



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA DE  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



**DISEÑO Y SIMULACIÓN DE UN INTERLEAVER PARA LOS TURBO  
CÓDIGOS EN LAS REDES VDSL**

JENNIFER SANDIA

Bárbula, 9 de Agosto del 2015



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA DE  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



## **DISEÑO Y SIMULACIÓN DE UN INTERLEAVER PARA LOS TURBO CÓDIGOS EN LAS REDES VDSL**

TRABAJO ESPECIAL DE GRADO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD DE  
CARABOBO PARA OPTAR AL TÍTULO DE INGENIERO DE TELECOMUNICACIONES

JENNIFER SANDIA

Bárbula, 9 de Agosto del 2015



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA DE  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



## CERTIFICADO DE APROBACIÓN

Los abajo firmantes miembros del jurado asignado para evaluar el trabajo especial de grado titulado «DISEÑO Y SIMULACIÓN DE UN INTERLEAVER PARA LOS TURBO CÓDIGOS EN LAS REDES VDSL», realizado por el bachiller JENNIFER SANDIA, cédula de identidad 20.365.378, hacemos constar que hemos revisado y aprobado dicho trabajo.

**Firma**

Prof. ELIMAR HERNÁNDEZ  
TUTOR

**Firma**

Prof. ANTONIO FEDON  
JURADO

**Firma**

Prof. AHMAD OSMAN  
JURADO

Bárbula, 9 de Agosto del 2015

# Dedicatoria

*A Dios porque es mi todo*

*A mis padres*

*A mi abuela Flor, para que desde el cielo contemple que ya por fin me gradué*

*A mi, porque a pesar de los imposibles, lo logré.*

**JENNIFER SANDIA**

# Agradecimientos

Principalmente agradecida con Dios que me bendijo con una buena salud para estar hoy donde estoy, porque como dice mi abuela Isaura Contreras de Sandia sin salud no hay nada.

Agradecida eternamente con todos y cada uno de los miembros de mi familia, porque me brindaron una ayuda siempre que la necesite, especialmente con mi mamá Carolina Carcamo que es mi ejemplo de mujer a seguir, con mis padrinos Gertrudis Sandia y Edwin Sánchez que me han brindado ayuda económica incondicionalmente y me regalaron mis implementos de estudios más importantes de la carrera mi pc y mi calculadora y todo sea por verme con un título en la mano, con mi tía Angelina Sandia porque siempre ha sido como una segunda mamá para mí. A mi tío Eduardo Carrero por su idea de que esta era la carrera para mí y por su ayuda. A mi papá José Sandia, mi hermano Richard Sandia y a mis tías, tíos, primas y primos gracias!

A mis compañeros Manuel De Oliveira, Daniel Martínez y Jonás Sánchez porque más que estar conmigo en muchas horas de clases, muchas noches de estudios se convirtieron en mis amigos, juntos los cuatro somos JaJuJo, los quiero muchachos! A mis también compañeros y próximamente colegas Yutzani Gallardo, Rosana Márquez, Yanina Perdomo y Ronald Ustariz porque solo ustedes pueden entender mis penas y alegrías académicas, porque con uds compartí esta experiencia de ser tesista.

Profundamente agradecida con mi casa de estudios porque me dio la oportunidad de formarme profesionalmente, porque solo un estudiantes universitario sabe el amor que se siente por su universidad, orgullosa de ser UCista!! Gracias a todos mis profesores, tratarlos respetuosamente desde que los conocí fue mi manera de agradecerles su enseñanza. Especialmente gracias a mi tutora Elimar Hernández

por guiarme en la última de mis tareas como bachiller. También agradezco al Cine Club de Ingeniería, uds me enseñaron tips de la vida de ingeniero.

Ahora bien, sé que olvidare a más de uno y me disculpo de ante mano, pero quiero darle un enorme gracia a todos los que conmigo estudiaron y compartieron en la UC. A Maikol Rodríguez, Yuriandris Lugo y a Susana Mujica porque fueron los primeros que conocí en la UC y en el camino se volvieron mis amigos de verdad, también a Luis Guedez otro amigo del básico y que en la escuela me animo a seguir en Telecom cuando ni yo misma estaba segura de sí ésta era la carrera de mi vida. A Daniela Salermi por ser mi amiga y hermana, gracias por esas noches en las que me amparaste. Arley Noguera, Carlos Palma, Daniela Cordero, Patricia Eubieda, David Cardona, Gustavo Malpica, Simón por los favores y sus ayuda en algún momento, gracias!

Finalmente quiero agradecerles a dos personas muy especiales que me acompañaron durante mis estudios, a todos nos ha pasado que llega un momento en tu vida en el que no tienes ánimos para seguir en la carrera cuesta arriba de tus estudios, en el que necesitas un respiro de la universidad y en el que necesitas también que te hagan presión con la universidad, y eso fue lo que uds hicieron. El primero de ellos, Carlos Díaz, contigo inicie este sueño de ser ingeniera, juntos nos inscribimos y juntos vivimos la experiencia de un primer día en la universidad, gracias por estar conmigo, esta meta cumplida es para los dos. El segundo de ellos, Yoel Hernández, gracias acompañarme en esta etapa de ser tesista, la etapa más dura de mi carrera, gracias por animarme y por estar pendiente de todos los avances de este trabajo de grado. Gracias!

# Índice general

<b>Índice de Figuras</b>	<b>XI</b>
<b>Índice de Tablas</b>	<b>XIII</b>
<b>Acrónimos</b>	<b>XV</b>
<b>Resumen</b>	<b>XVII</b>
<b>I. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	3
1.2.1. Objetivo General . . . . .	3
1.2.2. Objetivos Específicos . . . . .	3
1.3. Alcance . . . . .	4
<b>II. Marco conceptual</b>	<b>5</b>
2.1. Codificación de canal . . . . .	5
2.1.1. Códigos correctores de errores . . . . .	6
2.2. Turbo Códigos . . . . .	7
2.2.1. Aplicaciones de los Turbo Códigos . . . . .	8
2.2.2. Turbo Codificador . . . . .	8
2.2.3. Modulador DMT . . . . .	10
2.2.4. Mapeo de constelación . . . . .	12
2.2.5. Turbo Decodificador . . . . .	13
2.3. Interleaver, Permutador, Intercalador . . . . .	16
2.3.1. Interleaver de Bloque o Matricial . . . . .	16
2.3.2. Interleaver Espiral o Helicoidal . . . . .	17
2.3.3. Interleaver pseudo-aleatorio . . . . .	17
2.3.4. Interleaver Convolutacional . . . . .	19
2.4. El canal . . . . .	20
2.4.1. Canal AWGN . . . . .	20
2.4.2. Tasa de Error de Bit y $E_b/N_0$ . . . . .	21
2.5. VDSL . . . . .	22

2.5.1. Tipos de Servicios VDSL . . . . .	23
2.5.2. Bandas de Frecuencias implementadas en VDSL . . . . .	24
2.6. Teoría Combinatoria . . . . .	25
2.6.1. Permutaciones . . . . .	25
<b>III. Procedimientos de la investigación</b>	<b>27</b>
3.1. Etapa I. Revisión Bibliográfica . . . . .	27
3.2. Etapa II. Diseño del permutador . . . . .	28
3.2.1. Selección del software . . . . .	28
3.2.2. Simulación del canal de comunicación de la red VDSL . . . . .	29
3.2.2.1. Características del transmisor . . . . .	29
3.2.2.2. Características del canal . . . . .	35
3.2.2.3. Características del receptor . . . . .	35
3.2.3. Diseño y simulación del permutador para los Turbo Códigos . . . . .	36
3.3. Etapa III. Verificación del funcionamiento del interleaver . . . . .	37
3.4. Etapa IV. Análisis y conclusiones . . . . .	39
3.4.1. Permutador aleatorio . . . . .	39
3.4.2. Permutador matricial . . . . .	39
3.4.3. Permutador helicoidal . . . . .	39
3.4.4. Permutador convolucional . . . . .	40
<b>IV. Análisis, interpretación y presentación de los resultados</b>	<b>43</b>
4.1. Análisis del algoritmo del interleaver diseñado . . . . .	43
4.2. Variación del BER según el SNR . . . . .	46
4.3. Comparación de BER entre el canal típico VDSL y canal VDSL con interleaver el diseñado . . . . .	48
4.4. Comparación de BER usando diferentes permutadores en una canal VDSL . . . . .	49
<b>V. Conclusiones y recomendaciones</b>	<b>51</b>
5.1. Conclusiones . . . . .	51
5.2. Recomendaciones . . . . .	53
<b>A. Inicialización del parámetro b</b>	<b>55</b>
<b>B. Algoritmos</b>	<b>57</b>
2.1. Algoritmo de permutación del VDSL Interleaver . . . . .	57
2.2. Algoritmo para graficar el canal típico VDSL usando un interleaver genérico y el diseñado . . . . .	60
2.3. Algoritmo para graficar el rendimiento en términos de BER implementando diferentes permutadores . . . . .	62

Índice general IX

---

**Referencias Bibliográficas 63**

# Índice de figuras

2.1. Esquemas de tipos de codificadores de control de errores [1]. . . . .	7
2.2. Estructura del codificador turbo [2]. . . . .	9
2.3. Ejemplo de un codificador convolucional [2]. . . . .	10
2.4. Constelaciones para $b = 2$ y $b = 4$ [3]. . . . .	13
2.5. Esquema del Turbo decodificador [2]. . . . .	14
2.6. Esquema de los tipos de algoritmos de decodificación.[2]. . . . .	15
2.7. Ejemplo de implementación de interleaver convolucional [3]. . . . .	20
2.8. Topología de VDSL [4]. . . . .	23
2.9. Bandas de frecuencias del plan 998. [3] . . . . .	24
3.1. Esquema básico de un Turbo Codificador. . . . .	28
3.2. Esquema básico simulado de la red VDSL. Fuente propia. . . . .	31
3.3. Estructura del aleatorizador [3]. . . . .	32
3.4. Estructura del codificador turbo. Fuente propia. . . . .	32
3.5. Tabla de parámetros del interleaver convolucional usado en VDSL [5].	41
4.1. Curva del BER para el canal VDSL con interleaver genérico y el di- señado. Fuente propia. . . . .	48
4.2. Comparación de rendimiento usando diferentes permutadores. Fuen- te Propia. . . . .	49

# Indice de tablas

3.1. Tabla de diferentes longitudes de restricción usadas en VDSL.[6] . . .	33
4.1. Matriz 1. Matriz rectangular C*R . . . . .	44
4.2. Matriz 2. Matriz rectangular formada a partir del vector $s$ . . . . .	45
4.3. Matriz 3. Matriz rectangular permutada . . . . .	45
1.1. Inicialización del vector fila $b$ . . . . .	56

# Acrónimos

<b>ATM</b>	Asynchronous Transfer Mode.
<b>AWGN</b>	Additive White Gaussian Noise.
<b>BER</b>	Bit Error Rate.
<b>CAP</b>	Carrierless Amplitude Phase Modulation.
<b>CRC</b>	Cyclic Redundancy Check.
<b>DMT</b>	Discrete Multitone Technique.
<b>Eb/No</b>	Energy per bit to Noise power spectral density ratio.
<b>FDM</b>	Frequency Division Multiplexing.
<b>FEC</b>	Forward Error Correction.
<b>FIFO</b>	First In First Out.
<b>FTTCab</b>	Fiber To The Cabinet.
<b>FTTH</b>	Fiber To The Home.
<b>GUI</b>	Graphical User Interface.
<b>IFFT</b>	Inverse Fast Fourier Transform.
<b>LLR</b>	Log- Likelihood Ratio.
<b>MAP</b>	Maximum A-Posteriori.
<b>OFDM</b>	Orthogonal Frequency Division Multiplexing.
<b>PCCC</b>	Paralel Concatenated Convolutional Codes.
<b>QAM</b>	Quadrature Amplitude Modulation.
<b>RSC</b>	Recursive Systematic Convolutional.
<b>S/N</b>	Signal/Noise.
<b>SISO</b>	Soft In Soft Out.
<b>SOVA</b>	Soft Output Viterbi Algorithm.
<b>VDSL</b>	Very high bit-rate Digital Subscriber Line.

# **DISEÑO Y SIMULACIÓN DE UN INTERLEAVER PARA LOS TURBO CÓDIGOS EN LAS REDES VDSL**

por

JENNIFER SANDIA

Presentado en el Departamento de Señales y Sistemas  
de la Escuela de Ingeniería en Telecomunicaciones  
el 9 de Agosto del 2015 para optar al Título de  
Ingeniero de Telecomunicaciones

## **RESUMEN**

El presente trabajo describe el diseño y simulación de un permutador para los Turbo Códigos que permita mejorar su rendimiento en términos de BER en los sistemas de comunicaciones VDSL. Para ello se inicia con la evaluación del principio de funcionamiento de permutados ya existente y que tienden a ser usados en los sistemas de transmisión actuales, para a partir de ellos lograr el diseño de un nuevo permutador que ofrezca mejoras en la corrección de errores que se presentan durante la transmisión. Adicionalmente, este trabajo presenta una descripción y análisis de los elementos que conforman la estructura del codificador y decodificador de un Turbo código y los demás elementos que forman parte de un sistema VDSL.

Las simulaciones del sistema modelado se obtuvieron a través de una plataforma realizada en un software, que se adaptó a los parámetros del canal de una red VDSL. Los resultados ilustraron en forma gráfica el rendimiento del sistema en términos de BER contra  $E_b/N_0$ , los cuales se compararon con los resultados obtenidos por otros permutadores ya existentes. Finalmente se comprueba que el rendimiento obtenido con el interleaver diseñado es más óptimo que los resultados obtenidos por otros permutadores ya existentes.

Palabras Claves: Interleaver, Turbo Códigos, VDSL

Tutor: ELIMAR HERNÁNDEZ

Profesor del Departamento de Señales y Sistemas

Escuela de Telecomunicaciones. Facultad de Ingeniería

# Capítulo I

## Introducción

### 1.1. Motivación

Los sistemas de comunicación digital, día tras día, han ido revolucionando la tecnología de tal forma que se han convertido en pilares fundamentales para la mayoría de las sociedades del mundo [7]. Estos sistemas permiten la comunicación entre personas o empresas a kilómetros de distancia; ya que su propósito principal es el de transportar eficientemente una señal que contenga información desde un origen hacia un destino a través de un canal de comunicaciones.

Por otro lado, durante la transmisión de la información pueden producirse alteraciones de la misma debido a la presencia de ruido en el canal, a estas alteraciones se les denomina errores [8]. El estándar VDSL, utiliza técnicas de corrección de errores hacia adelante (FEC), que se encargan de detectar y si es posible corregir los errores que se presenten. Esta técnica se basa en la modificación de la información transmitida mediante el agregado de redundancia, para luego ser procesada por el receptor para extraer los datos originales con la mayor fiabilidad posible.

Una de las técnicas de corrección FEC que han generado resultados muy cercanos al límite teórico de Shannon, son los Turbo Códigos [9], basado en esto se elige

un sistema VDSL que incluya a los Turbo Códigos como técnica de codificación de canal, los cuales codifican la información mediante una combinación de dos codificadores convolucionales junto con un permutador o interleaver. De acuerdo a [10], el interleaver es un dispositivo que permuta los bits de información de una manera predefinida, de forma tal que dos bits que están muy cercanos en la secuencia de información original, queden los más separados que se pueda en la secuencia de información permutada. El interleaver, es un elemento importante en el diseño de los Turbo Códigos ya que ha demostrado que permite mejorar su rendimiento en términos de BER [11].

Por otra parte, estudios realizados en trabajos de grado reciente de la Universidad Politécnica de Cartagena, en la Escuela Técnica Superior de Ingeniería de Telecomunicaciones, sobre la tecnología WiMax demuestran que la técnica de codificación de canal que mejores resultados aporta son los Turbo Códigos [2]. Se ha comprobado también que los decodificadores turbo son de alta complejidad y que requieren de varias iteraciones para recuperar el mensaje original, siendo pieza clave del proceso el interleaver, que normalmente representa el cuello de botella de los decodificadores turbo, es por ellos, el interés en el estudio y diseño de nuevas estructuras para los permutadores de bits de los Turbo Códigos.

Otra referencia para esta investigación esta en uno de los trabajos de grado realizados en la Escuela Nacional de Ecuador en la Facultad de Ingeniería Eléctrica y Electrónica donde se realizó el diseño de un interleaver para Turbo Códigos y se concluye, que a mayor tamaño y diseño, y mayor número de iteraciones utilizadas, se incrementa el desempeño ofrecido por un Turbo Código pero aumenta también la memoria y el tiempo requerido por el turbo decodificador para producir su secuencia de salida [12].

Por lo antes expuesto, en las últimas décadas con muchos los investigadores que se han dedicado al estudio de los permutadores para mejorar el rendimiento de los Turbo Códigos. Entre estos investigadores se destacan los trabajos de Divsalar, el cual propuso un tipo de intercalador S-Random [11], con el que logró aumentar

significativamente el rendimiento de los turbo códigos cuando el tamaño del permutador crece, pero trajo como desventaja un incremento del ancho de banda y por ende una disminución de su velocidad causada por la alta carga computacional.

Dentro de esta perspectiva, el diseño del permutador juega un papel importante en la rendimiento del Turbo Código, con el objeto de obtener una buena relación de señal a ruido SNR y una tasa de error bit en valores lo más óptimos posibles [13]. Por estas razones, surge la motivación de realizar la presente investigación, basada en el desarrollo de un interleaver para los turbo códigos que permita mejorar el rendimiento en términos de BER para las redes VDSL.

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Desarrollar un permutador para los Turbo Códigos que permita mejorar el rendimiento en términos de BER de las redes VDSL.

### **1.2.2. Objetivos Específicos**

- Revisar las fuentes bibliográficas de los Turbo Códigos, sus elementos y las redes VDSL para seleccionar la estructura de turbo código a utilizar.
- Simular el canal típico de las redes VDSL para la evaluación del desempeño de los Turbo Códigos.
- Diseñar el permutador de los Turbo Códigos para que trabaje en redes VDSL.
- Simular en el canal desarrollado VDSL el permutador de los Turbo Código para evaluar su rendimiento en términos de BER.
- Comparar los resultados generados por el permutador diseñado, con los obtenidos por otros permutadores existentes.

### **1.3. Alcance**

El diseño del permutador para los turbo códigos se limita a lograr mejores prestaciones en los sistemas de redes VDSL en términos de la tasa de error de bit BER, intentando mantener un nivel de complejidad y de tiempo de respuesta aceptable. Para el diseño y simulación del permutador se creó en primer lugar un modelo del sistema de comunicación típico de las redes VDSL permitiendo de esta manera evaluar el desempeño de los turbo código como técnica de corrección de errores, mediante la construcción de las gráficas de BER versus la  $E_b/N_0$ . Finalmente, los resultados obtenidos con el permutador diseñado serán comparados con los obtenidos por otros permutadores existentes.

## Capítulo II

# Marco conceptual

### 2.1. Codificación de canal

Todo sistema de comunicación tiene un mismo objetivo: la transmisión de información desde un emisor hasta un receptor a través de un canal. Debido a la naturaleza ruidosa de todo canal el mensaje es distorsionado hasta llegar al receptor, por lo que en muchos casos se producen diferencias entre las secuencias de datos enviadas y las recibidas, estas diferencias son llamadas errores. Para disminuir los errores en recepción se realiza la codificación de canal, cuyo objetivo es que el receptor sea capaz de detectar y corregir los errores producidos durante la transmisión.

La codificación de canal consiste en introducir redundancia, al convertir un bloque de mensaje de longitud  $k$  en una palabra código de longitud  $n$  ( $n > k$ ) de forma que sea posible reconstruir la secuencia de datos original de la manera más fiable posible en el receptor utilizando la redundancia ( $n - k$ ). Existen dos técnicas fundamentales en la corrección de errores:

- ARQ (*Automatic Repeat reQuest*).

Solicitud de repetición automática (Detección de errores). Cuando el receptor

detecta un error solicita al emisor la repetición del bloque de datos transmitido. El emisor retransmitirá los datos tantas veces como sea necesario hasta que los datos se reciban sin errores [11].

- FEC (*Forward Error Correction*).  
Corrección de errores hacia delante. Se basa en el uso de códigos autocorrectores que permiten la corrección de errores en el receptor. [14]

Es fácilmente observable que las técnicas de corrección FEC son más eficaces que las ARQ, ya que no se debe retransmitir el mensaje original, por lo que son indicadas para sistemas sin retorno o aplicaciones en tiempo real donde no se puede esperar a una retransmisión.

### 2.1.1. Códigos correctores de errores

- Códigos bloque: (codificador sin memoria) Es una técnica empleada para transformar una secuencia de datos binarios (secuencia de información), estructurándolos en conjuntos de bloques o vectores binarios denotados por  $m$  (mensaje). Cada bloque binario tiene una longitud fija de  $k$  elementos de información, a los cuales se añade una cierta longitud fija de bits redundantes, con la finalidad de detectar errores y, eventualmente corregirlos. [15]
- Códigos convolución: (codificador con memoria) Esta técnica va codificando los bits tal y como van llegando al codificador. Cabe destacar que la codificación de uno de los bits está enormemente influenciada por la de sus predecesores. La decodificación para este tipo de código es compleja ya que en principio, es necesaria una gran cantidad de memoria para estimar la secuencia de datos más probable para los bits recibidos. [16]

Entre los códigos de convolución y con el propósito de poder construir un turbo código con ellos, los más empleados son los códigos de convolución recursivos

sistemáticos (RSC). En un código sistemático, la secuencia de información de entrada se encuentra formando parte de la secuencia codificada de salida; los códigos de convolución sistemáticos presentan un mejor desempeño comparados con los no sistemáticos, siempre y cuando se realice una codificación recursiva. [14]. En la Figura 2.1 se muestra el diagrama de los esquemas de codificación de canal.

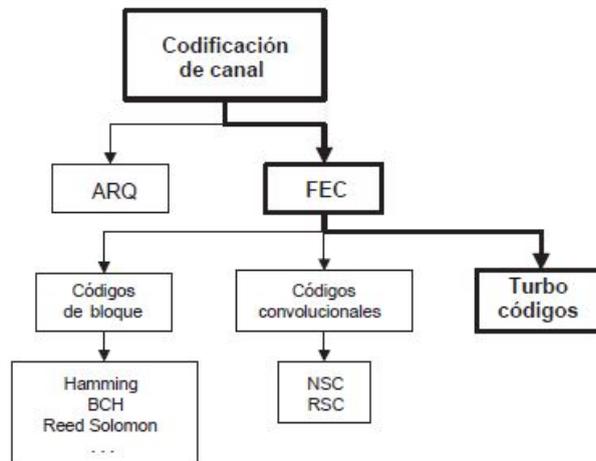


Figura 2.1: Esquemas de tipos de codificadores de control de errores [1].

## 2.2. Turbo Códigos

En el transcurso del año 1993, cuando en una conferencia de la IEEE sobre Comunicaciones en Ginebra (Suiza), dos ingenieros electrónicos franceses Claude Berrou y Alain Glavieux [17], hicieron una fabulosa aclamación: habían inventado un esquema de codificación digital que podría llevar a una eventual transmisión sin errores y potencias de transmisión eficaces, más allá de lo que la mayoría de los expertos podían llegar a pensar. Defendían que con ese esquema de codificación y haciendo uso de la misma potencia de transmisión se podía llegar a conseguir el doble de velocidad que con otros esquemas de codificación, o viceversa, con la mitad de potencia en transmisión se conseguían las mismas tasas de transmisión.

Este nuevo esquema de codificación era llamado Turbo Código y pertenece a la familia de códigos correctores de errores convolucionales. Ésta codificación posee un rendimiento en términos de tasa de error de bit (BER) muy próximos al conocido límite de Shannon [9], límite matemático teórico de la máxima tasa de transferencia en entornos ruidosos.

### 2.2.1. Aplicaciones de los Turbo Códigos

Desde la aparición de los códigos correctores de errores hacia delante y los resultados que se obtuvieron en simulaciones, una de las prioridades ha sido implementar dichos códigos en distintas aplicaciones con el objetivo de poder asegurar el manejo de la información a altas velocidades y en diferentes medios.

En la década de los años 90's con el descubrimiento de los turbo códigos, se inició una nueva etapa de desarrollo en la implementación de estas técnicas para resolver y mejorar soluciones en los medios de comunicación. En el caso de los turbo códigos se necesitaron simplificar y reducir los algoritmos inicialmente propuestos y a la vez diseñar arquitecturas orientadas a éstos.

En la actualidad algunas aplicaciones en donde se utilizan los turbo códigos son: comunicaciones satelitales y aeroespaciales, comunicaciones móviles e inalámbricas, almacenamiento multimedia, módems VDSL, comunicaciones sobre fibra óptica, televisión digital, telemetría y comunicaciones en el espacio libre.

### 2.2.2. Turbo Codificador

El codificador turbo implementa dos codificadores convolucionales, sistemáticos y recursivos, el primero codifica la secuencia de entrada (de longitud  $m$  bits) mientras que el segundo codifica la secuencia de entrada permutada por el interleaver [18].

Finalmente la palabra código a enviar al canal, es conformada con los bits originales sin codificar ( $m$  bits), y las dos secuencias de paridad ( $\frac{n}{2}$  bits cada una) producidas por cada codificador convolucional, como se muestra en la Figura 2.2 [19], por lo que la palabra código presenta una longitud de  $m+n$  bits. Una vez que son transmitidos los bits por el canal, éstos son expuestos al ruido pudiendo llegar erróneos o no.

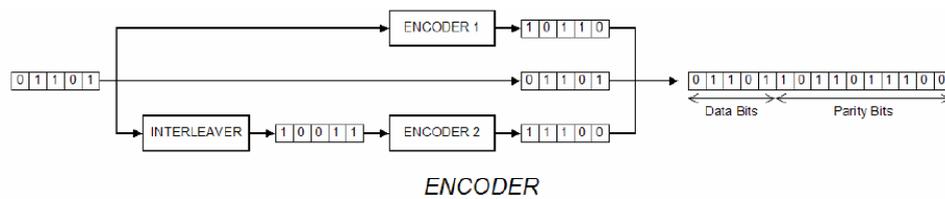


Figura 2.2: Estructura del codificador turbo [2].

La combinación de los codificadores se conoce como concatenación y los posibles arreglos de arquitectura son:

- Turbo Códigos Convolucionales Concatenados Paralelamente o PCCC (*Parallel Concatenated Convolutional Codes*)
- Turbo Códigos Convolucionales Concatenados Serialmente o SCCC (*Serial Concatenated Convolutional Codes*).

Se muestra en la Figura 2.3 un ejemplo de la estructura de un turbo codificador, este esquema tiene una entrada  $u$  y una salida sistemática que corresponde con el valor de la entrada  $v^{(1)}$  y dos salidas de bits de pariedad  $v^{(2)}$  y  $v^{(3)}$ . Así de esta manera, al tener una sola entrada sólo se tendrá una restricción de longitud de valor igual al número de registros más uno, o sea 3. En lo que al polinomio generador se refiere, se tiene  $g_0 = 1101$  y  $g_1 = 1111$ . Estos valores binarios corresponden en octal a 15 y 17 respectivamente, por lo que el valor de la variable generador de código es [15 17]. De aquí también obtenemos el valor de la variable conexión de Feedback, y para este ejemplo es 15.

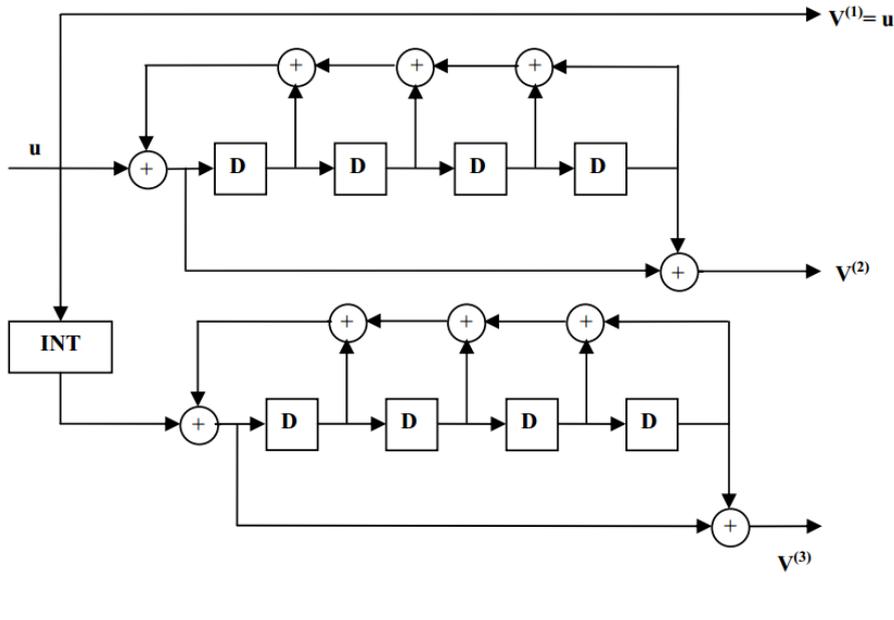


Figura 2.3: Ejemplo de un codificador convolucional [2].

### 2.2.3. Modulador DMT

La modulación DMT (Modulación por Multitono Discreto) utiliza el mismo principio que la Multiplexación por División de Frecuencia Ortogonal (OFDM). Es decir, se divide el espectro útil de frecuencias en canales paralelos, donde el centro de cada canal se representa por una subportadora modulada en QAM (Modulación de Amplitud en Cuadratura).

Así mismo en DMT, una modulación multitono o multiportadora, divide el ancho de banda disponible en un amplio número de subcanales idealmente independientes y aislados espectralmente [20]. El modulador DMT realiza esta modulación mediante implementaciones ortogonales sobre bloques de datos, este proceso es llamado subcanalización. Cada subcanal confina la potencia dentro de una banda estrecha, superponiéndose a los canales inmediatamente adyacentes. En una aplicación real, el proceso de subcanalización no consigue un aislamiento espectral perfecto entre sub-bandas.

Por ello es necesaria la modulación DMT, ya que es capaz de adaptarse a las condiciones del canal, asignando un mayor número de bits o bytes a aquellos subcanales con mayor relación S/N (Señal/Ruido), y menos número de bits o bytes a aquellos con menor relación S/N. Es posible eliminar las interferencias de banda estrecha anulando las sub-bandas afectadas, optimizando así el uso del ancho de banda del canal.

Con anticipación a la transmisión de datos, el sistema analiza las características de la línea asignando inicialmente igual número de bits por canal, lo que representa la capacidad de transmisión teórica y realiza una transmisión de diagnóstico [21]. Basándose en los resultados de relación S/N obtenidos en cada subcanal, por el transceptor se reasigna el número de bits a un número óptimo. En este momento, el sistema ya está preparado para la transmisión de datos. El transceptor continuará adaptándose a las características de la línea en las transmisiones siguientes.

Como fue antes mencionado, a cada subcanal le serán asignados diferentes números de bits, dependiendo de sus características de relación S/N. Los bits asignados a cada subcanal son codificados al símbolo correspondiente en la constelación QAM en términos de sus características espectrales. En VDSL el número de bits que puede transportar cada subcanal varía entre 0 y 15 bits. El número total de bits que pueden transmitirse a través del canal se determina con la siguiente ecuación:

$$b_{\text{total}} = \sum_{i=n_1}^{n_2} b_i \quad (2.1)$$

Donde  $b_i$  es el número de bits en el  $i$ -ésimo subcanal,  $n_1$  y  $n_2$  son números de bits determinados para el canal de subida y para el de bajada. Con la ecuación 2.1 se puede determinar la capacidad del canal según las fórmulas de Shannon:

$$C = \sum_{n_1}^{n_2} \Delta f \times b_i = 4,3125 \times b_{\text{total}} \quad (2.2)$$

Donde  $\Delta f$  es el ancho de banda de cada subcanal, en este caso particular, VDSL

es igual a 4,3125 KHz. Una vez que el transceptor se ha inicializado, cada subcanal o subportadora conteniendo los bits determinados por la tabla de asignación se modulan utilizando la técnica de modulación digital QAM. La colección de  $N$  símbolos QAM (información en el dominio de la frecuencia, correspondiente a los  $N$  tonos, con su correspondiente fase y amplitud en formato digital) en el siguiente bloque del transmisor, son procesados en paralelo mediante la Transformada Rápida Inversa de Fourier, IFFT y a su salida se entregan muestras en el tiempo de señales reales.

ANSI [5] seleccionó la técnica de modulación DMT para formar parte del estándar de las redes VDSL por 3 razones:

- Mejor rendimiento en la transmisión QAM.
- Aplicación más fácil que el método de modulación sin portadoras CAP.
- Flexibilidad del ancho de banda en la demanda.

#### **2.2.4. Mapeo de constelación**

La modulación QAM es una modulación digital en la que el mensaje está contenido tanto en la amplitud como en la fase de la señal transmitida. Se basa en la transmisión de dos mensajes independientes por un mismo y único camino. Esto se consigue modulando la misma portadora, desfasada entre uno y otro mensaje. Esto supone la formación de dos canales ortogonales en el mismo ancho de banda, con lo cual se mejora en eficiencia del ancho de banda.

La constelación de la codificación es un proceso que es independiente para cada tono. El número de puntos para cada tono depende del número de bits que se asignan a cada tono. Los tonos más pequeños con dos bits pueden dar cuatro puntos de la constelación diferentes y el mayor con 15 bits pueden dar 32.768 puntos diferentes.

Entonces  $b$  bits dan  $2^b$  puntos de constelación. Si  $b$  es un número par entonces sólo se toman los puntos de constelaciones de  $b-2$  y cambiar todos los puntos a cuatro puntos nuevos. Si  $b$  es impar, es más difícil, pero la idea es no tener los puntos de constelación más lejos de lo necesario. En la Figura 2.4 se muestran las constelaciones de ejemplo para  $b = 2$  y  $b = 4$

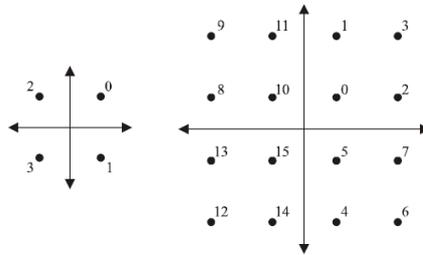


Figura 2.4: Constelaciones para  $b = 2$  y  $b = 4$  [3].

### 2.2.5. Turbo Decodificador

La estructura general del turbo decodificador se encuentra formada por dos decodificadores en serie, los cuales se encuentran separados por una etapa de desintercalado e intercalado de la información entre ellos. Teniendo a su vez una retroalimentación de la información de salida del segundo decodificador hacia el primero (razón por la que fueron llamados turbo códigos), lo cual produce una ganancia en la decodificación al tener un cierto número de iteraciones.

El algoritmo utilizado para la decodificación iterativa es el algoritmo modificado BCJR [12] de máxima probabilidad a posteriori (MAP) en honor a sus creadores: Bahl, Cocke, Jelinek y Raviv o algoritmo “*forward-backward*” publicado en el año 1974 [22]. Aunque en ese entonces la importancia del algoritmo no fue muy resaltada debido a que presentaba el mismo grado de complejidad que otros algoritmos de decodificación iterativa, fue hasta 1993 cuando fue utilizado por Berrou, Glavieux y Thithimajshima en 1996 para la decodificación de los turbo códigos y se mostraron sus verdaderos alcances [17]. En la Figura 2.5 se ilustra la estructura general de un turbo decodificador utilizando el algoritmo MAP.

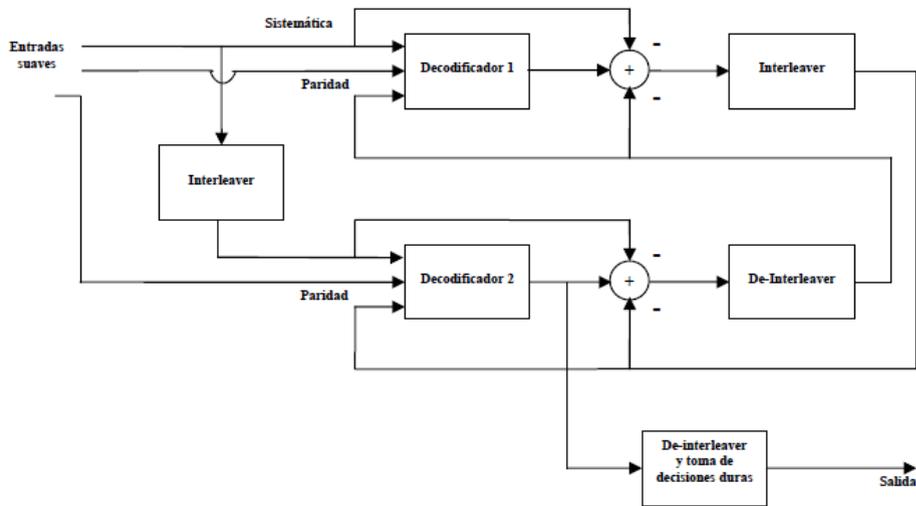


Figura 2.5: Esquema del Turbo decodificador [2].

Haciendo énfasis que dicha estructura (figura 2.5) corresponde a la de un codificador turbo de tasa 1/2, y representa el tipo de decodificador iterativo utilizado más comúnmente. Cada decodificador tiene dos entradas: los bits de salida del canal codificados sistemáticamente, y la información de los bits de paridad generados por los codificadores correspondiente en el turbo codificador (información a priori).

Los decodificadores tienen que entregar salidas *soft*, salidas suaves, valiéndose de las entradas provenientes del canal y de la información entregada por el otro decodificador. Estas salidas *soft* se representan generalmente mediante la relación LLR (tasa de similitud logarítmica), que dan información sobre el signo de cada bit y la probabilidad de una. La LLR para el valor de un bit decodificado está dada por:

$$L_{u_k} = \ln \left[ \frac{P(u_k = +1)}{P(u_k = -1)} \right] \quad (2.3)$$

Donde  $P(u_k = +1)$  es la probabilidad de que el bit decodificado sea un "1" binario, mientras que  $P(u_k = -1)$  representa la probabilidad de que se trate de un "0" binario. Una LLR positiva y alta indicará una probabilidad alta de que se trate de un 1 y una negativa (por tratarse de un logaritmo) y alta indicará una probabilidad alta de que se trate de un 0.

Existen dos algoritmos de decodificación SISO como se puede observar en la siguiente Figura 2.6, los algoritmos MAP (*Maximum a posteriori*) o algoritmos que minimizan la probabilidad de error de símbolo; y los algoritmos SOVA (*Soft Output Viterbi Algorithm*) o algoritmos que minimizan la probabilidad de error de palabra o secuencia.

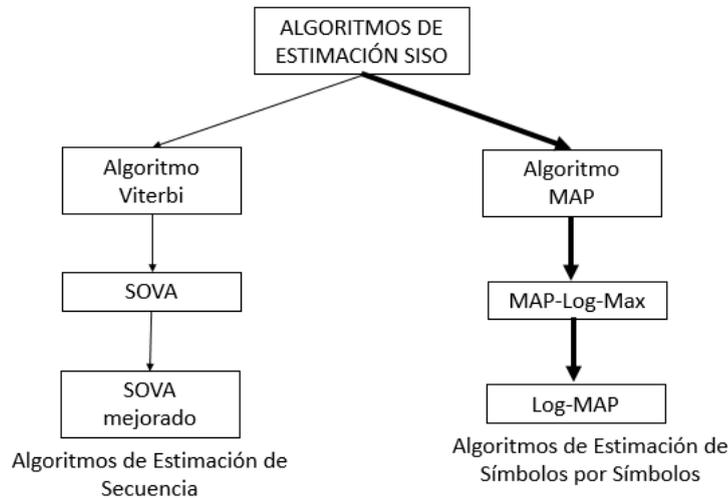


Figura 2.6: Esquema de los tipos de algoritmos de decodificación.[2].

La complejidad del algoritmo MAP se debe a que debe efectuar una enorme cantidad de multiplicaciones y adiciones por cada estimado que produce la probabilidad a-posteriori de cada bit de información decodificado. Este problema se aminora si el algoritmo entero se lleva a cabo en el dominio logarítmico, con el objetivo de poder tener resultados lo más cercanos posibles con menor complejidad en el tipo de operaciones y a la vez una menor latencia en el sistema, así no se espera hasta el último paso para tomar el logaritmo de la LLR.

Por los requerimientos de hardware, el algoritmo MAP no es práctico para ser implementado, es por ello entonces que la versión logarítmica del algoritmo MAP y el algoritmo SOVA, son los algoritmos de decodificación que son utilizados en el presente trabajo. Estos algoritmos son menos sensibles a la relación S/N y un poco más exactos a la estimación de la varianza de ruido. El principal beneficio es que, en el dominio logarítmico la multiplicación se convierte en suma. Esta es la idea principal de los algoritmos Max-Log-MAP y Log-MAP para reducir la complejidad.

Ahora bien, la diferencia entre los algoritmos Max-Log-MAP y Log-MAP radica en la forma en que efectúan la operación de suma en el dominio logarítmico. En el dominio logarítmico, el algoritmo Max-Log-MAP efectúa la operación de suma como  $\ln(e^x + e^y) \approx \max(x, y)$ , mientras que el algoritmo Log-MAP efectúa dicha operación como  $\ln(e^x + e^y) = \max(x, y) + f_c(|y - x|)$  donde  $f_c(|y - x|)$  es término de corrección. Por esto, el desempeño del algoritmo Max-Log-MAP en comparación con el MAP es sub-óptimo debido a que utiliza una aproximación para efectuar las operaciones de suma en el dominio logarítmico. El algoritmo Log-MAP fue propuesto para corregir dicha aproximación y su desempeño es similar al del algoritmo MAP.

### 2.3. Interleaver, Permutador, Intercalador

El permutador consiste en tomar un bloque de N-bits de datos de entrada y reordenarlos en un orden pseudoaleatorio, el cual debe ser conocido en el codificador y el decodificador. El objetivo del intercalador es descorrelacionar las entradas a los codificadores constituyentes y también que los bits de chequeo de paridad sean independientes. Esto mejora la probabilidad de corregir un patrón de error por alguno de los decodificadores y reduce la cantidad de errores en ráfaga generados en la transmisión de la información [10].

#### 2.3.1. Interleaver de Bloque o Matricial

Es el permutador más usado frecuentemente en los sistemas de comunicación. El escribe en modo de columna de arriba abajo y de izquierda a derecha y lee en modo de fila de izquierda a derecha y de arriba abajo, es decir, un arreglo en forma matricial de tamaño  $M1 \times N1$  [18]. Todos los elementos de la matriz deben ser llenados con un dato y la operación de permutar los datos en esta forma generará un retardo de  $M1 \times N1$  intervalos. Una forma de explicar la memoria del proceso de este permutador es suponiendo la siguiente secuencia de símbolos:

A B C D E F G H I J K L M N O P

A	F	K
B	G	L
C	H	M
D	I	N
E	J	O

De acuerdo con su definición, la salida de este intercalador estará dada por:

A F K B G L C H M D I N E J O

### 2.3.2. Interleaver Espiral o Helicoidal

Un interleaver de este tipo consiste en una memoria en que la información es escrita como en el ejemplo anterior pero leída de forma diagonal, entonces la salida estará dada por:

E I M B F O D H L A J N C G K

### 2.3.3. Interleaver pseudo-aleatorio

Como su nombre lo indica, la función de este intercalador es tomar cada bloque entrante de N-bits de datos y re=arreglarlos en una forma pseudo-aleatoria previa a la codificación por el segundo codificador, este intercalador usa una permutación pseudo-aleatoria y mapea la secuencia de entrada de acuerdo al orden de permutación [8].

Una forma de explicar el proceso del permutador es suponiendo una secuencia de símbolos como la siguiente:

A B C D E F G H I J K L M N O P

Supóngase que ocurre un error en ráfaga y se dañan los símbolos contiguos DEFG. La secuencia recibida será:

A B C X X X X H I J K L M N O P

Donde las X representan a los símbolos dañados. El decodificador del receptor no será capaz de corregir ese error. Ahora bien, si en el proceso de intercalado en el codificador se reordena la secuencia anterior en forma de una matriz que se escribe fila a fila con la información de entrada, por ejemplo:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

El intercalador entrega ahora la información, pero leyendo la matriz columna a columna, es decir:

A	E	I	M
B	F	J	N
C	G	K	O
D	H	L	P

Así, la secuencia de símbolos entregada al codificador interno será:

A E I M B F J N C G K O D H L P

La entrada al decodificador del receptor será la secuencia anterior dañada por el error en ráfaga:

A E I X X X X N C G K O D H L P

En el decodificador esta secuencia se reordena a la forma de la secuencia en el codificador, antes del intercalado, con lo que se tiene:

A X C D E X G H I X K L X N O P

Como se puede apreciar, los símbolos erróneos en la secuencia recibida y reordenada están ahora aislados y la probabilidad de corregirlos es considerablemente mayor.

#### 2.3.4. Interleaver Convolutional

Un intercalador convolutional se compone de un conjunto de  $N=I-1$  registros multiplexados de forma que cada registro consecutivo contiene  $M$  símbolos más que el registro anterior. El primer registro no tiene ningún retardo y los datos son instantáneamente multiplexados, a diferencia de los otros registros, los cuales son multiplexados en cada instante de tiempo por lo cual este tipo de intercalador también es conocido como intercalador multiplexado. El intercalador convolutional se define mediante dos parámetros: la longitud de bloque del entrelazador,  $I$ , y la profundidad de entrelazado,  $D$ .

El entrelazado convolutional introduce un retardo de lectura a escritura,  $\Delta_j$ , que aumenta linealmente con el índice de octetos dentro de un bloque de  $I$  octetos:

$$\Delta_j = (D - 1) \times j \quad (2.4)$$

donde  $j = 0, 1, 2, 3, 4, I - 1$

En la Figura 2.7 se muestra un ejemplo para tratar de explicar de una manera más fácil el funcionamiento del intercalador convolutional, se utiliza el parámetro  $I = 7$ . Se implementan  $I$  ramas paralelas (numeradas de  $0 \dots I-1$ ) con un aumento de retardo de  $M$  octetos por rama. Cada rama representa un registro de desplazamiento que da servicio al primero que llega FIFO (línea de retardo) con una longitud  $0 \times M \dots (I-1) \times M$  bytes. El desentrelazador es similar al entrelazador, pero los índices de las ramas se invierten de modo que el retardo más grande del entrelazador corresponda al retardo más pequeño del desentrelazador. La sincronización del desentrelazador se logra encaminando el primer byte de un bloque de  $I$  bytes del entrelazador en la rama 0.

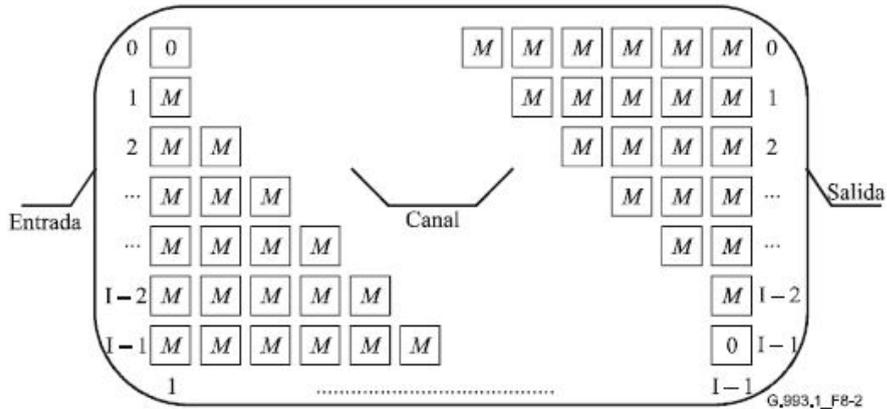


Figura 2.7: Ejemplo de implementación de interleaver convolucional [3].

## 2.4. El canal

### 2.4.1. Canal AWGN

En comunicaciones se denomina ruido a toda señal eléctrica no deseada que se mezcla con la señal útil que se desea transmitir. Es el resultado de diversos tipos de perturbaciones que tiende a enmascarar la información cuando se presenta en la banda de frecuencias del espectro de la señal, es decir, dentro de su ancho de banda, limitando la capacidad del receptor a hacer una decisión correcta de cada símbolo.

Con relación a lo anterior, el ruido proviene de una variedad de fuentes provocadas por el hombre o por la naturaleza. El ruido provocado por la naturaleza incluye elementos como son el sol, la atmósfera, entre otras fuentes. Empleando buenas técnicas y métodos se puede eliminar la mayoría del ruido o efectos indeseables provenientes de ambas fuentes, pero hay una fuente de ruido natural que no puede ser eliminado, llamado ruido térmico o ruido de Johnson, causado por el movimiento de electrones [23].

Partiendo de lo antes mencionado, el ruido térmico se describe como un proceso aleatorio Gaussiano con media cero. Un proceso Gaussiano  $n(t)$  es una función

aleatoria cuyo valor  $n$  en cualquier tiempo  $t$  arbitrario es estadísticamente caracterizado por la función de densidad de probabilidad (pdf) Gaussiana, que al ser estandarizada se tiene media cero y desviación estándar  $\sigma = 1$ , por lo que la pdf se expresa por:

$$p(r) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(r-s)^2}{2\sigma^2}\right] \quad (2.5)$$

Donde  $\sigma^2$  es la varianza de  $n$ . El ruido Gaussiano es una variable aleatoria normalmente distribuida como se mostró en la ecuación (2.5)

Debido a que el ruido térmico está presente en todos los sistemas de comunicación y es la fuente de ruido predominante en la mayoría de sistemas, es el modelo frecuentemente empleado en sistemas de comunicación, sus características son:

- *Gaussiano*: como se presentó con anterioridad las variables aleatorias estadísticamente independientes se aproximan a la distribución Gaussiana.
- *Blanco*: la principal característica del ruido térmico es que la densidad espectral de potencia es la misma para todas las frecuencias de interés en la mayoría de los sistemas de comunicación.
- *Aditivo*: significa que el ruido simplemente es superpuesto o agregado a la señal. Por lo tanto el modelo del ruido empleado en este trabajo de investigación es el canal AWGN.

#### 2.4.2. Tasa de Error de Bit y $E_b/N_0$

La Tasa de Error de Bit (BER) se define como la probabilidad de que el bit de información decodificado proveniente de la salida del decodificador presente error. La probabilidad de error debe de ser tan baja como sea posible teniendo en cuenta las restricciones del ancho de banda y potencia [14]. Para obtener la BER se emplea la fórmula:

$$\text{BER} = \frac{\text{Números de bits erróneos}}{\text{Números de bits enviados}} \quad (2.6)$$

Cuando se Turbo codifica  $k$  bits a la entrada se tiene  $n$  bits o símbolos a la salida, se tiene una tasa de codificación  $r = k/n$ . La energía por símbolo o ramificación de

la palabra codificada transmitida, denotada por  $E_s$  sería un factor de la energía por bit de información

$$E_b = \frac{E_s}{r} \quad (2.7)$$

Los sistemas de comunicación que utilizan técnicas de codificación de canal, son medidos a menudo por su probabilidad de error. Esta probabilidad de error se puede expresar en el receptor en términos de la relación de energía por bit  $E_b$ / densidad espectral de potencia de ruido  $N_0$ , que expresa la tasa de señal a ruido o *Signal to Noise Ratio* (SNR) denotada en decibeles (dB), esto significa que  $10\log E_b/N_0$  es una medida normalizada de SNR, también conocida como SNR por bit. Por lo tanto la ecuación (2.7) puede reescribirse como:

$$E_b/N_0 = \frac{E_s}{r \cdot N_0} \quad (2.8)$$

Una de las métricas más importantes del rendimiento en sistemas digitales es la gráfica de la probabilidad de error de bit contra  $E_b/N_0$ .

## 2.5. VDSL

Se trata de una tecnología de acceso a internet de Banda Ancha, perteneciente a la familia de tecnologías xDSL que transmiten los impulsos sobre pares de cobre [24]-[25]. Desde el punto de vista tecnológico, VDSL puede considerarse como la sucesora de ADSL [4]. VDSL puede transportar datos de video y de otros tipos de tráfico a velocidades de hasta 58 Mbps, de cinco a diez veces superiores a ADSL. Adicionalmente, al instalarse de forma simétrica o asimétrica, se adapta mejor a las exigencias del mercado.

VDSL ofrece a los usuarios residenciales video de una calidad superior al transmitido mediante difusión, junto con tráfico de Internet y las habituales llamadas telefónicas de voz. Se pueden ofrecer simultáneamente varias películas (en difusión o bajo petición). En el entorno de oficinas, VDSL satisface la demanda, siempre creciente, de acceso de datos más rápido y hace realidad, por ejemplo, las llamadas de videoconferencia de gran calidad entre varias localidades. Entre las aplicaciones

comerciales típicas que VDSL puede soportar, se encuentran la interconexión de VPN y LAN.

Debido a las limitaciones de distancia, VDSL será suministrada a menudo desde un gabinete situado en la calle equipado con una fibra óptica conectada a la red backbone. Esta topología, es la FTTCab y se muestra en la Figura 2.8 [4].

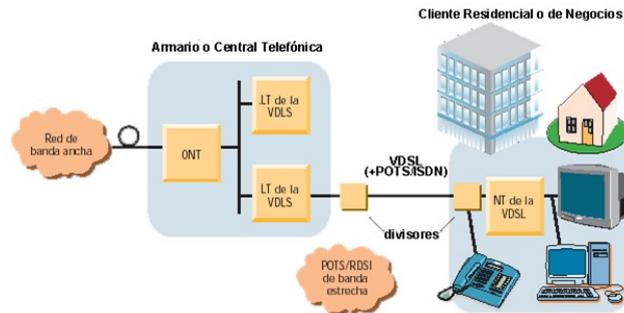


Figura 2.8: Topología de VDSL [4].

### 2.5.1. Tipos de Servicios VDSL

- Simétrico

VDSL ha sido diseñado para el envío al usuario de servicios de banda ancha asimétricos, incluyendo difusión digital de TV, video bajo demanda (VoD), acceso a Internet de alta velocidad, aprendizaje a distancia, telemedicina, entre otros. El envío de estos servicios requiere que el canal de bajada tenga mayor ancho de banda que el canal de subida por lo que es asimétrico. Por ejemplo, HDTV requiere 18 Mbps para la bajada del video contenido, sin embargo, en la subida solo requiere el envío de información de señalización (ej. cambio de canal o selección de programas), la cual está en el orden de los Kbps. Las velocidades en sentido descendente son submúltiplos de la velocidad básica de los sistemas SONET y SDH de 155.52 Mbps, ellas son: 51.84, 25.92 y 12.96 Mbps [3].

La distancia sobre la que pueden utilizarse tales velocidades está limitada

debido a limitaciones físicas, principalmente la elevada atenuación con la frecuencia de los pares trenzados. Generalmente, VDSL funcionará en líneas de longitud inferior a 1.5 Km.

- Asimétrico

VDSL también ha sido diseñado para proveer servicios simétricos para clientes de negocios pequeños y medianos, como, aplicaciones de datos de alta velocidad, aplicaciones de video de teleconferencia y teleconsulta, entre otras. El VDSL simétrico puede ser utilizado para proveer circuitos  $n \times T1$  de corto alcance. Aunque ANSI no especifica la distancia y velocidades para servicios simétricos de alto rango, se soportan lazos desde 900 m a 3000 m a velocidades desde 6 Mbps a 1.5 Mbps [4].

### 2.5.2. Bandas de Frecuencias implementadas en VDSL

Los sistemas G.993.1 [3] utilizan el plan 998, implementado en Norte América y en Japón, de cuatro bandas para la transmisión que comienza a 138 kHz y se extiende hasta 12 MHz. Las cuatro bandas de frecuencia designadas por DS1, US1, DS2, y US2, corresponden respectivamente a la primera banda en el sentido descendente, la primera banda en el sentido ascendente, la segunda banda en el sentido descendente, y la segunda banda en el sentido ascendente, como se muestra en la figura 2.9, serán atribuidas de acuerdo con las frecuencias de separación de banda  $f_0 = 25\text{KHz}$ ,  $f_1 = 138\text{KHz}$ ,  $f_2 = 3.75\text{MHz}$ ,  $f_3 = 5.2\text{MHz}$ ,  $f_4 = 8.5\text{MHz}$  y  $f_5 = 12\text{MHz}$ .

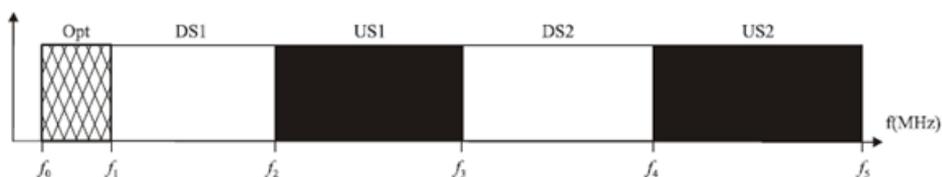


Figura 2.9: Bandas de frecuencias del plan 998. [3]

## 2.6. Teoría Combinatoria

La Teoría Combinatoria es la rama de las matemáticas que se ocupa del estudio de las formas de contar [26]. Estudia las agrupaciones que pueden ser formadas cuando se toman todos, o algunos, de los elementos de un conjunto finito. Los elementos del conjunto pueden ser de cualquier naturaleza: números, personas, empresas, artículos producidos por una fábrica, entre otros. La teoría combinatoria estudia especialmente el número de agrupaciones que pueden ser obtenidas bajo algún modo de composición de los elementos. Para ello, distingue básicamente tres conceptos: arreglos, permutaciones y combinaciones. Para el presente trabajo especial de grado se enfocará el estudio en las permutaciones el cual es el punto de interés para el diseño de un interleaver.

### 2.6.1. Permutaciones

Dado un conjunto de  $n$  elementos, llamaremos permutación de  $n$ , denotado por,  $P_n$ , a cada forma de ordenar los  $n$  elementos dados [27]; en las permutaciones se caracterizan por:

- No se descarta ningún elemento del conjunto  $n$ .
- El orden es importante.
- No se repite ningún elemento.

Por lo antes mencionado, el número de permutaciones formadas con un conjunto de  $n$  elementos coincide con su factorial:

$$P_n = n! \quad (2.9)$$

Por ejemplo la permutación de tres elementos,  $a, b, c$  será:

$$P_3 = 3! = 3 * 2 * 1 = 6 \quad (2.10)$$

Entonces, el resultado de una permutación de tres elementos es, seis conjuntos:

$$a, b, c \rightarrow \text{Permutaciones} \begin{cases} (a, b, c)(a, c, b) \\ (b, a, c)(b, c, a) \\ (c, a, b)(c, b, a) \end{cases} \quad (2.11)$$

## Capítulo III

# Procedimientos de la investigación

Con el fin de desarrollar un interleaver para los Turbo Códigos en las redes VDSL, el procedimiento metodológico que se llevó a cabo fue dividido en las siguientes cuatro etapas, con el propósito de cumplir con cada objetivo específico y por ende alcanzar el objetivo general.

### 3.1. Etapa I. Revisión Bibliográfica

En esta primera etapa se realizó una revisión exhaustiva en trabajos de investigación, texto de consulta y documentos publicados en Internet acerca de la estructura y funcionamiento de los Turbo Códigos y las redes VDSL, también sobre arquitecturas de permutadores y simulaciones del rendimiento en términos de BER. Finalmente se seleccionó para trabajar la estructura del Turbo Código en paralelo (PCCC) puesto que es la más usada y la que alcanza mejores resultados en términos de BER. Esta selección se llevó a cabo bajo las recomendaciones de la ITU G.993.1 aprobada en el 2004 y rigiendo el Plan 998 de atribución de frecuencias, ya que es el usado en Norte América y en Japón [3]

Así mismo, un esquema básico en paralelo del turbo codificador sería el que se muestra en la Figura 3.1, como se observa, los bits de información son codificados

por ambos codificadores RSC. El primero de ellos los recibe en su orden original, mientras que el segundo los recibe permutados por el interleaver. Este bloque consiste en la permutación de los bits de entrada en un orden aleatorio o programable con el objetivo de aleatorizar la secuencia de información de manera tal que la secuencia de entrada al segundo codificador sea muy diferente a la secuencia de bit de información original.

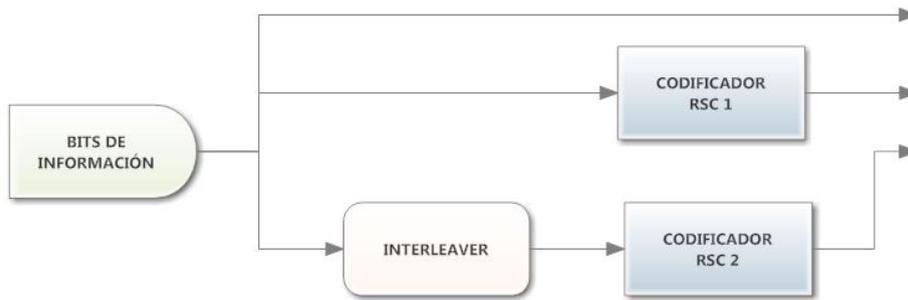


Figura 3.1: Esquema básico de un Turbo Codificador.

## 3.2. Etapa II. Diseño del permutador

Esta segunda etapa se inició con la selección del software utilizado para las simulaciones necesarias y seguidamente con el desarrollo del canal de una red básica VDSL analizando los componentes que lo conforman. Para cerrar esta etapa, se diseñó un permutador que permitiese mejorar el rendimiento del sistema en términos de BER con respecto a los resultados obtenidos por otros permutadores ya existentes.

### 3.2.1. Selección del software

Trabajar con software libre (GNU Radio) o Matlab es una decisión que en general depende de varios aspectos. Matlab es un lenguaje matemático interpretado que permite de forma sencilla simular sistemas de todo tipo. Ya sea mediante su

programación en un script o utilizando Simulink, de forma parecida, en software libre se puede trabajar en GNU Radio Companion.

Por otro lado, comparado con GNU Radio Companion, Matlab lleva utilizándose casi tres décadas en simulación de sistemas de comunicaciones. De forma que los toolboxes o librerías de funciones especializados en comunicaciones digitales ya disponen de todos los elementos o bloques necesarios para simular un sistema VDSL, incluir en ellos los Turbo Códigos, crear fácilmente el nuevo permutador y poder construir gráficas para evaluar el rendimiento del sistema, a diferencia de GNU Radio donde se requería crear algunos de los elementos que conforman la red, generando de esta forma trabajo adicional.

### 3.2.2. Simulación del canal de comunicación de la red VDSL

En esta sección se simuló el sistema transmisor y receptor de las redes VDSL, siguiendo las recomendaciones especificadas en la ITU Serie G.993.1 [3].

#### 3.2.2.1. Características del transmisor

El proceso de transmisión VDSL básicamente consta de las siguientes etapas:

1. Codificar la señal de entrada.
2. Modular la señal.
3. Calcular la IFFT de los símbolos complejos modulados.

El sistema VDSL creado, el cual se observa en la Figura 3.2, consta básicamente de los siguientes elementos:

- **Fuente Binaria:** El primer bloque que aparece en la Figura 3.2 es el bloque de información a transmitir. Este bloque genera números binarios aleatorios

usando una distribución de Bernoulli. La distribución de Bernoulli con parámetro  $p$  produce "0" con una probabilidad  $p$  y "1" con una probabilidad de  $1 - p$ .

- **CRC:** La comprobación de redundancia cíclica (CRC) es una secuencia corta adicional que sirve para el cálculo de suma de comprobación para detectar cambios accidentales en los datos. Es una de las herramientas más comunes y potentes para la detección de errores. En VDSL se utiliza el CRC-8 y viene dado por el polinomio:

$$G(x) = 1 + x^2 + x^3 + x^4 + x^8 \quad (3.1)$$

Esto significa que ocho bits se añaden a la trama y estos bits son usados en el receptor para decidir si la transmisión es correcta. La CRC sólo puede detectar errores, en consecuencia no puede corregir errores. La evaluación se realiza de la siguiente manera, la trama entrante se divide con el polinomio  $G(x)$  y luego el resto que aparece a continuación es el resultado de la operación de CRC. Para controlar el resultado, el símbolo recibido se evalúa exactamente del mismo modo que el transmitido. Estas dos respuestas se comparan y si son iguales se dice que es aceptable la transmisión.

- **Aleatorizador (*Scrambler*):**

Se utilizó un aleatorizador como el que se muestra en la Figura 3.3 con el fin de mezclar los bits para evitar largas ráfagas de unos o ceros por el canal, una mejor mezcla da un mejor rendimiento. Este dispositivo es del tipo autosincronizado de modo que la desaleatorización pueda llevarse a cabo sin necesidad de una alineación particular con la secuencia aleatorizada. Además, debe poder representarse mediante la ecuación que sigue, siendo  $m(n)$  una muestra de bits del mensaje tomada en el instante  $n$  y la salida del aleatorizador  $x(n)$  vendrá dada por:

$$x(n) = m(n) \oplus x(n - 18) \oplus x(n - 23) \quad (3.2)$$

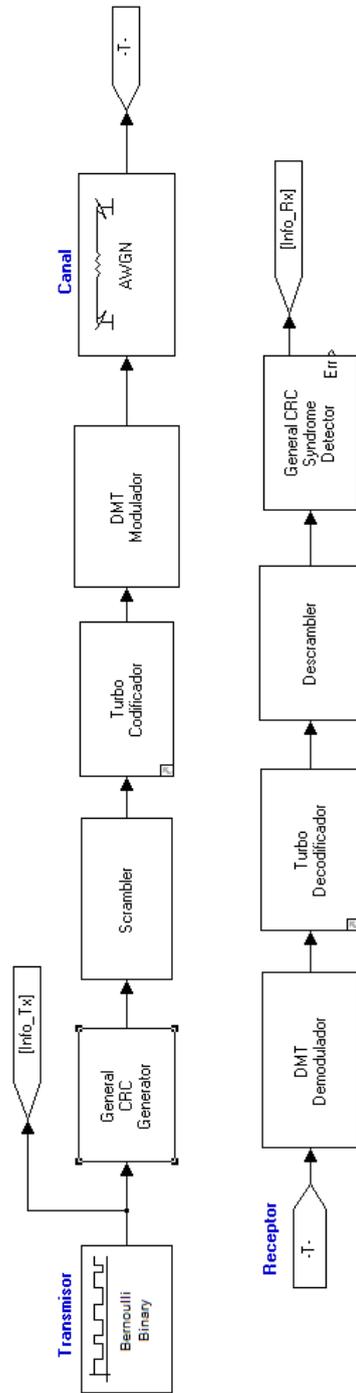


Figura 3.2: Esquema básico simulado de la red VDSL. Fuente propia.

Como se puede observar en la ecuación (3.2) toda la aritmética aplicada es módulo 2.

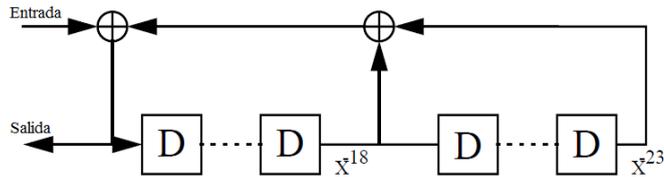


Figura 3.3: Estructura del aleatorizador [3].

- Turbo Codificador:** El subsistema que representa al Turbo Código que se observa en la Figura 3.2 está conformado por una estructura PCCC con dos bloques codificadores RSC y un bloque interleaver entre dichos codificadores como el que se muestra en la Figura 3.4.

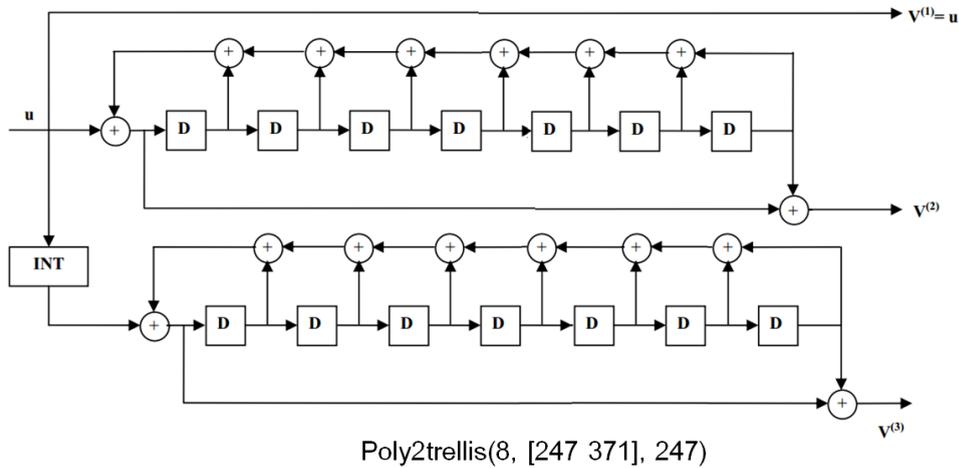


Figura 3.4: Estructura del codificador turbo. Fuente propia.

Para la definición de cada codificador convolucional en la herramienta de computo se hace uso de una función llamada `poly2trellis` y la misma se define a partir de tres parámetros:

`poly2trellis(Restricción de longitud, Generador de código, Conexión feedback)`

- *Restricción de longitud*: Es un valor entero positivo que indica el tamaño de la memoria del turbo codificador. En esta simulación es 8 y corresponde al número de retardadores más uno.
- *Generador de Código*: Es un polinomio que indica como se genera cada salida de los codificadores, indicando las conexiones entre la salida y los registros de memoria, se suele expresar de forma octal.

Hay muchas maneras de escoger los polinomios generadores para un Turbo Código, no todos los resultados en las secuencias de salida tiene buenas propiedades de protección contra errores pero, para el sistema simulado en este proyecto, estos se han escogido de acuerdo al estándar de comunicación VDSL.

En la tabla 3.1 se muestran los esquemas de generadores de códigos que son aceptados en el estándar de las redes VDSL, junto con los valores de restricción de longitud respectivamente, sabiendo que todos los componentes son independientes y que pueden ser aplicados a una amplia gama de constelaciones de señal y velocidades de código.

**Tabla 3.1:** Tabla de diferentes longitudes de restricción usadas en VDSL.[6]

k	$g_0(x)$	$g_1(x)$	$g_0(x)_{oct}$	$g_1(x)_{oct}$
4	$1 + x + x^3$	$1 + x + x^2 + x^3$	15	17
5	$1 + x^3 + x^4$	$1 + x + x^2 + x^4$	23	35
6	$1 + x^2 + x^4 + x^5$	$1 + x + x^2 + x^3 + x^5$	53	75
7	$1 + x^2 + x^3 + x^5 + x^6$	$1 + x + x^2 + x^3 + x^6$	133	171
8	$1 + x^2 + x^5 + x^6 + x^7$	$1 + x + x^2 + x^3 + x^4 + x^7$	247	371
9	$1 + x + x^2 + x^3 + x^5 + x^7 + x^8$	$1 + x^2 + x^3 + x^4 + x^8$	753	561

- *Conexión de Feedback*: Es la variable que nos indica que es un codificador recursivo, y su valor establece como se conecta la retroalimentación en el codificador.

Por otro lado, dentro del turbo codificador se encuentra el permutador o interleaver, el cual es un dispositivo que modifica la ubicación de los bits de información de una manera predefinida, a este elemento se le debe indicar el tamaño de la trama a permutar y el orden en que se van a aleatorizar los datos de información recibidos.

Seguidamente las tramas paralelas pasan por un bloque llamado Ordenador, el cual se encarga de organizar las tramas en una sola trama serial, que se encuentra compuesta por los bits sistemáticos, y las dos tramas de bits de paridad, una vez organizadas son finalmente enviadas a la fase de modulación a continuación.

- **Modulador DMT:** El subsistema correspondiente al modulador DMT está compuesto por 16 bancos moduladores que a su vez contiene cada uno 32 bloques moduladores QAM. Cada bloque representa un subcanal y cada uno confina la potencia dentro de una banda estrecha logrando un aislamiento espectral perfecto con separaciones con un ancho de banda igual a 4,3125 KHz. Ahora bien este sistema asigna dinámicamente el número de bits para cada subportadora, de modo que el rendimiento para cada subcanal sea máximo, adaptándose así a las condiciones del canal. Se suelen asignar un mayor número de bits a aquellos subcanales con mayor relación SNR y menor número de bits a aquellos con menor relación S/N.

En el apéndice [A](#) se muestra el vector de la cantidad de bits que se le asignó a cada subcanal para el sistema VDSL implementado en este trabajo.

Seguidamente, los bits asignados (valores entre 0 y 15) a cada subcanal son codificados al símbolo correspondiente en la constelación QAM (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 símbolos) en términos de sus características espectrales.

- **IFFT (*Inverse Fast Fourier Transform*):** Una vez se tienen todos los datos complejos codificados y distribuidos en frecuencia, el siguiente paso que establece el estándar es la aplicación de la IFFT con lo cual, a partir de este punto, se pasa a trabajar en el dominio temporal, es decir, valores reales.

Para que sea eficiente el algoritmo de la IFFT, el número de puntos con los que se debe trabajar tiene que ser potencia de dos, para este caso en particular se cumple dicho requisito pues se trabajó con 512 subcanales.

### 3.2.2.2. Características del canal

#### ■ AWGN:

Los bits que fueron modulados, pasan por un bloque AWGN, que simula las posibles alteraciones de los valores al pasar a través de un medio, este bloque adopta su tiempo de muestreo al de la señal bajo estudio.

Uno de los datos que debe ser ingresado al momento de la inicialización es el valor de la relación de  $E_b/N_0$  (energía de la señal por bit entre ruido del canal) en dB, que es uno de los parámetros que se utiliza para introducir ruido blanco en el canal.

### 3.2.2.3. Características del receptor

El proceso de recepción VDSL básicamente consiste en las siguientes etapas:

1. Calcular la FFT de los símbolos recibidos que luego serán demodulados.
2. Demodulación de la constelación compleja.
3. Decodificar la señal.

En este sentido, básicamente los bloques que constituyen el camino del receptor son los bloques inversos que se utilizaron en la fase de transmisión como se pueden observar en la Figura 3.2. A su vez cada uno fue configurado de la manera homóloga que en el transmisor. Comenzando por el bloque FFT, luego la trama pasa por la demodulación, seguidamente por los bloques que se encargan de la decodificación, el bloque desaleatorizador y por último por el detector de CRC.

Así mismo, se tiene que la entrada al receptor está compuesta por bits sistemáticos y bits de pariedad. Se debe distinguir y separar cada uno de los bits provenientes del primer y segundo codificador RSC, para trabajar con ellos en estimaciones futuras.

En la presente investigación; el turbo decodificador utilizó el mismo algoritmo en ambos decodificadores RSC, para decodificar la secuencia recibida. La LLR se emplea para simplificar el intercambio de información a priori en el proceso de decodificación entre componentes.

De manera particular en el subsistema del turbo decodificador aparece un nuevo parámetro a configurar y es *Número de iteraciones de decodificación*, el cual determina la cantidad de veces que se hacen las iteraciones en los decodificadores RSC y se selecciona el tipo de algoritmo de decodificación a utilizar entre: True APP (Probabilidad a posteriori), Max (Max-Log-MAP) y Max\* (Log-MAP). De acuerdo con las recomendaciones de la ITU [3] se implementa el algoritmo Max y se trabajo con un número de iteraciones igual a 6.

### 3.2.3. Diseño y simulación del permutador para los Turbo Códigos

Para realizar un diseño de permutador, sea cual sea su naturaleza, la obtención de resultados notables en términos de BER depende en gran manera de establecer los parámetros y estructura que caracterizan al mismo bajo el estándar del sistema de aplicación. Por esta razón, la actividad de diseño comenzó con una revisión de las características de los permutadores empleados en los distintos sistemas de codificación, haciendo énfasis en la red VDSL, para así establecer los parámetros característicos más relevantes, como el tamaño de la entrada N-bits de datos o tamaño del permutador.

En este sentido, no se puede afirmar que existe una fórmula o regla universal que permita diseñar un "mejor interleaver". Atendiendo a [19] se comenta que para un tamaño grande de bloque y una relación  $E_b/N_0$  alta, un permutador Pseudo-aleatorio ofrece un mejor desempeño que un permutador altamente estructurado como los permutadores convolucionales.

Ahora bien, de acuerdo con las características del canal simulado VDSL se decidió desarrollar un interleaver con características aleatorias en conjunto con los arreglos de estructura de un interleaver de características de renglón-columna. Basando esta decisión en los buenos aportes que ofrecen dichos intercaladores, los parámetros importantes en el diseño fueron:

- L: Tamaño del bloque a procesar. La longitud de la trama de entrada.
- *intrÍndices*: Especifica el tipo de función de mapeo que el bloque usará para ordenar o permutar la señal de entrada.
- Crows: Número de filas del arreglo matricial
- Ncolms: Número de columnas del arreglo matricial.

De igual manera que en el turbo codificador se implementa el interleaver diseñado, en el decodificador se debe hacer uso del mismo pero con una configuración inversa en sus parámetros con el fin de que re-ordene los bits que se están decodificando.

### **3.3. Etapa III. Verificación del funcionamiento del interleaver**

En esta etapa se procedió a realizar pruebas y verificar el funcionamiento del interleaver diseñado, *VDSL Interleaver*, para ello, se espera obtener un rendimiento en términos de BER relativamente aceptables.

Al correr la simulación, se compara la señal transmitida con la señal recibida, empleando para esta comparación el bloque Cálculo de tasa error, obteniéndose un BER altamente significativo, con respecto a los valores de BER que se estilan obtener en simulaciones de sistemas de comunicaciones.

A partir de las evidencias anteriores se necesitaron realizar cambios en el diseño *VDSL Interleaver*, y lo primero que se probó fue un ajuste óptimo en los valores de las variables de configuración del mismo, al no obtenerse cambios satisfactorios, se planteó por esta razón un cambio en la configuración del esquema y se llevó a cabo la implementación de un algoritmo.

Con objeto de permutar cada elemento del vector de entrada, se parte de la agrupación en forma matricial de dicho vector de entrada, seguidamente se implementan operaciones elementales matriciales en la misma, tales como, la permutación y cálculos matemáticos básicos a cada elemento que conforman dicha matriz.

Ahora bien, este algoritmo, que en realidad es una función, se ejecuta en la configuración de un bloque de interleaver general, de manera que la permutación de la variable de entrada sea realizada por el algoritmo llamando a la función.

**Algoritmo:** Algoritmo para la permutación del interleaver.

1. Tomar el vector de entrada y calcular la longitud  $K$  del mismo.
2. Calcular el número de filas  $R$ , para la matriz rectangular.
3. Determinar el número primo,  $p$  para ser utilizado en el cálculo del número de columnas,  $C$ , de la matriz rectangular.
4. Escribir la secuencia de bits de entrada en la matriz rectangular de dimensiones  $R \times C$ , llenando fila por fila.
5. Seguidamente se determinan dos vectores  $s$  y  $q$  los cuales serán utilizados luego para la permutación de la matriz rectangular fila por fila.
6. El proceso termina cuando todas las filas sean permutadas y luego se construye un vector de salida  $\text{vector}_p$  que contiene los mismo valores del vector de entrada, sin ninguna repetición pero ordenados de una nueva manera.

Finalmente se realizan pruebas de cálculo de rendimiento en términos de BER y se concluye que se consigue un diseño satisfactorio de permutación para los Turbo Códigos de la red simulada VDSL.

### **3.4. Etapa IV. Análisis y conclusiones**

En esta última etapa se realizaron pruebas en el sistema turbo simulado en la Etapa II, empleando 4 tipos diferentes de permutadores que son compatibles con el estándar VDSL, los resultados obtenidos de las curvas de rendimiento en términos de BER de las simulaciones se compararon con los obtenidos por el interleaver diseñado.

Los permutadores que fueron utilizados para las pruebas son los siguientes:

#### **3.4.1. Permutador aleatorio**

El permutador aleatoribloque reordena los bits de la entrada usando una permutación aleatoria. Se debe indicar el número de elementos que ingresarán al permutador para ser aleatorizados.

#### **3.4.2. Permutador matricial**

Este permutador realiza el entrelazado relleno una matriz con los símbolos de entrada fila por fila y luego enviar el contenido de la matriz a la columna por columna.

Los parámetros que condicionan a este permutador son:

- Número de filas de la matriz que se forma a partir del vector de entrada.
- Número de columnas de la matriz a formar.

#### **3.4.3. Permutador helicoidal**

El permutador helicoidal permuta los símbolos en la señal de entrada por su inclusión en una matriz en una forma helicoidal y luego envía las filas de la matriz para el puerto de salida.

Los parámetros que condicionan a este permutador son:

- Número de columnas en la matriz helicoidal.
- Tamaño del grupo: El tamaño de cada grupo de bits de entrada. La longitud de entrada es  $C \times N$ .
- Tamaño de paso de la matriz: El número de filas de separación entre los grupos de entrada consecutivos en sus respectivas columnas de la matriz helicoidal.
- Condiciones iniciales: Es un escalar con el que llena la matriz en las posiciones en las que el arreglo no tiene símbolos de entrada.

#### **3.4.4. Permutador convolucional**

El permutador convolucional permuta los símbolos de la señal de entrada internamente utilizando un registro de desplazamiento. Este dispositivo genera un retardo, éstos son múltiplos enteros no negativos de un valor entero fijo. El número de registros se define a partir del parámetro de filas del desplazamiento.

Con cada nuevo símbolo de entrada, un conmutador cambia a un nuevo registro y el nuevo símbolo se mueve mientras que el símbolo más antiguo de ese registro se desplaza hacia fuera. Cuando el conmutador alcanza el registro N-ésimo, en la siguiente nueva entrada, se devuelve al primer registro. Ahora bien se realizaron unas pruebas de configuración de parámetros siguiendo la Tabla 3.5 de valores que ofrece el estándar ANSI para las redes VDSL [5], seguidamente se seleccionaron los valores que mejor se ajustaban al sistema simulado y son los que se muestran a continuación:

Rate (kbps)	Interleaver parameters	Interleaver depth (J)	(De)interleaver memory size	Erasure correction	End-to-end delay
50x1024	$K = 72$ $M = 13$	937 blocks of 72 bytes	33228 bytes	3748 bytes 520 $\mu\text{s}$	9.23 ms
24x1024	$K = 36$ $M = 24$	865 blocks of 36 bytes	15120 bytes	1730 bytes 500 $\mu\text{s}$	8.75 ms
12x1024	$K = 36$ $M = 12$	433 blocks of 36 bytes	7560 bytes	866 bytes 501 $\mu\text{s}$	8.75 ms
6x1024	$K = 18$ $M = 24$	433 blocks of 18 bytes	3672 bytes	433 bytes 501 $\mu\text{s}$	8.5 ms
4x1024	$K = 18$ $M = 16$	289 blocks of 18 bytes	2448 bytes	289 bytes 501 $\mu\text{s}$	8.5 ms
2x1024	$K = 18$ $M = 8$	145 blocks of 18 bytes	1224 bytes	145 bytes 503 $\mu\text{s}$	8.5 ms

Figura 3.5: Tabla de parámetros del interleaver convolucional usado en VDSL [5].

Donde  $K = N$  para la nomenclatura usada en el presente trabajo de de grado, el parámetro  $J$  es la profundidad del registro, para la nomenclatura usada  $J = B$  y también en la Tabla 3.5 se puede visualizar los valores usados para definir el tamaño de la memoria.

Los parámetros que condicionan este permutador vienen dados por:

- Filas de registros de desplazamiento: Número de registros de desplazamiento que el bloque utiliza internamente.
- Longitud del paso del registro: Número de símbolos adicionales que se ajustan en cada registro de desplazamiento sucesivo, donde el primer registro contiene símbolos cero.

## Capítulo IV

# Análisis, interpretación y presentación de los resultados

### 4.1. Análisis del algoritmo del interleaver diseñado

Una forma de explicar el proceso de como trabaja del permutador diseñado es suponiendo una secuencia de símbolos (vector de entrada de dimensiones  $1 \times 50$ ) como la siguiente:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27  
28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

A partir de acá se determina  $K = 50$  que es la longitud del vector de entrada y se asigna un  $R = 5$  que será el número de filas de la matriz rectangular a formar; se define un vector  $T$  de dimensiones  $1 \times 5$  que utilizará de patrón para la permutación entre filas de la matriz rectangular.  $T=[5 \ 4 \ 3 \ 2 \ 1]$

Seguidamente se construye un vector  $p = 1 \times 52$  conformado por números primos a partir del 7 y partiendo de este vector se calcula  $C = 10$  que es el número de columnas de la matriz rectangular a formar. Entonces  $K < R(p + 1)$ , calculándose  $C$  como:  $C = p - 1$  si  $K \leq R(p - 1)$

$p=[7\ 11\ 13\ 17\ 19\ 23\ 29\ 31\ 37\ 41\ 43\ 47\ 53\ 59\ 61\ 67\ 71\ 73\ 79\ 83\ 89$   
 $97\ 101\ 103\ 107\ 109\ 113\ 127\ 131\ 137\ 139\ 149\ 151\ 157\ 163\ 167\ 173$   
 $179\ 181\ 191\ 193\ 197\ 199\ 211\ 223\ 227\ 229\ 233\ 239\ 241\ 251\ 257]$

Luego de dichos cálculos forma la matriz rectangular con los componentes del vector de entrada y de dimensiones  $R \times C$  y se muestra en la Tabla 4.1

**Tabla 4.1:** Matriz 1. Matriz rectangular  $C \times R$

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Se forman dos vectores o secuencias de símbolos, el primero de ellos un vector  $s$ , del mismo tamaño de  $C$ , formado por el cálculo del módulo después de la división, la secuencia  $s(j)$  con  $j=0, 1, 2, \dots, p-2$  para una permutación entre columnas como:

$$s(j) = (v \times s(j-1)) \bmod(p) \quad (4.1)$$

Siendo  $j = 1, 2, \dots, (p-2)$  y  $s(0) = 1$

El segundo vector o secuencia es  $q$  del mismo tamaño de  $R$  y se va formando de la siguiente manera: Se asigna  $q_0 = 1$  para que sea el primer número primo entero y los  $q_i$  enteros primos se determinan calculando el máximo común divisor, tal que:

$$\text{MCD}(q_i, p-1) = 1, q_i > 6yq_i > q(i-1) \quad (4.2)$$

Para  $i=1, 2, \dots, R-1$

Seguidamente, se permuta la secuencia  $q_i$  para crear el vector  $r$  tal que  $rT(i) = q_i$ , para  $i = 1, 2, \dots, R-1$  donde  $T$  es el patrón de permutación entre cada una de las filas que componen a la Matriz 1

$$s=[1\ 2\ 4\ 8\ 5\ 10\ 9\ 7\ 3\ 6]$$

$$q=[1\ 7\ 9\ 11\ 13]$$

$$r=[13\ 11\ 9\ 7\ 1]$$

Una vez formada la matriz rectangular se realiza una permutación a base de módulos después de la división entre el vector  $s$  y cada una de las 5 filas quedando la matriz como se muestra en la Tabla 4.2:

**Tabla 4.2:** Matriz 2. Matriz rectangular formada a partir del vector  $s$

8	9	6	4	10	3	2	5	7	1
12	14	18	15	20	19	17	13	16	11
26	23	27	29	30	25	28	24	22	21
37	35	32	33	40	34	36	39	38	31
42	44	48	45	50	49	47	43	46	41

Seguidamente se realiza una operación elemental de filas por filas de la matriz rectangular, utilizando el vector  $r$  resultando una matriz como se muestra en la Tabla 4.3:

**Tabla 4.3:** Matriz 3. Matriz rectangular permutada

42	44	48	45	50	49	47	43	46	41
37	35	32	33	40	34	36	39	38	31
26	23	27	29	30	25	28	24	22	21
12	14	18	15	20	19	17	13	16	11
8	9	6	4	10	3	2	5	7	1

Para finalizar el proceso de permutación se lee la matriz rectangular resultante columna por columna y se forma un vector  $vector\_p$  de salida de  $1 \times 50$  como el que se muestra a continuación:

```
vector_p= [42 37 26 12 8 44 35 23 14 9 48 32 27 18 6 45 33  
          29 15 4 50 40 30 20 10 49 34 25 19 3 47 36 28  
          17 2 43 39 24 13 5 46 38 22 16 7 41 31 21 11 1]
```

## 4.2. Variación del BER según el SNR

La mejor forma de verificar la fiabilidad del sistema diseñado es midiendo la probabilidad de error de bit (BER) para diferentes valores de relación señal a ruido (SNR). Según lo expresado en el capítulo II, utilizando permutadores se dispersan las ráfagas de errores causados por el canal de comunicación, por lo que el BER tiene que mejorar en comparación con un sistema que no utilice este dispositivo en su estructura de codificación.

Para verificar que existe una mejora en el BER para un sistema VDSL se realizan pruebas usando todo el sistema en conjunto, y además se analizara las variaciones que se puedan ocasionar en cada una de las etapas usadas en el diseño.

En cada gráfica se muestra el BER como una función de la energía de bit por ruido ( $E_b/N_0$ ), el cual es una medida de eficiencia de energía para los esquemas de modulación y codificación. Para todas las simulaciones se emplea el algoritmo MAX-Log-MAP, siendo el algoritmo utilizado en el turbo decodificador. Este algoritmo implica menor complejidad, menor número de recursos de memoria y retardo en cada iteración [16]

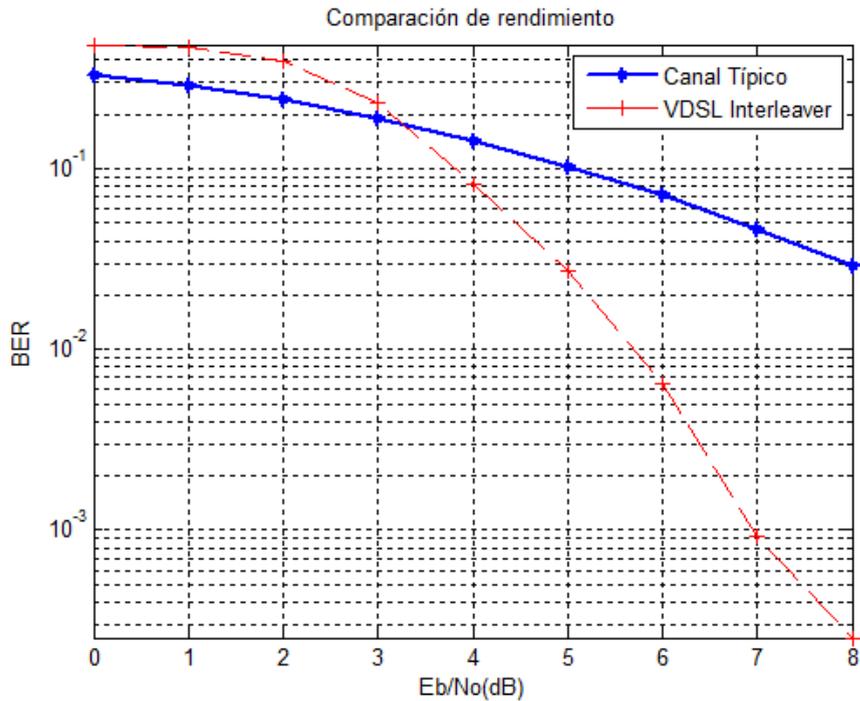
En las simulaciones se siguieron los pasos que se describen a continuación para obtener los resultados que se mostrarán en las siguientes secciones:

1. Obtención de la secuencia de entrada proveniente de la fuente de información.
2. Generación y adición de los bits redundantes CRC y aleatorización externa con el scrambler.
3. Turbo codificación empleando los códigos RSC concatenados en paralelo.

4. Reordenación de los bits, empleando un permutador a la entrada de uno de los codificadores convolucionales.
5. Modulación DMT utilizando 512 canales.
6. Generación y adición de ruido del canal AWGN a la secuencia modulada y codificada.
7. Demodulación DMT.
8. Decodificación iterativa de la información recibida empleando el algoritmo MAX-Log-MAP.
9. Cálculo del BER.

En el paso 4 es donde se ajustan los cambios de las diferentes pruebas realizadas en los parámetros de los permutadores, con el objeto de evaluar su comportamiento en conjunto con el sistema de la red VDSL. En concreto se realizaron una serie de simulaciones, con distintos permutadores, en las que el rendimiento en términos de BER varía para cada una de ellas.

### 4.3. Comparación de BER entre el canal típico VDSL y canal VDSL con interleaver el diseñado

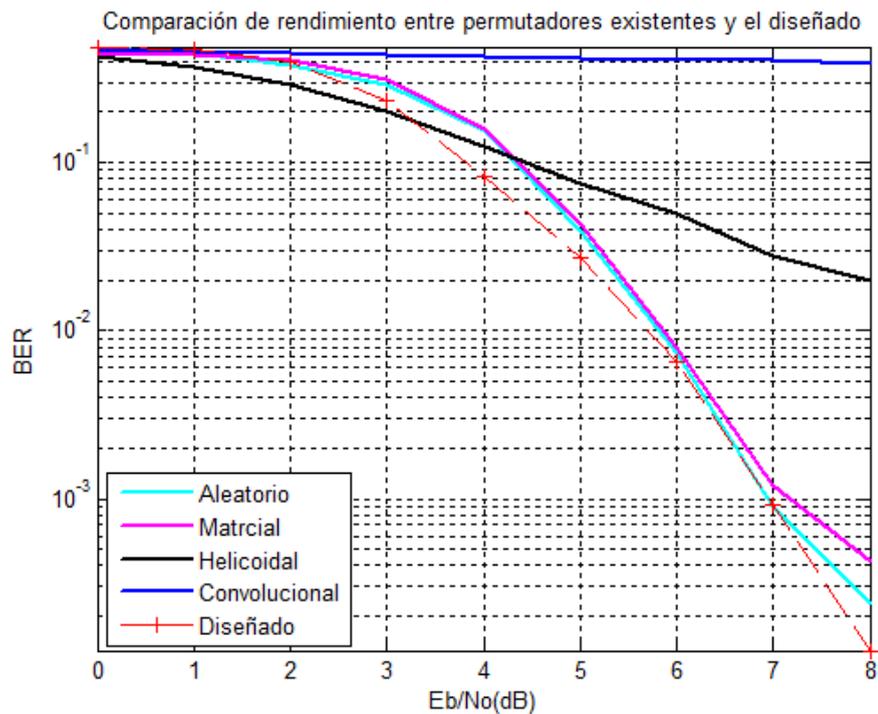


**Figura 4.1:** Curva del BER para el canal VDSL con interleaver genérico y el diseñado. Fuente propia.

La Figura 4.1 muestra las curvas de rendimiento de un sistema típico VDSL usando un permutador genérico y el mismo sistema pero usando el interleaver VDSL diseñado. La curva roja que representa el BER del canal VDSL implementando el permutador diseñado muestra un mejor rendimiento que la curva azul, la cual representa el uso de un permutador genérico como lo establece el estándar ANSI [5]. De lo antes mencionado se concluye que el interleaver diseñado mejora el rendimiento del canal en términos de BER.

#### 4.4. Comparación de BER usando diferentes permutadores en una canal VDSL

Luego de realizar las pruebas pertinentes, planteadas en los objetivos específicos, se encuentra que todas las etapas del sistema VDSL funcionan en base al estándar IEEE 802.03.ah y las pruebas realizadas en cómputo demuestran que el interleaver diseñado logró mejorar el rendimiento en términos de BER del sistema.



**Figura 4.2:** Comparación de rendimiento usando diferentes permutadores. Fuente Propia.

En la gráfica mostrada se aprecian las diferentes curvas de BER vs  $E_b/N_0$  si al canal VDSL se le aplican diferentes tipos de permutadores, como se puede observar, el permutador que ofrece un mejor rendimiento es el permutador diseñado *VDSL Interleaver*. En la Figura 4.2 se evidencia también que los interleaver del tipo convolutcional no presentan un buen rendimiento en un sistema VDSL, puesto que son permutadores altamente estructurados, en cambio los interleaver tipo bloque

dan un mejor rendimiento ya que su diseño se basa en una configuración pseudo-aleatoria, tal como lo explica [13].

## Capítulo V

# Conclusiones y recomendaciones

### 5.1. Conclusiones

El principal objetivo de este proyecto era diseñar y simular un interleaver para los Turbo Códigos en las redes VDSL, para ello se utilizó una herramienta de cómputo avanzado para realizar las pruebas de rendimiento en términos de BER. Con base a los resultados obtenidos en el desarrollo de cada una de las etapas de la investigación se concluye que:

En base a las simulaciones realizadas se concluye que el interleaver diseñado para los turbo códigos permitió obtener una tasa de rendimiento aceptable para los sistemas VDSL. Así mismo, de las gráficas en la que se comparan los diferentes permutadores, se observa como a partir de cierto valor de señal/ruido el sistema simulado con el interleaver diseñado mejoró la curva de BER con respecto a los otros, indicando que se logró un diseño que cumpliera con los objetivos planteados.

Así mismo, la estructura de permutador diseñado que logró ofrecer mejores prestaciones fue aquel que combinó técnicas empleadas en los permutadores matriciales, como lo son reorganización de filas y columnas unido con ciertos esquemas de lectura de los datos o elementos que conforman la misma, ya que ayudó a mejorar las propiedades de distancia del código, permitiendo de esta forma aumentar

el numero de errores corregidos.

Las características del Turbo Código que mejor se ajustaron al sistema VDSL fue aquella con codificadores convolucionales RSC definido por [8 (247, 371), 247] y algoritmo de decodificación MAX- Log-Map. En el caso del codificador se ensayó el sistema con todos los esquemas de codificadores RSC indicadas en las recomendaciones ITU [3] y el estándar ANSI [5].

Por otro lado, el desarrollo de este interleaver y la modelación de todo el sistema VDSL aporta beneficios académicos, ya que puede ser usado para entender el proceso de transmisión y recepción de los canales de comunicaciones digitales específicamente las redes VDSL y de este mismo poder evaluar el rendimiento del sistema cuando se realizan variaciones a cada uno de los elementos que lo conforman.

En relación con el desarrollo de los objetivos planteados, los mismos se cumplieron usando una herramienta de cómputo avanzado, el software que mejor se adaptara a la presente investigación, una herramienta de cómputo especialista en entornos de sistemas de comunicación. Se utilizaron bloques para simular la red típica VDSL , al igual que para el diseño, creación y simulación del Interleaver VDSL.

En virtud de los resultados, se alcanzaron en su totalidad los objetivos planteados, identificando en el camino desempeño, alcance y posibles mejoras en el sistema.

## 5.2. Recomendaciones

Con base en el estudio realizado y a los resultados obtenidos, como mejoras al presente trabajo de grado, se recomienda:

- Implementar el diseño y la simulación del interleaver para los turbo códigos en las redes VDSL en una de las familias de las tarjeta FPGA aplicando señales en tiempo real.
- Utilizar un software libre como herramienta principal de cómputo.
- Realizar un análisis comparativo del comportamiento del interleaver diseñado bajo una estructura serial de Turbo Código (SCCC).

Asimismo, se recomiendan otros estudios futuros en los que pueda resultar de base la presente investigación.

- Diseñar y simular un interleaver para los Turbo Código de las redes VDSL2.
- Implementar en el turbo decodificador otro algoritmo de secuencia de estimación de símbolos por símbolos.
- Implementar y probar el rendimiento en términos de BER el interleaver diseñado en otras técnicas de codificación que hagan uso en su estructura de un interleaver.

## Apéndice A

# Inicialización del parámetro $\mathbf{b}$

$\mathbf{b}$  es un vector fila de dimensiones  $b_{512 \times 1}$  y se muestra en la Tabla [1.1](#) a continuación:

Tabla 1.1: Inicialización del vector fila  $b$ .

1	8	8	9	10	9	7	7	7	7	7	5	5	3	3	2
1	8	8	9	10	9	7	7	7	7	7	5	5	3	3	2
2	8	8	10	10	9	7	7	7	7	7	5	5	3	3	2
2	8	8	10	10	9	7	7	7	7	7	5	5	3	3	2
4	8	8	10	10	9	7	7	7	7	7	5	5	3	3	2
4	8	8	10	10	9	7	7	7	7	7	5	5	3	3	2
4	8	8	10	10	9	7	7	7	7	7	5	5	3	3	2
4	8	8	10	10	9	7	7	7	7	7	5	5	3	3	2
6	6	8	10	10	8	6	7	7	7	7	5	5	3	3	2
6	6	8	10	10	8	6	7	7	7	7	5	5	3	3	2
6	6	8	10	10	8	8	7	7	7	7	5	5	3	3	2
6	6	8	10	10	8	8	7	7	7	7	5	5	3	3	2
6	6	8	10	10	8	8	7	7	7	6	5	5	3	3	2
6	6	8	10	10	8	8	7	7	7	6	5	5	3	3	2
6	4	8	10	9	8	8	7	7	7	6	5	5	4	2	2
6	4	8	10	9	8	8	7	7	7	6	5	5	4	2	2
7	4	8	10	9	8	8	7	7	7	6	5	4	4	2	2
7	4	8	10	9	8	8	7	7	7	6	5	4	4	2	2
7	4	8	10	9	8	8	7	7	7	6	5	4	4	2	2
7	4	8	10	9	8	8	7	7	7	6	5	4	4	2	2
7	6	8	10	9	8	8	7	7	7	6	5	4	4	2	2
7	6	8	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	8	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	8	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	8	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	9	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	9	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	9	10	9	8	8	7	7	7	6	5	4	4	2	2
8	6	9	10	9	8	8	7	7	7	6	5	4	4	2	2
8	8	9	10	9	8	8	7	7	7	5	5	4	3	2	2
8	8	9	10	9	8	8	7	7	7	5	5	4	3	2	2

## Apéndice B

# Algoritmos

### 2.1. Algoritmo de permutación del VDSL Interleaver

---

```
function vector_p=interleaverVDSL(x)
%*****
% variables
% K Longitud del vector de entrada de bits al turbo codigo
% R Numero de filas de la matriz rectangular
% C Numero de columnas de la matriz rectangular
% p Numero primo
% v raiz primitiva
% T permutacion entre filas

K=length(x);
% Paso 1 Determinar la longitud del vector de entrada
if K>=40 & K<=159
    R=5;
    T=[4, 3, 2, 1, 0]+1;
else
    R=20;
    T=[19, 9, 14, 4, 0, 2, 5, 7, 12, 18, 10, 8, 13, 17, 3, 1, 16, 6, 15,
    11]+1;
end
```

```

% Paso 2 Determinar el numero primo p para ser usado en la determinacion
del numero de columnas de la matriz rectangular, C
v_primos=[7      11      13      17      19      23      29      31
37      41      43 ...
47 53      59      61      67      71      73      79      83
89      97 ...
101 103      107      109      113      127      131      137      139
149      151 ...
157 163      167      173      179      181      191      193      197
199      211 ...
223 227      229      233      239      241      251      257];
if K>=481 & K<=530
    p=53;
    C=p;
else
    % encontrar el numero primo menor p del vector v_primos
    ii=1;
    while (v_primos(ii)+1)*R<K
        ii=ii+1;
    end
    p=v_primos(ii);
    %Determinar C
    if K<=(p-1)*R
        C=p-1;
    elseif K>(p-1)*R & K<=R*p
        C=p;
    elseif K>R*p
        C=p+1;
    end
end

% Paso 3 Se escribe la matriz rectangular de dimensiones RxC fila por
fila
if K~=R*C
    x(1,(K+1):(R*C))=0; %relleno con ceros
end
matriz_1=(reshape(x,C,R))';
% Se define una tabla para luego determinar una secuencia s, q y r
tabla_v=[
    3      2      2      3      2      5      2      3      2      6      3 ...

```

```

5      2      2      2      2      7      5      3      2      3      5 ...
2      5      2      6      3      3      2      3      2      2      6 ...
5      2      5      2      2      2      19     5      2      3      2 ...
3      2      6      3      7      7      6      3      ];

```

```
%Paso 4 Seleccionar la raiz primitiva de la v_tabla
```

```
v=tabla_v(ii);
```

```
%Paso 5 Se construye una secuencia s(j)
```

```
s(1)=1;
```

```
for j=2:p-1
```

```
    s(j)=mod(v*s(j-1),p);
```

```
end
```

```
%Paso 6 Se determina el entero primo qi
```

```
q(1)=1;
```

```
q(1,2:R)=6;
```

```
for i=2:R
```

```
    while ((gcd(q(i),p-1)==1) & (q(i)>6) & (q(i)>q(i-1)))==0
```

```
        q(i)=q(i)+1;
```

```
    end
```

```
end
```

```
%Paso 7 Se permuta la secuencia qi para crear la secuencia ri
```

```
r(T)=q;
```

```
%Paso 8 Realizar la permutacion entre las filas
```

```
for i=1:R
```

```
    if C==p
```

```
        for j=1:p-1
```

```
            U(i,j)=s(mod(j*r(i),p-1)+1);
```

```
        end
```

```
        U(i,p)=0;
```

```
    elseif C==p+1
```

```
        for j=1:p-1
```

```
            U(i,j)=s(mod(j*r(i),p-1)+1);
```

```
        end
```

```
        U(i,p)=0;
```

```
        U(i,p+1)=p;
```

```
        if K==R*C & i==R
```

```
            temp=U(R,p+1);
```

```
            U(R,p+1)=U(R,1);
```

```
            U(R,1)=temp;
```

```

        end
    elseif C==p-1
        for j=1:p-1
            U(i,j)=s(mod(j*r(i),p-1)+1)-1;
        end
    end
end

for i=1:R
    matriz_2(i,:)=matriz_1(i,U(i,:)+1);
end

for i=1:C
    matriz_3(:,i)=matriz_2(T,i);
end

k=1;
for i=1:C
    for j=1:R
        if matriz_3(j,i)~=0
            vector_p(k)=matriz_3(j,i);
            k=k+1;
        end
    end
end
end

```

---

Configuración del VDSL Interleaver

## 2.2. Algoritmo para graficar el canal típico VDSL usando un interleaver genérico y el diseño

---

```

clc;
clear;
x=[0 1 2 3 4 5 6 7 8 ]
y=[0.3272 0.2871 0.2403 0.19 0.1429 0.1034 0.07194 0.04654 0.02936]
z=[0.487 0.4732 0.3946 0.2304 0.08266 0.02692 0.006525 0.0009309
    0.0002539]

```

```
semilogy(x,y,'*b-', 'linewidth',2.0);    %Simulacion del canal usando un
      permutador generico
hold on;

semilogy(x,z,'+r—', 'linewidth',1.0);    %Simulacion del canal usando el
      permutador dise~nado
hold on;

title('Comparacion de rendimiento'); xlabel('Eb/No(dB)'); ylabel('BER');
legend('Canal tipico', 'VDSL Interleaver');
axis tight
grid
```

---

Algoritmo para graficar las curvas de BER

### 2.3. Algoritmo para graficar el rendimiento en términos de BER implementando diferentes permutadores

---

```

clc;
clear;
x=[0 1 2 3 4 5 6 7 8]
t=[0.4765 0.4519 0.3784 0.2864 0.1548 0.03906 0.007406 0.0008976
    0.000236]
v=[0.444 0.4323 0.40379 0.3082 0.1575 0.04284 0.007849 0.001191 0.00043]
w=[0.4289 0.3727 0.2893 0.2005 0.123 0.07548 0.04959 0.02791 0.01988]
y=[0.4609 0.4536 0.4438 0.4332 0.4239 0.4159 0.4096 0.4045 0.39]
z=[0.487 0.4732 0.3946 0.2304 0.08266 0.02692 0.006525 0.0009309
    0.00012539]

semilogy(x,t,'c-', 'linewidth',2.0);           %permutador aleatorio
hold on;

semilogy(x,v,'m-', 'linewidth',2.0);           %permutador matricial
hold on;

semilogy(x,w,'k-', 'linewidth',2.0);           %permutador helicoidal
hold on;

semilogy(x,y,'b-', 'linewidth',2.0);           %permutador convolucional
hold on;

semilogy(x,z,'r—', 'linewidth',1.0)           %permutador diseñado
hold on ;

title('Comparacion de rendimiento entre permutadores existentes y el
    diseñado'); xlabel('Eb/No(dB)'); ylabel('BER');
legend('Aleatorio', 'Matricial', 'Helicoidal', 'Convolucional', 'Diseñado
    ');
axis tight
grid

```

---

Algoritmo para graficar las curvas de BER

## Referencias Bibliográficas

- [1] María Aracely Martínez Amaya. Análisis del decodificador de turbo códigos con el empleo de ventanas deslizantes y algoritmo max-log-map. Master's thesis, Instituto Politécnico Nacional Centro de Investigación y Desarrollo de Tecnología Digital., Noviembre, 2011.
- [2] Juan Valera Requena. Diseño y desarrollo del interleaver para el decodificador turbo código wimax. Master's thesis, Escuela Técnica Superior de Ingeniería de Telecomunicación. Universidad Politécnica de Cartagena, Julio 2007.
- [3] Recomendación UIT-T G.993.1. *Transceptores de línea de abonado digital de velocidad muy alta*. 2004.
- [4] J.A.C. Bingham. *ADSL, VDSL, and multicarrier modulation*. Wiley series in telecommunications and signal processing. Wiley, 2000. ISBN 9780471290995. URL <https://books.google.es/books?id=kPtSAAAAMAAJ>.
- [5] ANSI T1.413. 1998. Revision of ANSI T1.413. 1995. Standards project for interfaces relating to carrier to customer connection of asymmetrical digital subscriber line (adsl) equipment. *Network and Customer Installation Interfaces-Asymmetric Digital Subscriber Line (ADSL) Metallic Interface.*, pages-, 2004.
- [6] D.J. Rauschmayer. *ADSL/VDSL Principles: A Practical and Precise Study of Asymmetric Digital Subscriber Lines and Very High Speed Digital Subscriber Lines*. Macmillan technology series. Macmillan Technical Publishing, 1999. ISBN 9781578700158. URL <https://books.google.es/books?id=rCgfAQAAIAAJ>.

- [7] L.W. Couch. *Sistemas de comunicación digitales y analógicos 7ED*. Pearson Educación, 2008. ISBN 9789702612162. URL <https://books.google.es/books?id=A6im0gAACAAJ>.
- [8] S.S. Haykin and M. Moher. *Communication Systems*. Wiley, 2010. ISBN 9780470169964. URL <https://books.google.es/books?id=nQFVPgAACAAJ>.
- [9] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, volume 2, pages 1064–1070 vol.2, May 1993. doi: 10.1109/ICC.1993.397441.
- [10] M.Z. Wang, A. Sheikh, and F. Qi. Interleaver design for short turbo codes. In *Global Telecommunications Conference, 1999. GLOBECOM '99*, volume 1B, pages 894–898 vol. 1b, 1999. doi: 10.1109/GLOCOM.1999.830210.
- [11] S. Benedetto, D. Divsalar, and J. Hagenauer. *Concatenated Coding Techniques and Iterative Decoding: Sailing Toward Channel Capacity*. IEEE journal on selected areas in communications. 1998. URL <https://books.google.es/books?id=ntNSQwAACAAJ>.
- [12] Diego Fernando Vallejo Huanga. Estudio y simulación de turbo códigos utilizando el algoritmo map y sova. Master's thesis, Facultad de Ingeniería Eléctrica y Electrónica, Septiembre 2011.
- [13] *Digital Communications: Fundamentals & Applications, 2/E*. Pearson Education, 2009. ISBN 9788131720929. URL <https://books.google.es/books?id=TuvKdwNMS48C>.
- [14] J.G. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill International Edition. McGraw-Hill, 2008. ISBN 9780071263788. URL <https://books.google.es/books?id=ksh0GgAACAAJ>.
- [15] S. Lin and Jr D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice Hall, 1983.
- [16] T. K. Moon. *Error correction coding, mathematical methods and algorithms*. 1st ed. Wiley Interscience & Sons, 2005.

- [17] C. Berrou and A. Glavieux. Near optimum error correcting coding and decoding: turbo-codes. *Communications, IEEE Transactions on*, 44(10):1261–1271, Oct 1996. ISSN 0090-6778. doi: 10.1109/26.539767.
- [18] J. C. M. y Patrick Guy Farrell. *Essentials of error-control coding*. 1st ed. John Wiley & Sons, 2006.
- [19] C. Heegard and S.B. Wicker. *Turbo Coding*. The Springer International Series in Engineering and Computer Science. Springer, 1999. ISBN 9780792383789. URL [https://books.google.es/books?id=aTWkPD\\_C0soC](https://books.google.es/books?id=aTWkPD_C0soC).
- [20] S. Le Goff, A. Glavieux, and C. Berrou. Turbo-codes and high spectral efficiency modulation. In *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.'* IEEE International Conference on, pages 645–649 vol.2, May 1994. doi: 10.1109/ICC.1994.368804.
- [21] K.S. Jacobsen. Synchronized dmt (sdmt) for very high-speed digital subscriber line (vdsl) transmission. In *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE*, volume 2, pages 856–861 vol.2, 1998.
- [22] Ing. Alfonso Martínez Cruz. Implementación de códigos correctores de errores para un módem para la comunicación sobre la red eléctrica, Diciembre, 2011.
- [23] L.L. Hanzo, C. Somerville, and J. Woodard. *Voice and Audio Compression for Wireless Communications*. IEEE Series on Digital & Mobile Communication. Wiley, 2008. ISBN 9780470516027. URL <https://books.google.es/books?id=9u74lCc1ltAC>.
- [24] Recomendación UIT-T G.992.1. *Transceptores de línea de abonado digital asimétrica*. 06,1999.
- [25] Recomendación UIT-T G.993.2. *Transceptores de línea de abonado digital de velocidad muy alta 2*. 02, 2006.

- [26] J.S. Moreno-Gil. *Manual de estadística universitaria: inductiva*. Colección Universidad. ESIC, 1995. ISBN 9788473561150. URL <https://books.google.es/books?id=3IXxqc2uNxEC>.
- [27] D.E. Knuth. *Clasificación y búsqueda*. El arte de programar ordenadores. Reverté, 1986. ISBN 9788429126648. URL <https://books.google.es/books?id=KGkUjFxBYB8C>.