mientras se aumenta la SNR



**Figura 7-4.** Desempeño 1024QAM UC vs NUC codificación Turbo

Fuente: Realizado por Mónica F.

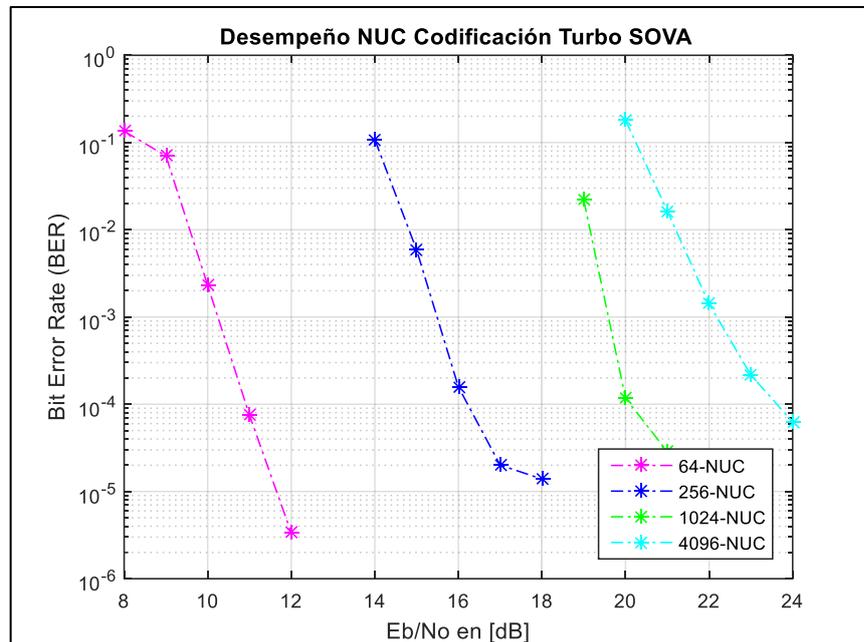
En la figura 8-4, se muestran los resultados del desempeño para 4096QAM, en la que se observa una mejora del 83% de la codificación Turbo NUC con respecto a la codificación Turbo UC, se observa además que la curva es más constante en la caída con respecto al SNR.



**Figura 8-4.** Desempeño 4096QAM UC vs NUC codificación Turbo

Fuente: Realizado por Mónica F.

En la figura 9-4, se muestran una comparación entre las curvas de desempeño obtenidas para 64, 256, 1014 y 4096 NUC-QAM, en una canal AWGN con desvanecimiento en un intervalo de 8 a 24 dB, al utilizar codificación Turbo SOVA.

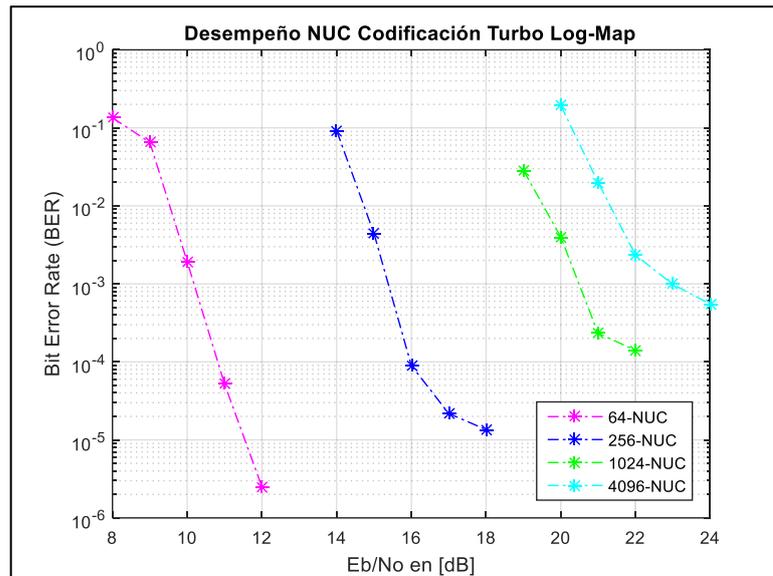


**Figura 9-4.** Codificación SOVA

Fuente: Realizado por Mónica F.

Como se observa en la gráfica, cuando el orden de la modulación es más pequeño el BER empieza a caer en dBs más pequeños (9 - 10 dBs); mientras sube el orden de la modulación, también sube el rango de dBs que empieza a caer la curva; hasta finalmente observar que la curva de orden 4096 caen entre 20 y 22 dBs.

En la figura 10-4, se muestran una comparación entre las curvas de desempeño obtenidas para 64, 256, 1014 y 4096 NUC-QAM, en una canal AWGN con desvanecimiento en un intervalo de 8 a 24 dB, al utilizar codificación Turbo LOG-MAP.

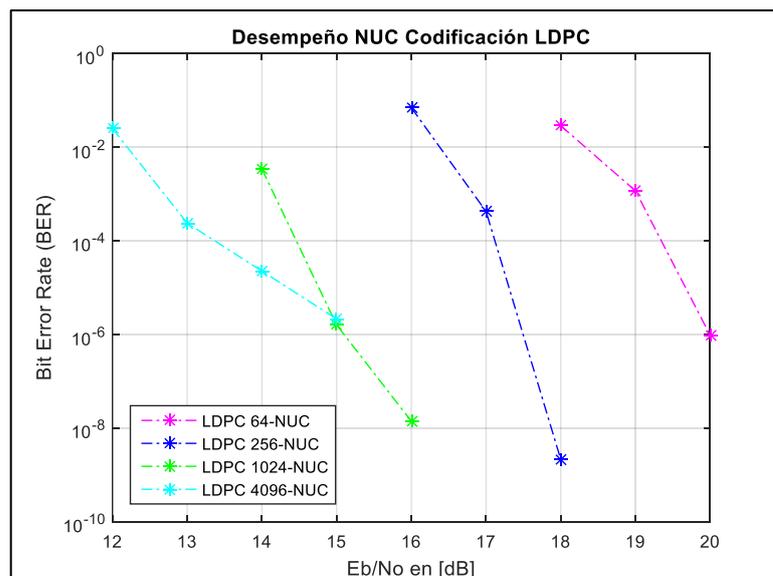


**Figura 10-4.** Codificación LOG-MAP

Fuente: Realizado por Mónica F.

En esta gráfica al igual que en la anterior, se observa que las caídas de la curva dependen directamente del orden de la modulación; es decir entre menor es el orden de modulación la curva cae en un rango menor de dBs. Así, la curva de orden 256 cae en un rango de 14 a 16 dBs hasta estabilizarse.

En la figura 11-4 se observa el desempeño de la codificación LDPC en los cuatro órdenes de modulación aplicados, en esta se observa que entre mayor es el orden de la modulación, la caída de la curva es en valores de dBs más bajos. Es decir, en el orden de modulación de 4096 la caída de la curva es entre 13 y 15 dBs, mientras que en el orden de 64 la curva cae entre 18 y 20 dBs.



**Figura 11-4.** Codificación LDPC

Fuente: Realizado por Mónica F.

### 4.3. Resultados para un canal AWGN con desvanecimiento al utilizar codificación LDPC y Turbo con Constelaciones No Uniformes

En este apartado se muestra el comportamiento de la relación señal a ruido y el BER de cada orden de modulación y las codificaciones LDPC y Log-Map.

En la figura 12-4. se observa que cuando  $M = 64$ , el BER es más pequeño con la codificación Turbo que con la LDPC, es decir en este orden de modulación es mejor codificar con la codificación Turbo.

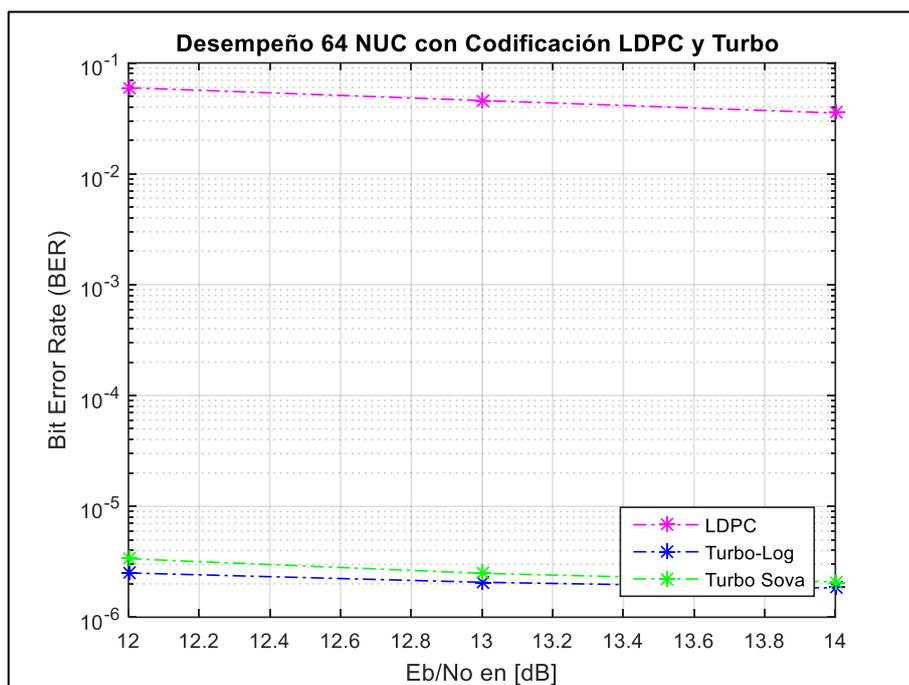
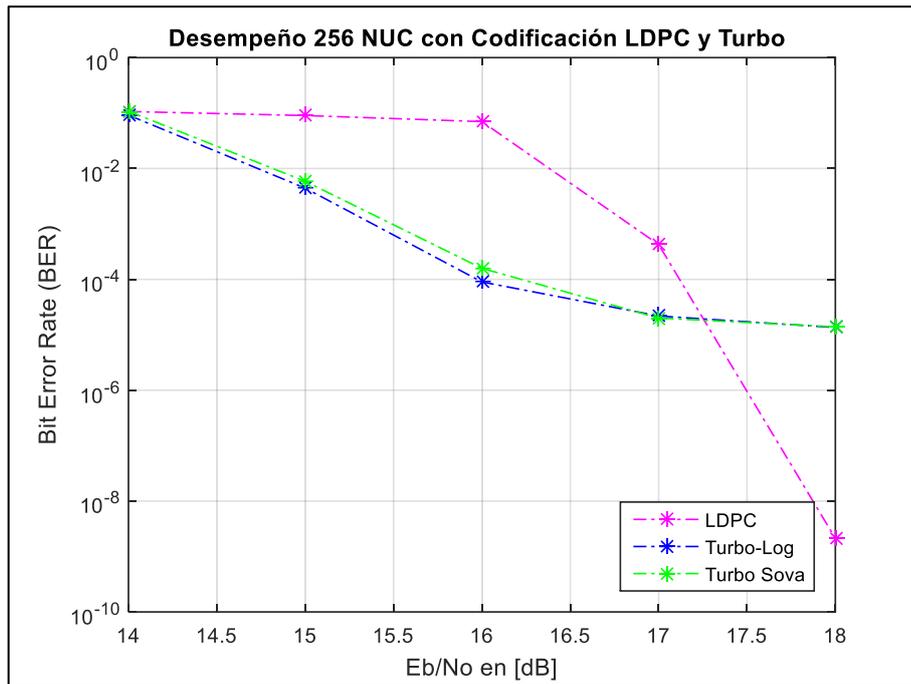


Figura 12-4. Codificación LDPC y Turbo,  $M=64$

Fuente: Realizado por Mónica F.

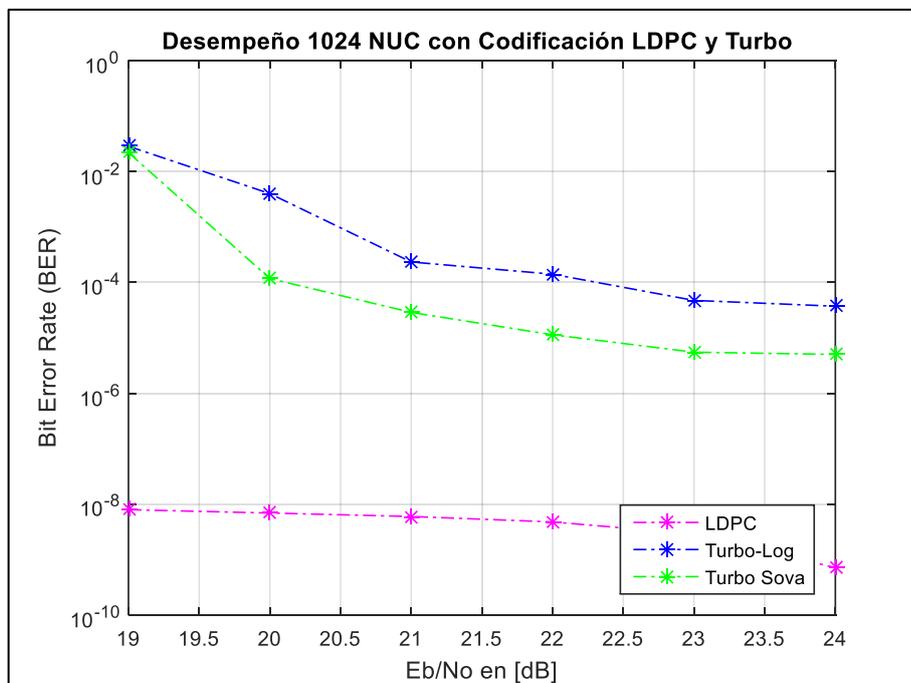
En la figura 13-4. se observa el comportamiento de las curvas de BER/SNR cuando el orden de la modulación es 256, se observa que al utilizar la codificación Turbo el BER disminuye gradualmente y mientras aumenta la SNR este estabiliza; mientras que al utilizar la codificación LDPC se observa que en valores pequeños de SNR el BER está en el orden de  $10^{-1}$ , el BER va a disminuir proporcionalmente hasta estabilizarse en un valor de  $10^{-9}$ ; se obtiene una ganancia de aproximadamente de 1dB con relación a la codificación turbo.



**Figura 13-4.** Codificación LDPC y Turbo, M=256

Fuente: Realizado por Mónica F.

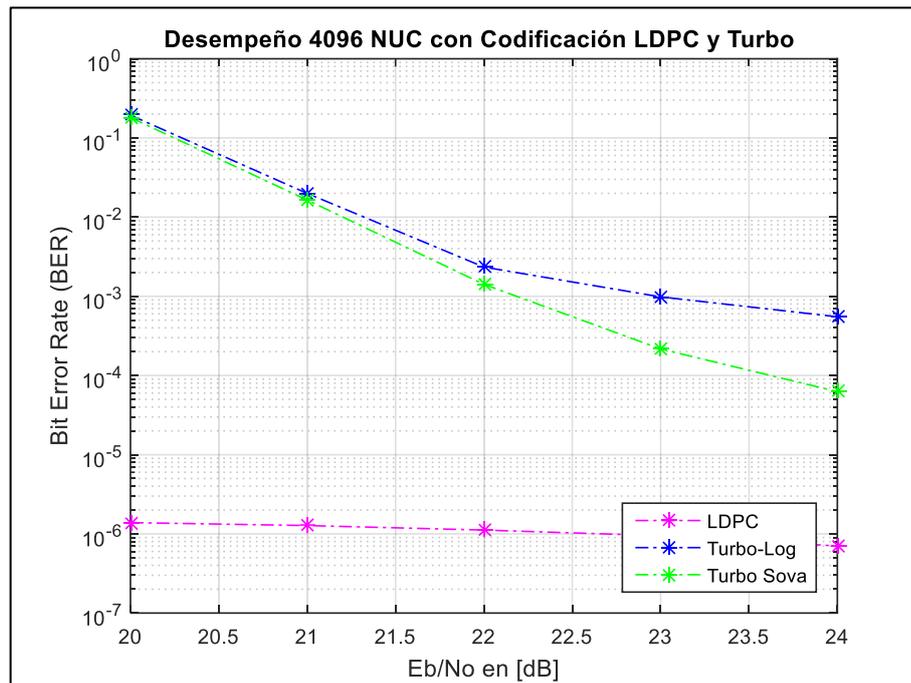
En la figura 14-4. se observa que cuando el orden de modulación es 1024 la codificación LDPC se estabiliza en un BER aproximado de  $10^{-8}$  en el rango de 19 a 24 dBs, mientras que con la codificación turbo la curva se empieza a estabilizar aproximadamente en un BER de  $10^{-4}$ , se tiene menor número de errores al utilizar la codificación LDPC.



**Figura 14-4.** Codificación LDPC y Turbo, M=1024

Fuente: Realizado por Mónica F.

En la figura 15-4. se observa el desempeño de la optimización de canal al utilizar un orden de 4096; aquí se observa un comportamiento parecido al de la gráfica anterior ( $M = 1024$ ), con la codificación LDPC se tiene una probabilidad de error de aproximada de  $10^{-6}$  en la que se estabiliza la curva; mientras que con la codificación Turbo la curva empieza a estabilizarse aproximadamente con una probabilidad de error de  $10^{-4}$ , con lo que se observa claramente que la mejor codificación para este orden de modulación es la LDPC.



**Figura 15-4.** Codificación LDPC y Turbo,  $M=4096$

Fuente: Realizado por Mónica F.

## CONCLUSIONES

La principal interrogante, que surge del tema de la presente investigación, tiene que ver con el desempeño de los tres tipos de codificación, LDPC, SOVA y LOG-MAP. ¿Cuál de ellas se comporta de mejor manera al utilizar modulaciones no uniformes? Para responder esta interrogante, en el Capítulo 4, en el apartado 4.3. se demuestra que cuando  $M = 64$  la mejor codificación es la Turbo ya que la probabilidad de error es menor ( $10^{-1}$ ) y  $M = 256$  la mejor codificación es la Turbo ya que la probabilidad de error es menor ( $10^{-5}$ ) que cuando se utiliza la codificación LDPC.

Mientras sube el orden de la modulación ( $M=1024$  y  $4096$ ) la mejor codificación es la LDPC ya que da probabilidades de error más pequeñas ( $10^{-6}$ ) en comparación con la codificación Turbo.

Al evaluar el comportamiento de la optimización (BER vs SNR) se concluye que al evaluar los diferentes órdenes de modulación en diferentes SNR se tiene que al utilizar la codificación de canal las señales llegan a estabilizarse en probabilidades de error bien pequeñas entre  $10^{-4}$  y  $10^{-8}$ , con lo que se concluye que al utilizar codificación de canal se disminuye considerablemente el número de errores que hay en la transmisión.

La modulación se evaluó en rangos diferentes de SNRs de acuerdo al orden de la misma; debido a que en un determinado SNR la probabilidad de error empieza a ser constante, esto se debe a que los errores en la transmisión bajaron considerablemente.

La modulación no uniforme (NUC) se basa en constelaciones cuyas distancias entre punto y punto son distintas, las mismas que dependen de la capacidad BICM, la SNR y el orden de la modulación; lo que influye directamente en la codificación de canal que también depende de la tasa de bit, de la SNR y del orden de la modulación.

## RECOMENDACIONES

Antes de elegir utilizar una codificación LDPC o Turbo, es necesario observar su comportamiento, en diferentes regiones de SNR, ya que esto ayudará a definir cuál de las codificaciones resulta más conveniente, según los requerimientos del Sistema de Comunicaciones.

Cabe recalcar que la optimización de las constelaciones no uniforme se los realiza bit a bit, y al final se le suma todos los resultados; además al momento de realizar las modulaciones no uniformes se trabajó también en bits mas no en símbolos.

La optimización BICM para modulaciones no uniformes representa una gran ventaja con relación a las convencionales, ya que, al tener distancias diferentes entre los puntos de la constelación, los bits mapeados pueden elegir la menor distancia entre ellos y los puntos de constelación lo que proporciona un mejor mapeado de la información, para que en el receptor sea más fácil reconstruirla.

La codificación LDPC con el algoritmo desarrollado en este proyecto, presenta el BER más bajo en comparación a las otras codificaciones, sin importar el orden de la modulación ni la relación Señal-Ruido; se ha logrado este resultado sacrificando tiempo de ejecución; es decir el tiempo de evaluación de la codificación y decodificación LDPC es bastante más largo, con relación a las otras codificaciones presentadas en el presente proyecto.

## GLOSARIO

**Modulación:** Técnica que se aplica en el transporte de datos sobre ondas portadoras. El proceso de modulación consiste en variar un parámetro que está en la onda portadora en función de las alteraciones de la señal moduladora (Haykin, Communication Systems, 2001)

**Constelación:** Es una representación gráfica (llamada I-Q) de los símbolos digitales recibidos en un periodo de tiempo. Existen distintos tipos de diagramas de constelación para los distintos tipos de modulación (Muela, 2017).

**Codificación:** Consiste en traducir los valores obtenidos en el proceso de cuantificación (ya son valores digitales) al sistema binario, mediante la utilización de una serie de códigos preestablecidos. De este modo al finalizar este procedimiento, se obtiene la deseada señal digital equivalente a la inicial analógica (Mazumdar, Chaturvedi, & Banerjee, 2005).

**Algoritmo:** Un algoritmo es una serie de instrucciones secuenciales que van uno después del otro, que permiten ejecutar acciones o programas. Es crucial para avances tecnológicos como la inteligencia artificial, transmisiones digitales (Jr., 1973)

## BIBLIOGRAFÍA

- Alam, S. (2008). *Introduction to Data Communications*. London.
- Ana Lorena Velásquez; Edwin Mudarra. (2009). *Tecnologías de Radio Digital*. *Universidad Tecnológica de Panamá*, 15.
- Avíles, D. (2017). *ANÁLISIS DE DESEMPEÑO DE MODULACIONES NO UNIFORMES ULTRA-MULTINIVEL*. Sangolqui.
- Barbosa, W. G. (2013). *Estructuras y Características de Desempeño de los Códigos LDPC*. Puebla: INAOE.
- Belen, T. T. (2014). *EVALUACIÓN DEL DESEMPEÑO DE LOS CÓDIGOS DE COMPROBACIÓN DE PARIDAD DE BAJA DENSIDAD (LDPC) EN TRANSMISION DE IMAGENES*. Sangolqui, Quito: ESPE.
- Belkacem Mouhouche; Hakju Lee; Carlos Barjau. (2015). Design of Non Uniform Constellations for Layered Division Multiplexing. *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*.
- Belloch, C. (2012). *Las Tecnologías de la Información y Comunicación en el aprendizaje*. Universidad de Valencia: Depto MIDE.
- Berrou, C., Glavieux, A., & Thitimajshima, P. (1993). *Near shannon limit error-correcting coding and decoding*. Geneva, Switzerland.
- Cándido López García; Manuel Fernández Veiga. (2002). *Teoría de la Información y Codificación*. Pontevedra, España: Universidad de Vigo.
- Constantino Pérez Vega, J. M. (2003). *Fundamentos de televisión analógica y digital*. Cantabria: Servicio de publicaciones de la Universidad de Cantabria.
- Ecuador, G. N. (2017). *Plan Nacional de Desarrollo 2017-2021-Toda una Vida*. Quito, Ecuador: Senplades.
- Escabosa, E. M. (1997). Introducción a la Codificación de canal. *Proyectista del Departamento de Matemática Aplicada y Telemática, UPC*.
- Gallager, R. G. (1962). Low-Density Parity-Check Codes. *IEEE Transactions*.

- Hagenauer, J., & Hoehner, P. (1989). A Viterbi decoding algorithm with soft-decision output and its applications. *IEEE Global Conference on Communications*.
- Haykin, S. (2001). *Communication systems*. New York, Chichester, Toronto: John Wiley & Sons, Inc.
- Joan, G. J., David, G.-B., & Manuel, F. (2016). Optimization and Performance of Non-Uniform Rotated Constellations With Multi-RF Transmission Techniques. *IEEE Broadcast Technology Society*.
- Josefina Castañeda Camacho; Domingo Lara Rodríguez. (2001). Sistemas Personales de Comunicación. *Revista Digital Universitaria UNAM*.
- Jr., G. D. (1973). The viterbi algorithm. *IEEE*.
- Lachlan Michael; David Gómez-Barquero. (2015). Modulation and Coding for ATSC 3.0. *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*.
- Lava, I. d. (2010). *Implementación y evaluación de la codificación LDPC para la transmisión de ficheros en entornos unidireccionales*. España.
- MacKay, D. (2003). Information Theory, Inference and Learning Algorithms. *Cambridge University Press*.
- Marvin K., S., & Mohamed-Slim, A. (2000). *Digital Communication over Fading Channels*. New York: JOHN WILEY & SONS, INC.
- Mazumdar, A., Chaturvedi, A. K., & Banerjee, a. A. (2005). Construction of turbo code interleavers from 3-regular hamiltonian graphs. *arXiv*.
- Mendiburu, A., Miguel-Alonso, J., & Lozano, J. A. (2018). Implementación paralela del algoritmo Belief Propagation. *Facultad de Informática, Universidad del País Vasco*.
- Muela, M. F. (2017). *Non-Uniform Constellations for Next-Generation Digital Terrestrial Broadcast Systems*. Valencia.
- Pisciotta, N. O. (2010). Sistemas ISDB-Tb. Córdoba, Argentina: Universidad Blas Pascal.
- Planificación, R. d. (2017). *Plan Nacional de Desarrollo*. Quito.
- Rao, K. D. (2015). *Channel Coding Techniques for Wireless Communications*. Springer New Delhi Heidelberg New York Dordrecht London: M.N.S. Swamy.

- Sibaja Terán, P., & García Ugalde, F. (2009). Estudio de códigos LDPC y su implementación en plataformas reconfigurables. *México D.F: Universidad Nacional Autónoma de México*.
- Sklar, B. (1999). *Rayleigh Fading Channels*. Boca Raton: CRC Press LLC: Suthan S. Suthersan.
- Stott, J. (2013). CM and BICM limits for rectangular constellations. *Research & Development BBc*.
- Stüber, G. L. (2002). *Principles of Mobile Communication*. New York, Boston, Dordrecht, London, Moscow: KLUWER ACADEMIC PUBLISHERS.
- Utby, H. (2006). *A Survey on Turbo Codes and Recent Developments*. Bergen: University of Bergen, Department of Informatics.
- Valenti, Matthew C.; Jian Sun. (2004). *Handbook of RF and Wireless Technologies*. Editor-in-Chief Lawrence Livermore National Laboratory, USA.
- Vucetic, B. J. (2000). *Turbo Codes Principles and Applications*. New York: Kluwer Academic Publishers.
- Woodard, J. P., & Hanzo, L. (2000). Comparative study of turbo decoding techniques: An overview. *IEEE Transactions on*.
- Xiao, P. (2008). *Introduction to Data Communications*.
- Zoellner, J., & Loghin, N. (2013). Optimization of High-order Non-uniform QAM Constellations. *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2*.

## ANEXOS

### ANEXO A

#### **SIMULADOR DEL DESEMPEÑO DE LAS MODULACIONES NO UNIFORMES AL UTILIZAR CODIFICACIÓN LDPC Y TURBO CÓDIGOS SOBRE UN CANAL AWGN CON DESVANECIMIENTO.**

##### **Funcionamiento del Simulador**

Se desarrolló un Simulador que permite codificar un flujo de bits, con Constelaciones NUC-QAM, de distintos órdenes, optimizados para varios valores de SNR. Los bits mapeados atraviesan canal un canal AWGN con desvanecimiento, y se recuperan en un receptor. Se generan gráficas de los símbolos transmitidos y de los símbolos recibidos. Se calcula la tasa de error de bit y se grafica la relación entre la tasa de error y la relación Señal-Ruido.

El Simulador fue construido en la plataforma MATLAB R2015a, al utilizar varias de sus funciones. Se divide en varios scripts o archivos, con extensión .m, cada uno de ellos simula las diferentes codificaciones. A continuación, se explica cada parte del Simulador.

##### **TRANSMISOR**

En esta parte se cumplen siete pasos que, en conjunto, conforman la etapa del transmisor:

**Configuración inicial:** es donde se determina el número de símbolos que se conformarán a partir del flujo de bits, que se genera posteriormente. De acuerdo al orden de la Modulación, se calcula el número de bits por símbolo y el número total de bits, que emulan un flujo de datos.

```
% Configuración Inicial
nsimb=1000; % Número símbolos
M=64; %Orden de la modulación
k=log2(M); % Número de bits por símbolo
nbits=nsimb*k;% Número total de bits
```

**Figura 1-A.** Configuración inicial de los bits

**Fuente:** Realizado por Mónica F.

**Generación del flujo de bits:** en base a los cálculos realizados en el paso anterior, se genera un flujo aleatorio de bits, que es dividido en grupos más pequeños.

```
% Generación del Flujo de Bits
bits1=round(rand(1,nbits)); %Genera un flujo aleatorio de BITS
l=length(bits1)/k;
a=reshape(bits1,[k,l]);
b=a';
```

**Figura 2-A.** Generación del Flujo de Bits

**Fuente:** Realizado por Mónica F.

**Codificación:** una vez que se tiene el flujo de bits se realiza la codificación, al depender del tipo de codificación que se desee aplicar.

### CODIFICACIÓN LDPC:

Para realizar la codificación LDPC se separó la cadena de bits en paquetes más pequeños (100 bits), para una mejor codificación.

Esta función crea la matriz H, realiza las operaciones de GF2, es decir multiplicación de las matrices A y B para generar la palabra de código y multiplicarla por los datos.

```

%% Función Codificación LDPC
%% Creación matriz H
rows=100;
cols=200;
H=gen_ldpc(rows,cols);%100x200
[newh,rearranged_cols]=rearrange_cols(H);
%% Matrices A y B
for i=1:rows
    for j=1:rows
        A(i,j)=newh(i,j);
    end
end
for i=1:rows
    for j=rows+1:cols
        B(i,j-rows)=newh(i,j);
    end
end
end
%%%%%%%%%%%%%%
d=mul_GF2(inv_GF2(A),B);
c=mul_GF2(d,s);
%%%%%%%%%%%%%%form u'%%%%%%%%%%%%%%
u1=c;
for i=rows+1:cols
    u1(i)=s(i-rows);
end
%% Reordenar u
u=reorder_bits(u1,rearranged_cols);
%% Mensaje a transmitir
c=u;

```

**Figura 3.A.** Función codificación LDPC

**Fuente:** Realizado por Mónica F.

En este proceso de codificación también participan otras funciones, como son:

**gen\_ldpc:** esta función crea la matriz H con un número de unos en filas y en columnas específicas y distribuidas uniformemente.

```

bits_per_col=3;
row_flag=zeros(1,rows);
parity_check=zeros(rows,cols);
for i=1:cols
    a=randperm(rows);
    for j=1:bits_per_col
        parity_check(a(j),i)=1;
        row_flag(a(j))=row_flag(a(j))+1;
    end
end

```

```

    end
end
max_ones_per_row=ceil(cols*bits_per_col/rows);
for i=1:rows
    if row_flag(i)==1
        j=unidrnd(cols);
        while parity_check(i,j)==1
            j=unidrnd(cols);
        end
        parity_check(i,j)=1;
        row_flag(i)=row_flag(i)+1;
    end
    if row_flag(i)==0
        for k=1:2
            j=unidrnd(cols);
            while parity_check(i,j)==1
                j=unidrnd(cols);
            end
            parity_check(i,j)=1;
            row_flag(i)=row_flag(i)+1;
        end
    end
end
end
H=parity_check; %Matriz H

```

**Figura 4-A.** Creación de la Matriz H

**Fuente:** Realizado por Mónica F.

**mul\_GF2:** esta función realiza operaciones de suma y producto de la matriz H con la cadena de bits.

```

dim=size(A);
m=dim(1); %número de filas de la primera matriz
n=dim(2); %número de columnas lera y número de filas de la 2da matriz
dim=size(B);
p=dim(2); %número de columnas de la segunda matriz
for i=1:m
    for j=1:p
        temp1=A(i,1:n);
        temp2=B(1:n,j);
        prod=temp1.*(temp2. ');
        sum1=0;
        for k=1:n
            sum1=xor(sum1,prod(k));
        end
        c(i,j)=sum1;
    end
end
end

```

**Figura 5-A.** Función multiplicación de dos matrices

**Fuente:** Realizado por Mónica F.

**reorder\_bits:** reordena los bits codificados

```
dim=size(rearranged_cols);
rows=dim(2);
for i=rows:-1:1
    if rearranged_cols(i)~=0
        temp=c(i);
        c(i)=c(rearranged_cols(i));
        c(rearranged_cols(i))=temp;
    end
end
```

**Figura 6-A.** Reordena los bits codificados

**Fuente:** Realizado por Mónica F.

## CODIFICACIÓN TURBO

Para la codificación turbo se dividió la cadena de bits en paquetes más pequeños (1000 bits) para evitar errores de memoria y que el tiempo de ejecución sea menor.

En la codificación turbo se crea una matriz generadora  $g$  y una matriz  $\alpha$  que nos ayudaran a crear el código.

```
g = [1 1 1; 1 0 1];
[n,K] = size(g);
m = K - 1;
nstates = 2^m;
%Calculo de sigma
en = 10^(EbNodB(1)/10);
L_c = 4*1*en*rate;
sigma1 = 1/sqrt(2*rate*en);
x = bits2(1,1:(L_total-m));
[temp, alpha] = sort(rand(1,L_total));
```

**Figura 7-A.** Matriz generadora

**Fuente:** Realizado por Mónica F.

**codificadorTURBO:** esta función crea los codificadores 1 (RSC 1), el intercalador y el codificador 2 (RSC 2); los entrelaza y se tiene la palabra de código.

```
n,K] = size(g);
m = K - 1;
L_info = length(x);
L_total = L_info + m;
input = x;
output1 = rsc_encode(g,input,1);
y(1,:) = output1(1:2:2*L_total);
y(2,:) = output1(2:2:2*L_total);
% entrelaza la entrada con el decodificador 2
for i = 1:L_total
    input1(1,i) = y(1,alpha(i));
end
output2 = rsc_encode(g, input1(1,1:L_total), -1 );
y(3,:) = output2(2:2:2*L_total);
```

```

if puncture > 0    % preforado tasa 1/3
    for i = 1:L_total
        for j = 1:3
            en_output(1,3*(i-1)+j) = y(j,i);
        end
    end
else                % preforado tasa 1/2
    for i=1:L_total
        en_output(1,n*(i-1)+1) = y(1,i);
        if rem(i,2)
            % verificación de bits RSC1
            en_output(1,n*i) = y(2,i);
        else
            % verificación de bits RSC2
            en_output(1,n*i) = y(3,i);
        end
    end
end
end

```

**Figura 8-A.** Codificador Turbo

**Fuente:** Realizado por Mónica F.

**rsc\_encode:** crea el codificador RSC para la codificación

```

[n,K] = size(g);
m = K - 1;
if terminated>0
    L_info = length(x);
    L_total = L_info + m;
else
    L_total = length(x);
    L_info = L_total - m;
end
state = zeros(1,m);
% genera la palabra de código
for i = 1:L_total
    if terminated<0 | (terminated>0 & i<=L_info)
        d_k = x(1,i);
    elseif terminated>0 & i>L_info
        d_k = rem( g(1,2:K)*state', 2 );
    end
    a_k = rem( g(1,:)*[d_k state]', 2 );
    [output_bits, state] = encode_bit(g, a_k, state);
    % salida sistemática, bit de entrada
    output_bits(1,1) = d_k;
    y(n*(i-1)+1:n*i) = output_bits;
end

```

**Figura 9-A.** Codificador RSC

**Fuente:** Realizado por Mónica F.

## MODULACIÓN

La función de modulación se encarga de modular los bits codificados con modulación no uniforme; para la modulación se crea una constelación no uniforme, se mapea y se modulan los bits.

En esta función ingresan los bits en binario y en decimal, además de la relación señal-ruido. Se realiza esta configuración inicial antes de la modulación ya que los bits se los envía en paquetes más pequeños (100 o 1000 bits).

```
inf_cod_dec=(bi2de(cod_LDPC))';
inf_cod_dec=inf_cod_dec+1;
Mpam=sqrt(M); % Orden de PAM
kpam=log2(Mpam); % Numero de bits por símbolo, en PAM
st=inf_cod_dec; % Información en decimal
snrdb=EbNodB; %Relación Señal/Ruido
```

**Figura 10-A.** Parámetros para a modulación

Fuente: Realizado por Mónica F.

**Constelaciones NUQAM:** en esta parte se encuentran definidas las posiciones positivas correspondientes a uno de los ejes PAM de la Constelación, para cada valor seleccionable de SNR. Se realiza una selección automática del conjunto correspondiente de posiciones PAM, de acuerdo al valor de SNR seleccionado. Para el ejemplo se toma SNR=10db y M=64.

```
snrdb==10
opt_64_posi=[1 1,8112 3,9386 6,1594]; % Para 10dB
```

**Figura 11-A.** Constelación 64 NUC-QAM

Fuente: Realizado por Mónica F.

**Mapeo Gray PAM:** en base al conjunto de posiciones PAM seleccionado anteriormente, se obtiene las posiciones negativas correspondientes para luego conforman un eje PAM Real y otro eje PAM Imaginario. Estos ejes sirven para construir la constelación.

```
% MAPEO GRAY PAM
opt_64_posi=opt_64_posi([3 4 2 1]); % parte positiva
opt_64_nega=-1*opt_64_posi; %parte negativa
opt_64_pamr=[opt_64_nega opt_64_posi]; % PAM parte Real
opt_64_pami=j*opt_64_pamr; % PAM Parte Imaginaria
```

**Figura 12-A.** Mapeo Gray 8 PAM

Fuente: Realizado por Mónica F.

**Mapeo Gray NUC-QAM:** se construye la Constelación NUC-QAM, en la que cada uno de los símbolos está representado por una coordenada obtenida en pasos anteriores.

```
mapeo_gray=[];  
for r=1:sqrt(M)  
    for i=1:sqrt(M)  
        mapeo_gray=[mapeo_gray opt_64_pamr(r)+opt_64_pami(i)];  
    end  
end  
m=mapeo_gray(st); % Símbolos 64-NUC-QAM, a transmitir
```

**Figura 13-A.** Mapeo gray 64 NUC-QAM

**Fuente:** Realizado por Mónica F.

## CANAL DE COMUNICACIÓN

Una vez realizada la codificación y la modulación se termina la etapa de transmisión.

Ahora se simula el canal de transmisión, al cual se le añade ruido AWGN y además el desvanecimiento a la cadena de bits modulada.

Primeramente, se calcula el valor de sigma en función del número de símbolos de la modulación, se crea la cadena aleatoria para simular el ruido blando y seguidamente se le suma a la señal.

```
% DESVANECIMIENTO RAYLEIGH  
sigmar=1/(sqrt(2));  
al=normrnd(0,sigmar,1,nsimb_mod)+j*normrnd(0,sigmar,1,nsimb_mod);  
alpha=abs(al);  
y=alpha.*inf_MOD+ruido;
```

**Figura 14-A.** Simulación Canal AWGN con Desvanecimiento

**Fuente:** Realizado por Mónica F.

## RECEPTOR

En la etapa de recepción se realizan tres funciones principales: la demodulación, la decodificación y la extracción del mensaje.

Una vez obtenida la señal que atravesó el Canal, es necesario decidir en qué posición debe ubicarse cada símbolo. Como los símbolos pasaron por el canal de transmisión, los mismos cambiaron de posición en la constelación; de ahí surge la necesidad de decidir cuál es la mejor posición de los símbolos en la constelación.

En este caso se implementó un decisor en base a distancias, lo que hace es comparar la posición del símbolo recibido con la posición del símbolo original y elige la distancia cuadrática mínima; cada posición se disminuye en uno de forma inversa a como se realizó en el transmisor.

```

% DECISION EN BASE A DISTANCIAS
INF_AWGN_DESV=INF_AWGN_DESV./alpha; % Ecualizador, para
Desvanecimiento
re1= repmat(INF_AWGN_DESV.',1,M);
re2= repmat(mapeo_gray,nsimb_mod,1);
d2=(abs(re1-re2)).^2;
d3=d2';
[x pos_sr]=min(d3);
sr=pos_sr-1;
% DEMAPEO NUQAM
sr_bin=de2bi(sr,k);
inf_DEMOD=sr_bin;
inf_demod_dec=sr;

```

**Figura 15-A.** Demodulador (decisión en base a distancias)

**Fuente:** Realizado por Mónica F.

## DECODIFICACIÓN

En la etapa de decodificación se debe elegir qué tipo de decodificación se realiza; tomar en cuenta codificación realizada en el transmisor.

**Decodificador LDPC:** esta función realiza el proceso inverso a la codificación, para obtener los bits originales.

```

vhat=decode_ldpc(inf_DEMOD,EbNodB,1,H,scale); %since amp=1;
vhat1 = extract_mesage(vhat,rearranged_cols);
vhat2=vhat1';
inf_RECIBIDA=vhat2;

```

**Figura 16-A.** Configuración inicial para decodificar

**Fuente:** Realizado por Mónica F.

En esta función intervienen dos funciones principales para la decodificación.

**decode\_ldpc:** Esta función decodifica los bits, primero en las filas y después en las columnas, para finalmente obtener los bits decodificados.

```

dim=size(h);
rows=dim(1);
cols=dim(2);
vhat(1,1:cols)=0;
zero(1,1:rows)=0;
s=struct('qmn0',0,'qmn1',0,'dqmn',0,'rmn0',0,'rmn1',0,'qn0',0,'qn1',
0,'alphamn',1);
%Probabilidades previas
for i=1:length(rx_waveform)
    p11(i)=1/(1+exp(-2*amp*rx_waveform(i)*scale(i)/(No/2)));
    p10(i)=1-p11(i);
end
%inicialización
for i= 1:rows

```

```

    for j=1:cols
        newh(i,j)=s;
        if h(i,j)==1
            newh(i,j).qmn0=p10(j);
            newh(i,j).qmn1=p11(j);
        end
    end
end
end
%% Iteraciones
for iteration=1:20
    %% Paso Horizontal
    for i=1:rows
        ones_in_row=0;
        index=1;
        for j=1:cols
            if h(i,j)==1
                ones_in_row(index)=j;
                index=index+1;
                newh(i,j).dqmn=newh(i,j).qmn0 -newh(i,j).qmn1;
            end
        end
        for j=1:index-1
            drmn=1;
            for k=1:index-1
                if k~=j
                    drmn=drmn*newh(i,ones_in_row(k)).dqmn;
                end
            end
            newh(i,ones_in_row(j)).rmn0=(1+drmn)/2;
            newh(i,ones_in_row(j)).rmn1=(1-drmn)/2;
        end
    end
    %% Paso Vertical
    for j=1:cols
        ones_in_col=0;
        index=1;
        for i=1:rows
            if h(i,j)==1
                ones_in_col(index)=i;
                index=index+1;
            end
        end
        for i=1:index-1
            prod_rmn0=1;
            prod_rmn1=1;
            for k=1:index-1
                if k~=i
                    prod_rmn0=prod_rmn0*newh(ones_in_col(k),j).rmn0;
                    prod_rmn1=prod_rmn1*newh(ones_in_col(k),j).rmn1;
                end
            end
            const1=p10(j)*prod_rmn0;
            const2=p11(j)*prod_rmn1;
            newh(ones_in_col(i),j).alphan=1/(const1 + const2);
            newh(ones_in_col(i),j).qmn0=newh(ones_in_col(i),j).alphan*const1;
            newh(ones_in_col(i),j).qmn1=newh(ones_in_col(i),j).alphan*const2;
            const3=const1*newh(ones_in_col(i),j).rmn0;
            const4=const2*newh(ones_in_col(i),j).rmn1;
            alpha_n=1/(const3+const4);
            newh(ones_in_col(i),j).qn0=alpha_n*const3;
        end
    end
end
end

```

```

        newh(ones_in_col(i),j).qn0=alpha_n*const3;
        newh(ones_in_col(i),j).qn1=alpha_n*const4;
        %tentative decoding
        if newh(ones_in_col(i),j).qn1>0.5
            vhat(j)=1;
        else
            vhat(j)=0;
        end
    end
end
if mul_GF2(vhat,h.')==zero
    break;
end
end
end

```

**Figura 17-A.** Decodificador LDPC (Probabilidades logarítmicas)

**Fuente:** Realizado por Mónica F.

**extract\_message:** esta función reorganiza los bits para extraerlos y tener finalmente los bits transmitidos.

```

dim=size(rearranged_cols);
rows=dim(2);
dim=size(c);
cols=dim(2);
for i=1:rows
    if rearranged_cols(i)~=0
        temp=c(i);
        c(i)=c(rearranged_cols(i));
        c(rearranged_cols(i))=temp;
    end
end
u=c(rows+1:cols);

```

**Figura 18-A.** Extracción de los bits recibidos

**Fuente:** Realizado por Mónica F.

**Decodificador Turbo:** la función de decodificación realiza un escalado de los bits y ciertas operaciones iniciales antes de aplicar el decodificador SOVA o LOG-MAP.

Como en el transmisor se creó dos codificadores RSC, en el receptor se realiza el proceso inverso, es decir en cada decodificador RSC se aplica el decodificador SOVA o LOG-MAP.

```

i=1;
[aa bb]=size(inf_DEMOD);
for k=1:1:aa
    for j=1:1:bb
        if inf_DEMOD(k,j)==1
            demodu(i)=inf_DEMOD(k,j);
        else
            demodu(i)=-1;
        end
        i=i+1;
    end
end
end

```

```

r = demodu+sigma1*randn(1,L_total*(2+puncture)); % received bits
yk = demultiplex(r,alpha,puncture); % demultiplexar para obtener la
entrada para el decodificador 1 y 2
rec_s = 0.5*L_c*yk; % Escalado de los bits recibidos
% Inicializar información intrínseca
L_e(1:L_total) = zeros(1,L_total);
for iter = 1:niter
% Decodificador UNO
L_a(alpha) = L_e; % información a priori
if dec_alg == 0
%rec_s1=rec_s(1,:); ind_dec=1;
L_all = MAP(rec_s(1,:), g, L_a, 1);

[L_all_rec]=correc_LOGMAP(L_all,EbNodB,M);
L_all=L_all_rec;
else
L_all = SOVA(rec_s(1,:), g, L_a, 1);
end
L_e = L_all - 2*rec_s(1,1:2:2*L_total) - L_a;
% Decodificador DOS
L_a = L_e(alpha); % información a priori
if dec_alg == 0
%rec_s = 0.5*L_c*yk; rec_s1=rec_s(2,:); ind_dec=2;
L_all = MAP(rec_s(2,:), g, L_a, 2);
[L_all_rec]=correc_LOGMAP(L_all,EbNodB,M);
L_all=L_all_rec;
else
L_all = SOVA(rec_s(2,:), g, L_a, 2);
end
%rec_s = 0.5*L_c*yk;
L_e = L_all - 2*rec_s(2,1:2:2*L_total) - L_a;
aux=find(L_all==0);
L_all(aux)=-1;
% Estimación de los bits de información
xhat(alpha) = (sign(L_all)+1)/2; %linea original
end
inf_RECIBIDA=xhat;

```

**Figura 19-A.** Decodificador Turbo

**Fuente:** Realizado por Mónica F.

**Algoritmo SOVA:** esta función aplica la decodificación SOVA, es decir el algoritmo de Vitervi de salida suave, esta función añade una cola a los bits y calcula las métricas de la mejor ruta. Además, crea los estados, es decir el estado presente y el estado anterior para realizar las operaciones en cada uno de los decodificadores.

```

L_total = length(L_a);
[n,K] = size(g);
m = K - 1;
nstates = 2^m;
Infty = 1e10;
delta = 30;
[next_out, next_state, last_out, last_state] = trellis(g);
for t=1:L_total+1
for state=1:nstates
path_metric(state,t) = -Infty;

```

```

    end
end
path_metric(1,1) = 0;
for t=1:L_total
    y = rec_s(2*t-1:2*t);
    for state=1:nstates
        sym0 = last_out(state,1:2);
        sym1 = last_out(state,3:4);
        state0 = last_state(state,1);
        state1 = last_state(state,2);
        Mk0 = y*sym0' - L_a(t)/2 + path_metric(state0,t);
        Mk1 = y*sym1' + L_a(t)/2 + path_metric(state1,t);
        if Mk0>Mk1
            path_metric(state,t+1)=Mk0;
            Mdiff(state,t+1) = Mk0 - Mk1;
            prev_bit(state, t+1) = 0;
        else
            path_metric(state,t+1)=Mk1;
            Mdiff(state,t+1) = Mk1 - Mk0;
            prev_bit(state,t+1) = 1;
        end
    end
end

if ind_dec == 1
    mlstate(L_total+1) = 1;
else
    mlstate(L_total+1) = find(
path_metric(:,L_total+1)==max(path_metric(:,L_total+1)) );
end

for t=L_total:-1:1
    est(t) = prev_bit(mlstate(t+1),t+1);
    mlstate(t) = last_state(mlstate(t+1), est(t)+1);
end

for t=1:L_total
    llr = Inf;
    for i=0:delta
        if t+i<L_total+1
            bit = 1-est(t+i);
            temp_state = last_state(mlstate(t+i+1), bit+1);
            for j=i-1:-1:0
                bit = prev_bit(temp_state,t+j+1);
                temp_state = last_state(temp_state, bit+1);
            end
            if bit~=est(t)
                llr = min( llr,Mdiff(mlstate(t+i+1), t+i+1) );
            end
        end
    end
    L_all(t) = (2*est(t) - 1) * llr;
end

```

**Figura 20-A.** Algoritmo de Decodificación SOVA

**Fuente:** Realizado por Mónica F.

**Algoritmo LOG-MAP:** esta función aplica la decodificación LOG-MAP, en esta función se encuentran los valores de alpha, beta, gamma que son las variables que se utilizan para calcular la probabilidad logarítmica de los bits y obtener una salida suave de los mismos.

```

L_total = length(rec_s)/2;
[n,K] = size(g);
m = K - 1;
nstates = 2^m;
[next_out, next_state, last_out, last_state] = trellis(g);
Infty = 1e10;
Alpha(1,1) = 0; % Inicialización de Alpha
Alpha(1,2:nstates) = -Infty*ones(1,nstates-1);
if ind_dec==1
    Beta(L_total,1) = 0; % Inicialización of Beta
    Beta(L_total,2:nstates) = -Infty*ones(1,nstates-1);
elseif ind_dec==2
    Beta(L_total,1:nstates) = zeros(1,nstates);
end
% Calculo de Alpha
for k = 2:L_total+1
    for state2 = 1:nstates
        gamma = -Infty*ones(1,nstates);
        gamma(last_state(state2,1)) = (-rec_s(2*k-3)+rec_s(2*k-
2)*last_out(state2,2))....
            -log(1+exp(L_a(k-1)));

gamma(last_state(state2,2)) = (rec_s(2*k-3)+rec_s(2*k-
2)*last_out(state2,4))....
            +L_a(k-1)-log(1+exp(L_a(k-1)));
        if(sum(exp(gamma+Alpha(k-1,:)))<1e-300)%300
            Alpha(k,state2)=-Infty;
        else
            Alpha(k,state2) = log( sum( exp( gamma+Alpha(k-1,:) ) ) );
        end
    end
    tempmax(k) = max(Alpha(k,:));
    Alpha(k,:) = Alpha(k,:) - tempmax(k);
    %% alpha, no puede ser cero
    aux1=find(Alpha==0);
    Alpha(aux1)=-1.99e-320;
end
% Calculo Beta
for k = L_total-1:-1:1
    for statel = 1:nstates
        gamma = -Infty*ones(1,nstates);
        gamma(next_state(statel,1)) = (-
rec_s(2*k+1)+rec_s(2*k+2)*next_out(statel,2))....
            -log(1+exp(L_a(k+1)));
        gamma(next_state(statel,2)) =
(rec_s(2*k+1)+rec_s(2*k+2)*next_out(statel,4))....
            +L_a(k+1)-log(1+exp(L_a(k+1)));
        if(sum(exp(gamma+Beta(k+1,:)))<1e-300)%300
            Beta(k,statel)=-Infty;
        else
            if log(sum(exp(gamma+Beta(k+1,:))))==0
                Beta(k,statel)=1.99e-320
            Else

```

```

        Beta(k, state1) = log(sum(exp(gamma+Beta(k+1,:))));
    end
    %Beta(k, state1) = log(sum(exp(gamma+Beta(k+1,:))));
end
end
Beta1(k,:) = Beta(k,:) - tempmax(k+1);
Beta(k,:) = Beta(k,:) - tempmax(k+1);
end
for k = 1:L_total
    for state2 = 1:nstates
        gamma0 = (-rec_s(2*k-1)+rec_s(2*k)*last_out(state2,2))....
            -log(1+exp(L_a(k)));
        gamma1 = (rec_s(2*k-1)+rec_s(2*k)*last_out(state2,4))...
            +L_a(k)-log(1+exp(L_a(k)));
        temp0(state2) = exp(gamma0 + Alpha(k,last_state(state2,1)) +
Beta(k, state2));
        temp1(state2) = exp(gamma1 + Alpha(k,last_state(state2,2)) +
Beta(k, state2));
    end
    if log(sum(temp1))== -Inf %0
        if log(sum(temp0))== -Inf
            L_all(k) = (9e-320);
        else
            L_all(k) = (9e-320) - log(sum(temp0));
        end
    else
        if log(sum(temp0))== -Inf
            L_all(k) = log(sum(temp1)) - (9e-320);
        else
            L_all(k) = log(sum(temp1)) - log(sum(temp0));
        end
    end
end
end

```

**Figura 21-A.** Algoritmo de Decodificación LOG-MAP

**Fuente:** Realizado por Mónica F.

Finalmente, después de la etapa de decodificación se obtienen los bits en el receptor.

**Calculo del BER:** Al término de todo este proceso se calculó el ver, aquí se realiza una comparación de los bits enviados y los bits recuperados en el receptor.

```

%% C A L C U L O   D E L   B E R
sr_bin_2=infRecib';
sr_bin_3=reshape(sr_bin_2,[1,nbits]);
[error, ber] = biterr(bits1,sr_bin_3);

```

**Figura 22-A.** Calculo del BER

**Fuente:** Realizado por Mónica F.

## ANEXO B

### PARÁMETROS DE OPTIMIZACIÓN PARA AWGN CON DESVANECIMIENTO

En esta sección se encuentran tabuladas todas las distancias entre bits de cada una de las modulaciones, con un rango de SNR amplio.

**Tabla 1-A.** Ubicaciones optimizadas, canales AWGN con desvanecimiento, 256 NUC-QAM

256 NUC-QAM							
SNR(dB)	AWGN Desvanecimiento						
	d1	d2	d3	d4	d5	d6	d7
7	1,07620	1,23170	1,14410	3,23900	3,06270	3,65210	5,02970
8	1,05370	1,35850	1,28630	3,28910	3,30050	4,27420	5,60810
9	1,03420	1,52250	1,46890	3,46020	3,58070	4,82420	6,27100
10	1,01470	1,74050	1,71110	3,70190	3,91820	5,39450	7,02820
11	0,99920	2,00630	2,00370	3,99660	4,30560	5,99700	7,84890
12	0,99180	2,26150	2,28730	4,27610	4,68710	6,54830	8,61070
13	0,99220	2,46540	2,52750	4,50130	5,03120	7,00340	9,24240
14	0,99890	2,60860	2,73070	4,66920	5,35760	7,38280	9,76120
15	1,02590	2,71350	2,94430	4,83390	5,74130	7,78870	10,29380
16	1,11550	2,84190	3,26590	5,13830	6,30820	8,41960	11,08790
17	1,39630	3,17950	3,96750	5,92810	7,43530	9,78250	12,79270
18	1,85340	3,71750	4,92440	7,04750	8,91350	11,57260	15,00090
19	2,22820	4,15410	5,67600	7,90720	10,02920	12,88640	16,56320
20	2,48910	4,46080	6,19700	8,48620	10,76940	13,72170	17,50170
21	2,66190	4,66400	6,53860	8,85210	11,22480	14,20180	17,98940
22	2,77290	4,79360	6,75310	9,06850	11,48230	14,44020	18,17640
23	2,84370	4,87580	6,88570	9,19060	11,61570	14,53260	18,19260
24	2,88900	4,92570	6,96390	9,25300	11,67320	14,54210	18,11720

Fuente: Realizado por Mónica F.

**Tabla 2-A.** Ubicaciones optimizadas, canales AWGN con desvanecimiento, 1024NUC-QAM

1024 NUC-QAM															
SNR(dB)	AWGN Desvanecimiento														
	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15
6	1,0458	1,1021	1,0508	1,1052	1,1571	1,1021	1,0521	3,4859	3,0858	2,8799	3,0645	3,4956	3,0965	3,5501	5,1430
7	1,0252	1,0930	1,0681	1,2121	1,2411	1,1634	1,1359	3,2744	3,2456	3,0861	3,0951	3,7397	3,6125	4,3656	5,6760
8	1,0251	1,0684	1,0408	1,3282	1,3645	1,3105	1,2768	3,2804	3,3136	3,3236	3,2940	4,4647	4,2570	4,7788	6,3035
9	1,0106	1,0421	1,0311	1,4929	1,5094	1,4612	1,4444	3,4275	3,4906	3,5804	3,5187	4,8383	4,7767	5,4497	6,9690
10	1,0022	1,0226	1,0201	1,7209	1,7258	1,6899	1,6851	3,6722	3,7323	3,9166	3,8461	5,3122	5,3692	6,2546	7,8139
11	0,9999	1,0034	1,0037	1,9830	1,9826	1,9706	1,9712	3,9540	3,9988	4,2842	4,2212	5,8215	5,9729	7,0898	8,7306
12	0,9993	0,9938	0,9938	2,2325	2,2312	2,2471	2,2486	4,2289	4,2550	4,6461	4,6016	6,3089	6,5416	7,8656	9,6322
13	0,9997	0,9916	0,9911	2,4349	2,4346	2,4860	2,4864	4,4576	4,4646	4,9706	4,9552	6,7222	7,0327	8,5382	10,4411
14	1,0006	0,9966	0,9957	2,5818	2,5854	2,6894	2,6871	4,6338	4,6222	5,2658	5,2889	7,0625	7,4575	9,1068	11,1400
15	1,0003	1,0149	1,0158	2,6848	2,6903	2,8820	2,8747	4,7815	4,7619	5,5779	5,6434	7,3854	7,8797	9,6350	11,7874
16	0,9986	1,0776	1,0779	2,7905	2,7982	3,1454	3,1351	5,0115	4,9912	6,0174	6,1340	7,8549	8,4919	10,3440	12,6414
17	0,9994	1,2742	1,2749	3,0323	3,0371	3,6813	3,6718	5,5854	5,5804	6,8559	7,0475	8,8436	9,7042	11,7270	14,2894
18	0,9989	1,6871	1,6874	3,5222	3,5220	4,5789	4,5753	6,6178	6,6403	8,1969	8,5126	10,4887	11,6571	13,9522	16,9161
19	1,0005	2,0897	2,0888	3,9945	3,9931	5,3843	5,3894	7,5206	7,6013	9,3371	9,8429	11,9255	13,3962	15,8981	19,1591
20	0,9993	2,3726	2,3722	4,3233	4,3250	5,9387	5,9728	8,0948	8,3014	10,0842	10,8635	12,9446	14,6601	17,2695	20,6892
21	1,0023	2,5667	2,5683	4,5468	4,5636	6,2876	6,4073	8,4282	8,8692	10,6112	11,6946	13,7334	15,6274	18,2933	21,7769
22	1,0042	2,6852	2,7063	4,6563	4,7536	6,4513	6,7860	8,6367	9,4056	11,0768	12,4000	14,4128	16,4161	19,0986	22,5894
23	1,0772	2,8011	2,9634	4,8127	5,1864	6,7838	7,5029	9,2380	10,3200	12,0115	13,5356	15,6099	17,7524	20,5256	24,1254
24	1,5857	3,3587	4,0473	5,9637	6,8591	8,6281	9,8160	11,7978	13,3009	15,3408	17,2901	19,7947	22,4306	25,7783	30,0840
25	2,0789	3,9147	5,0664	7,0579	8,3596	10,2901	11,8340	14,0092	15,8419	18,1472	20,4243	23,2381	26,2322	29,9730	34,7629
26	2,3946	4,2782	5,7293	7,7572	9,3297	11,3497	13,1252	15,3897	17,4299	19,8671	22,3301	25,2825	28,4484	32,3413	37,3028
27	2,5983	4,5193	6,1649	8,2107	9,9594	12,0321	13,9574	16,2598	18,4269	20,9273	23,4863	26,4823	29,7085	33,6247	38,5854
28	2,7301	4,6776	6,4472	8,5011	10,3587	12,4587	14,4744	16,7868	19,0243	21,5442	24,1421	27,1288	30,3512	34,2179	39,0866

<b>29</b>	2,8111	4,7777	6,6249	8,6817	10,6081	12,7205	14,7859	17,0907	19,3589	21,8683	24,4670	27,4152	30,5985	34,3818	39,1219
<b>30</b>	2,8638	4,8422	6,7392	8,7961	10,7659	12,8844	14,9800	17,2736	19,5552	22,0472	24,6335	27,5337	30,6651	34,3579	38,9603
<b>31</b>	2,8985	4,8846	6,8128	8,8683	10,8639	12,9827	15,0938	17,3718	19,6555	22,1242	24,6880	27,5378	30,6154	34,2186	38,6961
<b>32</b>	2,9204	4,9112	6,8596	8,9124	10,9225	13,0371	15,1530	17,4153	19,6923	22,1375	24,6782	27,4826	30,5030	34,0203	38,3675

Fuente: Realizado por Mónica F.

**Tabla 3-A.** Ubicaciones optimizadas, canales AWGN con desvanecimiento, 4096 NUC-QAM

4096 NUC-QAM																
SNR(dB)	AWGN Desvanecimiento															
	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16
<b>6</b>	1,0097	1,0467	1,0337	1,0885	1,1007	1,0611	1,0490	1,1057	1,1177	1,1579	1,1449	1,0892	1,1014	1,0619	1,0500	3,4408
<b>7</b>	1,0072	1,0288	1,0180	1,0841	1,0947	1,0713	1,0610	1,2062	1,2173	1,2437	1,2318	1,1572	1,1687	1,1440	1,1336	3,2474
<b>8</b>	1,0028	1,0323	1,0303	1,0665	1,0687	1,0375	1,0352	1,3196	1,3230	1,3637	1,3601	1,3143	1,3174	1,2769	1,2731	3,2673
<b>9</b>	0,9994	1,0103	1,0099	1,0416	1,0415	1,0295	1,0293	1,4848	1,4854	1,5033	1,5036	1,4560	1,4567	1,4404	1,4409	3,4167
<b>10</b>	0,9999	1,0036	1,0047	1,0207	1,0205	1,0180	1,0191	1,7124	1,7111	1,7169	1,7176	1,6837	1,6825	1,6764	1,6771	3,6611
<b>11</b>	0,9996	0,9995	0,9997	1,0048	1,0045	1,0044	1,0045	1,9789	1,9774	1,9770	1,9772	1,9674	1,9661	1,9651	1,9652	3,9531
<b>12</b>	1,0002	1,0000	1,0001	0,9958	0,9959	0,9963	0,9965	2,2284	2,2290	2,2279	2,2285	2,2398	2,2404	2,2423	2,2428	4,2293
<b>13</b>	1,0005	1,0007	1,0009	0,9914	0,9919	0,9921	0,9922	2,4296	2,4304	2,4297	2,4298	2,4752	2,4761	2,4776	2,4777	4,4538
<b>14</b>	1,0004	1,0009	1,0008	0,9953	0,9956	0,9956	0,9957	2,5769	2,5774	2,5794	2,5793	2,6756	2,6764	2,6747	2,6746	4,6252
<b>15</b>	1,0002	0,9991	0,9992	1,0125	1,0127	1,0133	1,0135	2,6774	2,6773	2,6833	2,6837	2,8656	2,8653	2,8576	2,8581	4,7674
<b>16</b>	1,0002	0,9984	0,9986	1,0670	1,0670	1,0683	1,0683	2,7777	2,7771	2,7840	2,7837	3,1097	3,1091	3,1006	3,1004	4,9751
<b>17</b>	0,9998	0,9984	0,9983	1,2356	1,2354	1,2372	1,2366	2,9864	2,9867	2,9906	2,9912	3,5858	3,5859	3,5770	3,5775	5,4740
<b>18</b>	0,9990	1,0001	0,9997	1,6294	1,6298	1,6304	1,6304	3,4541	3,4539	3,4555	3,4561	4,4555	4,4552	4,4507	4,4509	6,4759
<b>19</b>	0,9988	0,9996	0,9980	2,0374	2,0374	2,0374	2,0366	3,9300	3,9310	3,9297	3,9294	5,2770	5,2757	5,2792	5,2795	7,4011
<b>20</b>	0,9995	0,9992	0,9988	2,3337	2,3334	2,3318	2,3321	4,2773	4,2766	4,2772	4,2768	5,8584	5,8595	5,8832	5,8851	8,0147
<b>21</b>	0,9999	1,0004	1,0013	2,5341	2,5347	2,5328	2,5362	4,5077	4,5089	4,5202	4,5204	6,2318	6,2314	6,3219	6,3212	8,3718
<b>22</b>	1,0013	1,0032	1,0018	2,6647	2,6639	2,6769	2,6761	4,6411	4,6405	4,7063	4,7061	6,4228	6,4225	6,6852	6,6842	8,5777

<b>23</b>	0,9999	1,0364	1,0370	2,7604	2,7604	2,8536	2,8534	4,7342	4,7345	5,0028	5,0025	6,6184	6,6189	7,2150	7,2149	8,9581
<b>24</b>	1,0010	1,3959	1,3944	3,1499	3,1496	3,6527	3,6538	5,5345	5,5352	6,2647	6,2637	7,9618	7,9616	8,9989	8,9993	10,8873
<b>25</b>	0,9997	1,9271	1,9264	3,7388	3,7379	4,7448	4,7432	6,7158	6,7145	7,8812	7,8827	9,7611	9,7632	11,1777	11,1786	13,2883
<b>26</b>	1,0009	2,2720	2,2732	4,1332	4,1327	5,4651	5,4655	7,4777	7,4780	8,9382	8,9384	10,9166	10,9185	12,5855	12,5908	14,7944
<b>27</b>	0,9992	2,4940	2,4945	4,3943	4,3952	5,9409	5,9407	7,9756	7,9745	9,6301	9,6296	11,6754	11,6664	13,5279	13,4849	15,8428
<b>28</b>	1,0006	2,6454	2,6449	4,5762	4,5757	6,2598	6,2627	8,3042	8,3131	10,0729	10,1062	12,1078	12,2211	13,9787	14,2723	16,1851
<b>29</b>	1,0043	2,7379	2,7460	4,6828	4,7079	6,4365	6,5065	8,4445	8,6065	10,2540	10,5627	12,3025	12,8281	14,3539	15,1126	16,7817
<b>30</b>	1,4608	3,2219	3,6983	5,6197	6,1651	7,8821	8,5398	10,4315	11,2028	12,8833	13,7734	15,5990	16,6269	18,3392	19,5353	21,4333
<b>31</b>	2,0310	3,8448	4,8763	6,8402	7,9120	9,7373	10,8859	12,8891	14,1091	15,9704	17,2797	19,2948	20,7185	22,6727	24,2439	26,3916
<b>32</b>	2,3634	4,2255	5,5926	7,5722	8,9681	10,8558	12,3100	14,3481	15,8609	17,8101	19,3987	21,4876	23,1709	25,2367	27,0471	29,2986
<b>33</b>	2,5782	4,4775	6,0595	8,0510	9,6583	11,5914	13,2454	15,3017	17,0043	19,0084	20,7764	22,9043	24,7554	26,8878	28,8505	31,1470
<b>34</b>	2,7135	4,6426	6,3617	8,3598	10,0999	12,0627	13,8413	15,9032	17,7263	19,7625	21,6431	23,7857	25,7402	27,9051	29,9595	32,2702
<b>35</b>	2,8007	4,7502	6,5573	8,5611	10,3859	12,3691	14,2281	16,2913	18,1889	20,2440	22,1941	24,3407	26,3568	28,5338	30,6383	32,9453
<b>36</b>	2,8562	4,8197	6,6818	8,6878	10,5661	12,5612	14,4700	16,5325	18,4766	20,5396	22,5302	24,6737	26,7257	28,9056	31,0373	33,3306
<b>37</b>	2,8925	4,8656	6,7624	8,7697	10,6838	12,6874	14,6273	16,6862	18,6587	20,7276	22,7431	24,8824	26,9539	29,1316	31,2752	33,5558
<b>38</b>	2,9145	4,8927	6,8134	8,8219	10,7566	12,7630	14,7205	16,7743	18,7627	20,8283	22,8571	24,9883	27,0702	29,2408	31,3895	33,6536
<b>39</b>	2,9320	4,9155	6,8524	8,8609	10,8106	12,8207	14,7934	16,8469	18,8479	20,9151	22,9529	25,0776	27,1641	29,3276	31,4767	33,7270
<b>40</b>	2,9432	4,9282	6,8778	8,8865	10,8472	12,8615	14,8444	16,8975	18,9070	20,9752	23,0199	25,1403	27,2300	29,3905	31,5389	33,7773
<b>SNR(dB)</b>	d17	d18	d19	d20	d21	d22	d23	d24	d25	d26	d27	d28	d29	d30	d31	
<b>6</b>	3,3601	3,1046	3,1327	2,8894	2,8790	2,9999	3,0216	3,4745	3,3939	3,1184	3,1528	3,7542	3,6131	4,4880	5,6783	
<b>7</b>	3,2589	3,2433	3,2291	3,0788	3,0908	3,0955	3,0827	3,7641	3,7415	3,6371	3,6471	4,4918	4,3043	4,9563	6,2681	
<b>8</b>	3,2953	3,3198	3,2928	3,3078	3,3328	3,3114	3,2860	4,5388	4,5688	4,2838	4,2441	4,6545	4,6739	5,6429	6,9732	
<b>9</b>	3,4403	3,4991	3,4736	3,5536	3,5791	3,5221	3,4971	4,8012	4,8766	4,7955	4,7386	5,4141	5,4270	6,2434	7,6928	
<b>10</b>	3,6759	3,7370	3,7215	3,8874	3,9074	3,8387	3,8215	5,2768	5,3692	5,3854	5,3092	6,1357	6,2138	7,0601	8,5941	
<b>11</b>	3,9601	4,0075	3,9997	4,2677	4,2792	4,2143	4,2031	5,7983	5,8801	5,9967	5,9145	6,9502	7,0723	7,9787	9,5970	
<b>12</b>	4,2313	4,2628	4,2609	4,6320	4,6369	4,5858	4,5824	6,2845	6,3465	6,5550	6,4785	7,6934	7,8706	8,9010	10,5869	
<b>13</b>	4,4531	4,4661	4,4673	4,9550	4,9540	4,9287	4,9304	6,6904	6,7300	7,0280	6,9667	8,3177	8,5561	9,7418	11,4949	

14	4,6243	4,6188	4,6200	5,2460	5,2441	5,2556	5,2584	7,0279	7,0459	7,4305	7,3941	8,8516	9,1461	10,4828	12,3176	
15	4,7676	4,7505	4,7491	5,5472	5,5486	5,6032	5,6020	7,3462	7,3437	7,8203	7,8194	9,3478	9,7003	11,1807	13,1061	
16	4,9767	4,9585	4,9578	5,9566	5,9602	6,0637	6,0592	7,7844	7,7662	8,3639	8,4057	9,9831	10,4171	12,0295	14,0796	
17	5,4757	5,4675	5,4681	6,7028	6,7104	6,8816	6,8738	8,6613	8,6344	9,4101	9,5027	11,1654	11,7322	13,5243	15,7967	
18	6,4751	6,4921	6,4921	8,0145	8,0232	8,3055	8,2927	10,2586	10,2325	11,2613	11,4236	13,2585	14,0437	16,1109	18,7604	
19	7,3997	7,4646	7,4665	9,1828	9,1919	9,6410	9,6278	11,7039	11,6913	12,9619	13,2128	15,1610	16,2040	18,4804	21,4433	
20	8,0136	8,1858	8,1853	9,9708	9,9745	10,6757	10,6671	12,7329	12,7532	14,2091	14,5744	16,5544	17,8423	20,2275	23,3683	
21	8,3720	8,7429	8,7398	10,5001	10,4999	11,4966	11,5018	13,4830	13,5672	15,1187	15,6519	17,6098	19,1046	21,5413	24,7641	
22	8,5775	9,2532	9,2525	10,9359	10,9341	12,1712	12,2020	14,0789	14,2758	15,8369	16,6054	18,5000	20,1574	22,6144	25,8663	
23	8,9585	9,9400	9,9394	11,5978	11,6041	12,9965	13,0880	14,8780	15,2586	16,8096	17,8237	19,6965	21,4926	23,9997	27,2995	
24	10,8850	12,2345	12,2364	14,1236	14,1646	15,8137	16,0518	17,9826	18,6869	20,4248	21,8049	23,9472	26,1156	29,0224	32,8238	
25	13,2918	14,9831	15,0095	17,1287	17,2749	19,1281	19,6555	21,7256	22,8323	24,7955	26,5569	29,0147	31,5831	34,9290	39,2987	
26	14,8184	16,6805	16,8025	18,9131	19,3250	21,1680	22,0945	24,1425	25,5750	27,6541	29,6567	32,2679	35,0526	38,6023	43,2007	
27	15,6955	18,0725	17,6716	20,8359	20,0208	23,8620	22,6034	27,5194	25,8754	31,8142	29,6686	37,1329	34,4897	41,5575	45,9849	
28	16,7562	18,3851	19,2499	21,0165	22,1679	23,9006	25,3642	27,3645	29,1484	31,3343	33,5648	36,2682	39,1974	42,8228	47,4691	
29	17,7277	19,2664	20,3968	22,1002	23,4306	25,1704	26,7556	28,7536	30,6214	32,8290	35,1146	37,8272	40,7685	44,3725	48,9596	
30	22,7796	24,6112	26,1323	28,1574	29,8838	31,9904	33,9846	36,3885	38,6980	41,3722	44,1486	47,3995	50,9300	55,2190	60,6605	
31	28,1062	30,2335	32,1242	34,4644	36,5679	39,0270	41,4123	44,2002	46,9287	50,0315	53,2678	57,0085	61,0847	65,9903	72,1993	
32	31,2381	33,5042	35,6097	38,0874	40,3963	43,0119	45,5943	48,5385	51,4594	54,7340	58,1622	62,0839	66,3515	71,4580	77,8935	
33	33,2292	35,5666	37,8022	40,3385	42,7627	45,4482	48,1289	51,1272	54,1327	57,4607	60,9496	64,9050	69,2122	74,3362	80,7711	
34	34,4336	36,7993	39,1019	41,6513	44,1297	46,8305	49,5435	52,5353	55,5551	58,8707	62,3471	66,2558	70,5087	75,5397	81,8379	
35	35,1516	37,5228	39,8565	42,3946	44,8896	47,5760	50,2892	53,2485	56,2467	59,5120	62,9389	66,7683	70,9320	75,8248	81,9286	
36	35,5544	37,9155	40,2573	42,7722	45,2621	47,9200	50,6125	53,5222	56,4806	59,6830	63,0415	66,7757	70,8300	75,5766	81,4804	
37	35,7864	38,1359	40,4733	42,9647	45,4415	48,0690	50,7335	53,5931	56,5119	59,6552	62,9454	66,5764	70,5214	75,0956	80,7898	
38	35,8824	38,2116	40,5406	43,0011	45,4589	48,0512	50,6844	53,4946	56,3625	59,4393	62,6599	66,2081	70,0572	74,5263	80,0683	
39	35,9518	38,2677	40,5882	43,0265	45,4682	48,0322	50,6390	53,4105	56,2412	59,2680	62,4390	65,9151	69,6802	74,0405	79,4178	
40	35,9970	38,2999	40,6091	43,0262	45,4529	47,9900	50,5714	53,3077	56,1041	59,0883	62,2059	65,6227	69,3125	73,5803	78,8141	

Fuente: Realizado por Mónica F.

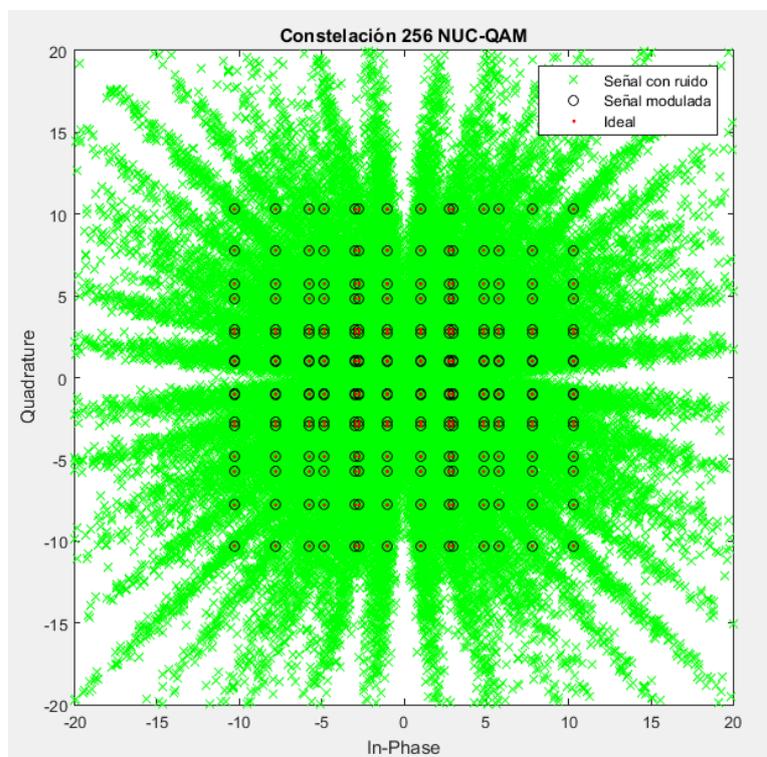
## ANEXO C

### CONSTELACIONES PARA MODULACIONES NO UNIFORMES

Aquí se encuentran algunas constelaciones optimizadas para diferentes órdenes de modulaciones y con diferentes SNR.

En la figura 23-A se puede observar una constelación 256 NUC-QAM con un SNR=15dB, se observa claramente que las distancias entre bits son diferentes.

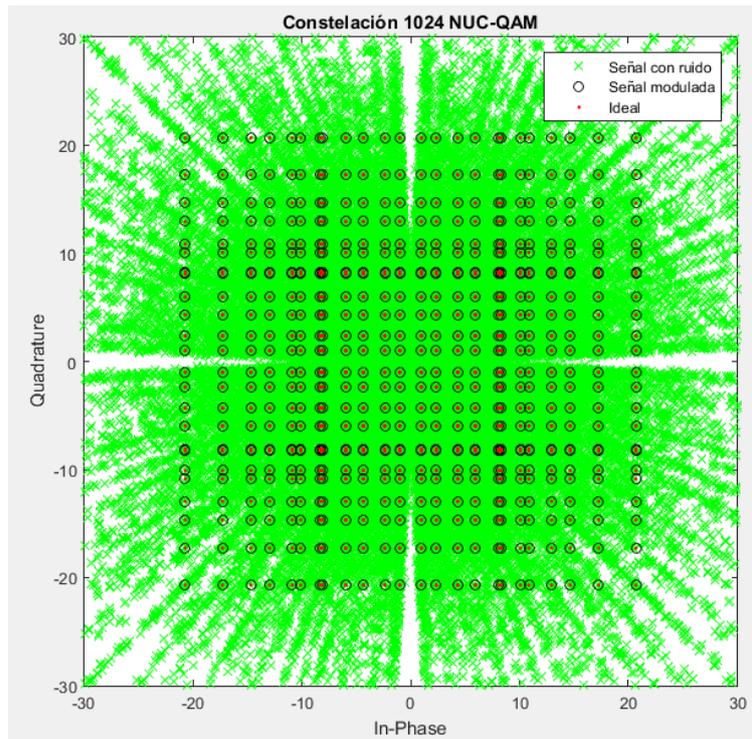
Además, se observa que en algunos puntos de la constelación se encuentran solapados, esto se debe a que en la optimización el mejor valor de  $d_k$  está cerca a su vecino más próximo.



**Figura 23-A.** Constelación 256 NUC-QAM, SNR=15 dB

Fuente: Realizado por Mónica F.

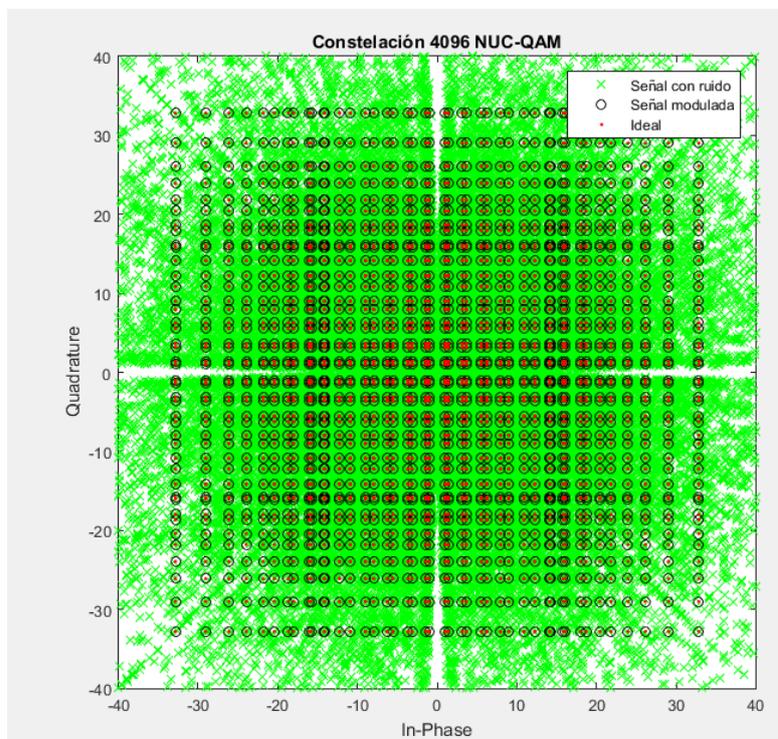
En la figura 24-A se puede observar una constelación 1024 NUC-QAM con un SNR=20 dB, se observa claramente que las distancias entre bits son diferentes.



**Figura 24-A.** Constelación 1024 NUC-QAM, SNR=20 dB

Fuente: Realizado por Mónica F.

En la figura 25-A se puede observar una constelación 4096 NUC-QAM con un SNR=24 dB, se observa claramente que las distancias entre bits son diferentes.



**Figura 25-A.** Constelación 4096 NUC-QAM, SNR=24 dB

Fuente: Realizado por Mónica F.

## ANEXO D

### MÉTRICAS DE DESEMPEÑO

En esta sección se encuentran tabuladas las métricas de desempeño de las modulaciones 256, 1024 y 4096 NUC-QAM, obtenidas mediante simulación con las distintas codificaciones desarrolladas en este proyecto sobre un canal AWGN con Desvanecimiento.

**Tabla 4-A.** Métricas de Desempeño 256 NUC-QAM Codificación LDPC

BER		
	UC	NUC
SNR dB	LDPC	LDPC
16	0,1	0,07010625
17	0,1	0,000424925
18	1,4608E-05	2,128E-09

Fuente: Realizado por Mónica F.

**Tabla 5-A.** Métricas de Desempeño 1024 NUC-QAM Codificación LDPC

BER		
	UC	NUC
SNR dB	LDPC	LDPC
14	0,027257	0,00343034
15	0,0045222	1,59622E-06
16	1,8452E-06	1,39993E-08

Fuente: Realizado por Mónica F.

**Tabla 6-A.** Métricas de Desempeño 4096 NUC-QAM Codificación LDPC

BER		
	UC	NUC
SNR dB	LDPC	LDPC
12	0,03619083	0,02513192
13	0,03081167	0,00023935
14	0,00255642	2,2612E-05
15	0,00020613	2,1454E-06

Fuente: Realizado por Mónica F.

**Tabla 7-A.** Métricas de Desempeño 256 NUC-QAM Codificación Turbo

BER				
	UC		NUC	
SNR dB	Log-Map	Sova	Log-Map	Sova
14	0,141	0,166375	0,09196625	0,10793625
15	0,02071875	0,0152	0,004425	0,00587838
16	0,001345	0,00141125	8,9825E-05	0,00015819
17	0,00082463	0,00013463	2,1838E-05	2,0063E-05
18	0,00062775	0,00013475	1,3538E-05	0,0000139

Fuente: Realizado por Mónica F.

**Tabla 8-A.** Métricas de Desempeño 1024 NUC-QAM Codificación Turbo

BER				
	UC		NUC	
SNR dB	Log-Map	Sova	Log-Map	Sova
19	0,122126	0,1012	0,028572	0,021934
20	0,0501601	0,00997	0,0039964	0,00011745
21	0,00841272	0,000997	0,00023331	0,00002872
22	0,00498937	0,001003	0,00014051	0,00001134

Fuente: Realizado por Mónica F.

**Tabla 9-A.** Métricas de Desempeño 4096 NUC-QAM Codificación Turbo

BER				
	UC		NUC	
SNR dB	Log-Map	Sova	Log-Map	Sova
20	0,49986417	0,4283335	0,193975	0,18112833
21	0,0855185	0,0856667	0,01994608	0,01643225
22	0,03323544	0,0128625	0,0023418	0,00141716
23	0,01886417	0,00424167	0,00098375	0,00021645
24	0,01068686	0,00084917	0,00055154	0,0000626

Fuente: Realizado por Mónica F.



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL APRENDIZAJE  
UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y DOCUMENTAL**

**REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA**

**Fecha de entrega:** 24 / 11 / 2021

<b>INFORMACIÓN DEL AUTOR/A (S)</b>
<b>Nombres – Apellidos:</b> <i>Mónica Jeanette Flores Villafuerte</i>
<b>INFORMACIÓN INSTITUCIONAL</b>
<i>Instituto de Posgrado y Educación Continua</i>
<b>Título a optar:</b> <i>Magíster en Sistemas de Telecomunicaciones</i>
<b>f. Analista de Biblioteca responsable:</b> <i>Lic. Luis Caminos Vargas Mgs.</i>

**LUIS  
ALBERTO  
CAMINOS  
VARGAS**

Firmado digitalmente por  
LUIS ALBERTO CAMINOS  
VARGAS  
Nombre de reconocimiento  
(DN): c=EC, I=RIOBAMBA,  
serialNumber=0602766974,  
cn=LUIS ALBERTO  
CAMINOS VARGAS  
Fecha: 2021.11.24 08:45:10  
-05'00'



0110-DBRAI-UTP-IPEC-2021