



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

EVALUACIÓN Y APLICACIÓN DE ESTRATEGIAS PARA CONTROL DE ERRORES  
EN CANALES SATELITALES MEDIANTE CODIFICACIÓN ALGEBRAICA

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL ELECTRICISTA

FELIPE IVÁN TRONCOSO HERNÁNDEZ

PROFESOR GUÍA:  
MARCOS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:  
PATRICIO PARADA SALGADO  
ALEX BECERRA SAAVEDRA

SANTIAGO DE CHILE  
AGOSTO 2012

RESUMEN DE LA MEMORIA  
PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL ELECTRICISTA  
POR: FELIPE IVÁN TRONCOSO HERNÁNDEZ  
FECHA: AGOSTO 2012  
PROF. GUÍA: SR. MARCOS DÍAZ QUEZADA

## EVALUACIÓN Y APLICACIÓN DE ESTRATEGIAS PARA CONTROL DE ERRORES EN CANALES SATELITALES MEDIANTE CODIFICACIÓN ALGEBRAICA

En el contexto de las Ciencias Aeroespaciales, surge la necesidad de disponer al alcance de universidades y otros organismos la posibilidad de realizar investigación en el área sin requerir inversiones costosas y planificaciones de largo plazo. Esta idea se concreta actualmente en el lanzamiento de pequeños satélites agrupados bajo el estándar *CubeSat*. El proyecto SUCHAI, del Departamento de Ingeniería Eléctrica, busca poner en órbita el primer satélite de este tipo desarrollado por una Universidad chilena.

Las severas limitaciones de peso, tamaño y recursos de estos satélites impone varios desafíos para mantenerlos funcionales en su órbita. La comunicación con la estación terrestre, desde donde se envían y reciben señales, es crítica para dicho fin. Varios elementos afectan el enlace, como son la alta velocidad de traslación del satélite y el ruido introducido por fenómenos climáticos y la ionosfera. Esto lleva a la necesidad de evaluar mecanismos para minimizar la corrupción en la información transmitida, entre los cuales son de amplia aplicación los esquemas de codificación.

En el contexto del proyecto SUCHAI, este trabajo tiene por objetivos evaluar estrategias de control de errores adecuadas para el entorno descrito, acotando el estudio a los esquemas de codificación algebraica, e implementar aquella que se ajuste mejor a los requisitos impuestos.

La evaluación realizada abarca la simulación de algunas estrategias de codificación, bajo esquemas de modulación en frecuencia y en fase binarios cuyos resultados indican que el código convolucional reduce significativamente los errores en la transmisión. Esto motiva la implementación de un esquema convolucional en un módulo FPGA Spartan-3E de Xilinx, la que se realizó parcialmente en el software ISE.

Con los logros alcanzados, se presentan desafíos de implementación completa y eficiente del código convolucional, realización de pruebas bajo condiciones más realistas a las simuladas, o la evaluación de otros esquemas de codificación y modulación bajo modelos de canal más representativos.

# Agradecimientos

En primer lugar, quiero expresar mi agradecimiento a quienes conforman mi Comisión de este trabajo, los profesores Marcos Díaz, Patricio Parada y el ingeniero Alex Becerra, en primer lugar por su actitud de apoyo, disposición, consejo y crítica constantes a lo largo de este proyecto, y en segundo lugar por su incentivo al trabajo bien realizado, cualidad que he intentado aplicar en el desarrollo que expongo, y que me propongo preservar.

En este espíritu debo el reconocimiento al ambiente creado en el laboratorio del proyecto SUCHAI, el que ha influido mucho en mi formación como ingeniero con la capacidad de resolver situaciones prácticas, algo poco enfatizado por la formación teórica de la Escuela. A mis colegas del equipo de Comunicaciones, Sebastián D. y Camilo R., debo desde mi bagaje actual en el software libre hasta las muchas conversaciones sobre cualquier tema, varias veces acompañadas de una cerveza. También a los colegas de programación Tomás O. y Carlos G. con quienes hemos compartido más de una tarde "terraceando". Las experiencias vividas gracias a este proyecto las debo a haberme atrasado en la carrera, por lo que estoy agradecido de dicha circunstancia.

De mi paso por Beaucheff, muchas personas y situaciones influyeron en lo que soy ahora. Doy gracias por todos a quienes conocí en la Pastoral de Ingeniería, en la cual participé los dos primeros años de mi carrera: Hernán, Miguel Ángel, Fernando, Claudia, María Teresa, Felipe, Heinz, y muchos otros. A quienes conocí de la Pastoral de la UDP en el memorable viaje a Chiloé, personas a quienes guardo un muy especial aprecio: Belén, Iván, Álvaro, Coté. A mis amigos del colegio que seguimos carrera en la FCFM: Juan, Miguel, Sergio, ha sido increíble lo que hemos cambiado desde el ya lejano 2001 en primero medio, ojalá podamos seguir apreciando nuestro crecimiento en todo ámbito. A mis amigos, integrantes y ex-integrantes de banda: Pablo, Charles, Nano, Marcel, Pailo, sin ustedes no podría haber florecido mi pasión inquieta por la música, y tengo la certeza de que nos hemos enriquecido muchísimo mutuamente.

Finalmente, pero no menos importante, agradezco a mis padres por el apoyo silencioso e incondicional a lo largo de mis años de estudio, y a P., cuyo soporte emocional, amor y amistad no soy capaz de dimensionar.

# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Motivación . . . . .	7
1.2. Objetivo General y Alcances . . . . .	8
1.3. Objetivos Específicos . . . . .	8
1.4. Estructura de la memoria . . . . .	9
<b>2. Aspectos Teóricos</b>	<b>10</b>
2.1. Elementos de canales de comunicaciones . . . . .	10
2.1.1. Sistema de comunicaciones . . . . .	10
2.1.2. Canales discretos . . . . .	10
2.1.3. Canales sin memoria . . . . .	11
2.1.4. Modelos de canal . . . . .	12
2.1.5. Capacidad de canal . . . . .	14
2.2. Modulación digital . . . . .	14
2.2.1. Modulación en banda-base . . . . .	14
2.2.2. Modulación en pasa-banda . . . . .	16
2.3. Indicadores de desempeño . . . . .	19
2.3.1. Energía-bit por densidad espectral . . . . .	19
2.3.2. Probabilidad de error de símbolos . . . . .	20
2.3.3. Tasa de errores de símbolos . . . . .	21
2.3.4. Ganancia de código . . . . .	22
2.4. Comunicaciones satelitales . . . . .	22

2.4.1.	Descripción general . . . . .	22
2.4.2.	Descripción de enlaces de baja órbita . . . . .	24
2.5.	Codificación para control de errores . . . . .	25
2.5.1.	Esquemas de control de errores . . . . .	25
2.5.2.	Códigos lineales de bloques . . . . .	26
2.5.3.	Esquemas de codificación más conocidos . . . . .	31
2.6.	Diseño de hardware digital . . . . .	34
2.6.1.	Field Programmable Gates Array . . . . .	34
2.7.	Estado del arte . . . . .	36
2.7.1.	Codificación en canales satelitales . . . . .	36
2.7.2.	Esquemas de codificación en FPGAs . . . . .	37
<b>3.</b>	<b>Simulaciones computacionales</b>	<b>38</b>
3.1.	Simulaciones canal binario simétrico . . . . .	38
3.1.1.	Planteamientos previos . . . . .	38
3.1.2.	Simulación sin codificación . . . . .	40
3.1.3.	Discusión de caso sin codificación . . . . .	41
3.1.4.	Simulación codificación BCH . . . . .	41
3.1.5.	Discusión codificación BCH . . . . .	42
3.1.6.	Simulación codificación Reed-Solomon . . . . .	44
3.1.7.	Discusión codificación Reed-Solomon . . . . .	45
3.1.8.	Simulación codificación convolucional . . . . .	47
3.1.9.	Discusión codificación convolucional . . . . .	48
3.1.10.	Comparación entre esquemas . . . . .	49
3.1.11.	Discusión comparativa entre codificaciones . . . . .	50
3.2.	Conclusión . . . . .	54
<b>4.</b>	<b>Implementación en FPGA</b>	<b>56</b>
4.1.	Especificaciones de diseño . . . . .	56

4.1.1. Codificador convolucional . . . . .	56
4.1.2. Decodificador convolucional . . . . .	57
4.2. Implementación . . . . .	59
4.2.1. Codificador convolucional . . . . .	59
4.2.2. Decodificador convolucional . . . . .	60
4.2.3. Revisión . . . . .	62
<b>5. Conclusiones</b>	<b>63</b>
5.1. Conclusiones generales . . . . .	63
5.2. Trabajos Futuros . . . . .	65
<b>Referencias</b>	<b>69</b>
<b>A. Distribuciones de tasas de error en codificación</b>	<b>I</b>
A.1. Distribución de BER código BCH . . . . .	II
A.2. Distribución de BER código Reed-Solomon . . . . .	III
A.3. Distribución de BER código convolucional . . . . .	IV
A.4. Aspectos de distribuciones observadas . . . . .	V

# Índice de figuras

2.1. Canal de comunicaciones . . . . .	11
2.2. Canal binario simétrico . . . . .	12
2.3. Canal AWGN con entrada discretizada. . . . .	13
2.4. Representación discretizada del canal AWGN. . . . .	13
2.5. Modulaciones en banda-base NRZ de la secuencia <b>10111001</b> . . . . .	15
2.6. Modulaciones en banda-base RZ de la secuencia <b>10111001</b> . . . . .	16
2.7. Modulaciones básicas pasa-banda. . . . .	17
2.8. Demodulador de símbolos binarios. . . . .	20
2.9. Codificación de bloques. . . . .	27
2.10. Detección de síndrome en el decodificador. . . . .	30
2.11. Bloques funcionales de FPGA Xilinx. . . . .	35
2.12. Tarjeta Spartan 3E. . . . .	35
3.1. BER sin codificación, casos teórico y simulación en canal BSC, para modulación en BPSK, BFSK demodulado en forma coherente y no coherente. . . . .	40
3.2. Desempeño codificador BCH(63,57) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK. Nota: el gráfico (a) se ha desplazado levemente a la derecha para apreciar el único valor no nulo en la cota de 10 % (BER=0.0526 para $E_b/N_0 = 0$ [dB]). . . . .	42
3.3. Desempeño codificador RS(255,239) en canal binario simétrico.(a) Modulación BPSK, (b) Modulación BFSK. . . . .	45
3.4. Diagrama del codificador convolucional (171,133). Figura extraída de [18]. . . . .	47
3.5. Desempeño codificador convolucional (171,133) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK. . . . .	48

3.6.	Desempeño de códigos BCH(63,57), RS(255,239) y Convolutacional(171,133) en canal BSC con modulación BPSK. . . . .	50
3.7.	Desempeño de códigos BCH(63,57), RS(255,239) y Convolutacional(171,133) en canal BSC, modulación BFSK con detección no coherente. . . . .	50
4.1.	Enmallado de decodificación para un codificador con $K = 2$ unidades de memoria. . . . .	57
4.2.	Unidades principales del decodificador con algoritmo de Viterbi. . . . .	58
4.3.	Esquemático RTL de codificador convolutacional (2,1,7) realizado en ISE. . . .	59
4.4.	Utilización de recursos en FPGA para decodificador Viterbi, generado en ISE.	60
4.5.	Esquemático RTL del decodificador con algoritmo de Viterbi, generado en ISE.	61
4.6.	Utilización de recursos en FPGA para decodificador Viterbi, generado en ISE.	62
A.1.	Distribución de BER en código BCH(63,57) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK. . . . .	II
A.2.	Distribución de BER en código RS(255,239) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK. . . . .	III
A.3.	Distribución de BER en código convolutacional(171,133) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK. . . . .	IV



# Índice de tablas

2.1.	Operaciones en $GF(2)$ . . . . .	28
2.2.	Capacidad de corrección $t$ de códigos BCH( $n, k$ ) en función de $R = k/n$ con $n = 63$ . Tabla extraída de [42] . . . . .	32
3.1.	Ganancias de modulación BPSK respecto a BFSK no coherente, para tasas de error de $10^{-3}$ , $10^{-4}$ y $10^{-5}$ . . . . .	41
3.2.	Comparación de potencias requeridas para esquemas de codificación con modulación BPSK, en [dB]. . . . .	51
3.3.	Comparación de potencias requeridas para esquemas de codificación con modulación BFSK, en [dB]. . . . .	51
3.4.	Ganancias de código con modulación BPSK, en [dB]. . . . .	52
3.5.	Ganancias de código con modulación BFSK, en [dB]. . . . .	52
3.6.	Tasas de código para esquemas de codificación. . . . .	54

# Capítulo 1

## Introducción

### 1.1. Motivación

Desde principios de la década actual se ha extendido ampliamente la creación de satélites de baja órbita, a nivel de investigación y aplicación académica, que permiten disponer al alcance de estudiantes de variados niveles y académicos nuevos desafíos tecnológicos, y al mismo tiempo fomentar la investigación de una amplia diversidad de fenómenos de manera innovadora. Este desarrollo se inició en los laboratorios de la Universidad Politécnica de California (*CalPoly*) y de la Universidad de Stanford en 1999; hoy se ha ampliado a un número creciente de Universidades y organismos tanto civiles como gubernamentales.

Los pequeños satélites lo conforman aquellos de tamaño y peso reducidos, usualmente bajo los 500 [kg], con combustible incluido [55]. Este grupo se divide en varios subgrupos de acuerdo a su tamaño, peso, propósito u otros criterios, y presenta ventajas en relación a costos, rapidez de desarrollo, objetivos acotados y multiplicidad de aplicaciones. Dentro de esta categoría, destaca el estándar *CubeSat*, originado en las universidades previamente mencionadas, cuyas características son su forma cúbica, de 10 [cm] de longitud, y un peso máximo de 1.3 [kg] constituido por la estructura y componentes del satélite. Este estándar es el empleado con frecuencia por las entidades mencionadas en el párrafo anterior.

En el año 2010, el Departamento de Ingeniería Eléctrica (DIE) de la Universidad de Chile se unió a este desafío y actualmente el profesor Marcos Díaz (DIE), en conjunto con otros dos académicos del DIMEC y del DFI, dirigen este proyecto cuyo lanzamiento se espera para fines de 2012. Este proyecto se ha denominado SUCHAI (*Satellite of University of Chile for Aerospace Investigation*) y además del lanzamiento mismo del satélite, busca ser el comienzo de una nueva área de investigación aeroespacial dentro del Departamento de Ingeniería Eléctrica.

Dentro del funcionamiento de un satélite es crucial asegurar que los datos enviados o recibidos por la unidad de comunicaciones sean confiables, es decir que los errores producidos en el enlace sean reducidos a una cantidad mínima. En comunicaciones digitales, una de las maneras de resolver esta situación es utilizando codificación de canal, la que mediante

diversos enfoques puede reducir la tasa de errores en la transmisión. El estudio y aplicación de un mecanismo de codificación es el enfoque del presente trabajo.

## 1.2. Objetivo General y Alcances

En el marco del proyecto satelital de SUCHAI, y de la proyección de construir un dispositivo de comunicaciones en el Departamento de Ingeniería Eléctrica, el objetivo general de este trabajo consiste en la investigación e implementación de un prototipo de codificador y decodificador que permita detectar y corregir los errores producidos en un canal satelital de baja órbita, situado entre 200 y 800 kilómetros de altitud. El prototipo es realizado utilizando módulos de desarrollo electrónicos basados en FPGAs disponibles en el Departamento de Ingeniería Eléctrica (Electrotecnologías). Tras una primera etapa de simulaciones computacionales, que permiten seleccionar el esquema de codificación a aplicar, éste se implementa en FPGA. Finalizado este trabajo, quedarán a disposición del laboratorio del proyecto SUCHAI los códigos computacionales de simulación e implementación, de manera que si eventualmente se construye el dispositivo de comunicaciones, pueda ser añadido un esquema de codificación y decodificación, y del mismo modo en la estación terrena.

Cabe señalar que, establecido el objetivo general, queda excluido el diseño de una placa o bloque para ser incluido en el lanzamiento, pues dicha labor además debería considerar la inclusión de etapas de modulación, amplificación, intercambio de datos con el procesador satelital, conexión a la antena y adecuar dicha placa a las restricciones del satélite.

## 1.3. Objetivos Específicos

Los objetivos específicos permiten enlistar las labores fundamentales a cumplir para concretar el objetivo principal de este trabajo, a saber:

- Revisión bibliográfica del estado del arte y análisis de los principales códigos. Para esto se realiza un estudio teórico, plasmado en el Capítulo 2, donde se resumen las características de los códigos más usuales, además de conceptos adicionales necesarios para su comprensión.
- Simulaciones computacionales de canales con modulación pasabanda, codificados y no-codificados, para comparar sus comportamientos y seleccionar el esquema de codificación con mejor desempeño.
- Decisión del esquema de codificación y decodificación a implementar. Esto requiere conocer las ventajas y limitaciones de los algoritmos tanto para codificar como para decodificar, pues puede existir un *trade-off* entre las mejoras propias del código y su complejidad de implementación.

- Implementación del prototipo. Esto requiere contar tanto con el hardware adecuado (tarjeta de desarrollo en FPGA, computador adecuado) y el software necesario (suite de Xilinx ®ISE). Bajo este objetivo se implementan el codificador y decodificador vía lenguaje descriptivo de hardware (HDL) o esquemáticos digitales, para ser embebidos en la arquitectura de la FPGA.

## 1.4. Estructura de la memoria

La estructura utilizada en este documento para exponer el trabajo realizado es la siguiente:

- **Capítulo 1. Introducción:** Corresponde a la descripción del tema, la motivación de éste y los alcances y objetivos del trabajo realizado.
- **Capítulo 2. Aspectos Teóricos:** Corresponde a la revisión bibliográfica o antecedentes. En este capítulo se explican los conceptos necesarios para la comprensión y contextualización del trabajo.
- **Capítulo 3. Simulaciones computacionales:** Se analizan mediante simulaciones los diversos esquemas de codificación; de este modo se decide con criterios cuantitativos el esquema de codificación a implementar.
- **Capítulo 4. Implementación en FPGA:** Se realiza la implementación en hardware del esquema utilizado, tanto para el codificador como el decodificador.
- **Capítulo 5. Conclusiones:** Se concluye con los resultados obtenidos, discutiendo la cobertura de los objetivos y el alcance del trabajo. Se exponen además los posibles aportes futuros, tales como la implementación del hardware, sea ésta para uso académico como prototipo ó para ser incluido en vehículos aeroespaciales.

# Capítulo 2

## Aspectos Teóricos

### 2.1. Elementos de canales de comunicaciones

#### 2.1.1. Sistema de comunicaciones

Un sistema de comunicaciones se constituye de tres elementos principales: el transmisor, desde donde se envía la información deseada; el receptor, donde es recibida, y el canal, medio físico entre el transmisor y el receptor. Es posible transmitir información mediante una señal que varía de forma continua, constituyendo una señal analógica. De otro modo, los datos pueden ser transformados en un conjunto de valores discretos, en lo que se denomina una señal digital. En cualquier caso, la información transmitida sufre degradación por diversas fuentes físicas de ruido en el canal y en el receptor.

En un sistema de comunicaciones cuyas señales son digitales (Figura 2.1), el transmisor y el receptor cuentan con módulos que procesan la señal para ser enviada a través de un medio físico. Un **codificador** detecta y/o corrige los errores producidos en el canal modificando la trama de datos, por lo general añadiendo bits de redundancia. En la etapa posterior, un **modulador** transforma el mensaje digital en bits a señales analógicas mediante algún esquema de modulación determinado. De manera similar, en el receptor se tendrá un **demodulador** que transforma la señal recibida en bits, y un **decodificador** que utiliza la redundancia o modificación introducida para detectar y/o corregir los errores que el canal introdujo.

#### 2.1.2. Canales discretos

Un *canal discreto* corresponde a la abstracción de un canal de transmisión físico, incluyendo las etapas de modulación y demodulación; por ello está caracterizado por entradas y salidas que toman una cantidad finita de valores. En casos prácticos como un sistema

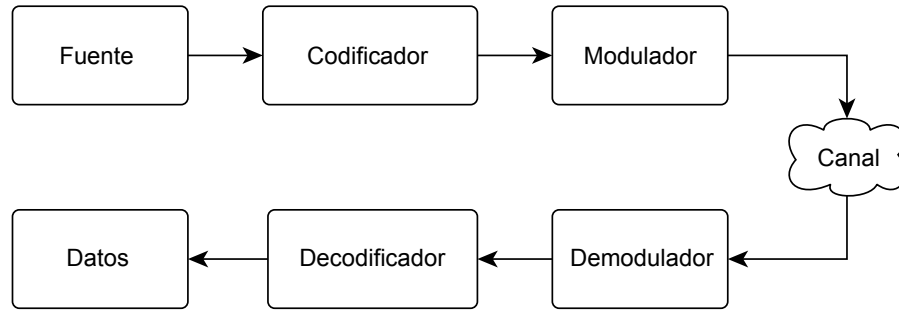


Figura 2.1: Canal de comunicaciones

satelital, el canal produce formas de onda continuas en el tiempo en su salida<sup>1</sup>, lo que lo constituye como *canal analógico*. Esta forma de onda puede tener cualquier amplitud real, limitada por la potencia máxima del transmisor o del ruido introducido por el canal, y su espectro en frecuencia es limitado por el ancho de banda del equipo receptor. Por medio de las hipótesis descritas en la Definición 2.1.2, el canal mencionado puede ser tratado como discreto.

En el caso de un canal discreto finito, la cantidad de valores que toman las entradas y salidas son finitas. El conjunto acotado de valores posibles en la entrada se denomina **alfabeto de entrada**. Análogamente, en la salida tomará sus valores en un **alfabeto de salida**. Los elementos de cada alfabeto se denominan **símbolos**. Esto permite definir un canal discreto como la correspondencia entre el alfabeto de entrada y el de salida: si  $x$  es un símbolo de entrada, e  $y$  el símbolo de salida, el canal asigna el valor  $y$  con probabilidad  $p(y|x)$  al transmitir  $x$  [45, Cap. 9].

### 2.1.3. Canales sin memoria

En términos generales, la salida de un canal discreto  $y_i$  depende no sólo de la entrada  $x_i$  correspondiente a ese instante, sino que de las entradas anteriores (en el caso de los canales con interferencia inter-simbólica), e incluso de entradas futuras (como en medios de almacenamiento [45, Cap. 9]). En tales casos la dependencia entre los símbolos se denomina *memoria*. Un canal discreto cuya salida  $y_i$  depende sólo de la entrada respectiva  $x_i$  se dirá *sin memoria* (DMC). Para este canal se cumple que, para una secuencia de símbolos de entrada  $\mathbf{x}$  y salida  $\mathbf{y}$ , ambas de largo  $n$ , la probabilidad conjunta entre  $\mathbf{x}$  e  $\mathbf{y}$  está dada por:

$$(2.1) \quad p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n p(y_i|x_i).$$

---

<sup>1</sup>O en la entrada del demodulador.

### 2.1.4. Modelos de canal

La naturaleza del medio físico puede ser modelada matemáticamente, lo cual da origen a varias familias de canal que se emplean en el área de comunicaciones. Los modelos principales que guardan relación con este trabajo son los siguientes:

**Definición 2.1.1** *Un canal binario simétrico (BSC, Figura 2.2) es aquel canal DMC donde el alfabeto de entrada  $\mathbf{I}$  y el alfabeto de salida  $\mathbf{O}$  constan de dos símbolos llamados bits, denotados por '0' y '1', donde la probabilidad de que cualquiera de dichos símbolos sea transmitido correctamente es  $1-p$ , y complementariamente, la probabilidad de que un símbolo '0' cambie a '1' o viceversa es  $p$  [11, Cap. 1].*

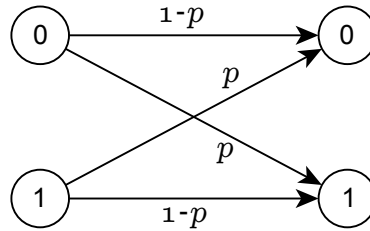


Figura 2.2: Canal binario simétrico

**Definición 2.1.2** *Un canal de ruido blanco gaussiano aditivo (AWGN) continuo es un modelo de canal aleatorio cuya salida es un proceso real aleatorio. En términos matemáticos la salida  $Y(t)$  se define como:*

$$(2.2) \quad Y(t) = X(t) + N(t)$$

donde  $X(t)$  es la forma de onda en la entrada vista como proceso aleatorio real, y  $N(t)$  es un proceso real de ruido gaussiano blanco con densidad de potencia de ruido  $N_0$ , independiente de  $X(t)$ .

La señal  $X(t)$  tiene potencia con media  $P$ ; además es acotada en el canal de banda  $B$ , siendo éste el intervalo de frecuencias no-negativas con **ancho de banda**  $W$ , medido en [Hz]. Dicho canal de banda es **banda-base** si  $B = [0, W]$ , **pasa-banda** si  $B = [W_0, W]$ ,  $W > W_0 > 0$ , y **pasa-alto** si  $B = [W, \infty[$ . Los parámetros que permiten caracterizar un canal AWGN continuo son su ancho de banda  $W$  y su **razón señal-a-ruido**  $SNR$ . Este último parámetro está dado por:

$$(2.3) \quad SNR = \frac{P}{N_0 W}$$

donde  $N_0 W$  es la potencia de ruido total en la banda  $B$ , y  $P$  la potencia de la señal modulada.

Dado un ancho de banda  $W$ , el teorema del muestreo establece que la señal continua se debe muestrear a una tasa de  $2W$  para ser reconstruida sin pérdida de información; además,

el teorema de irrelevancia permite asegurar la detección de  $X(t)$  a partir del muestreo de  $Y(t)$  desde un canal con ruido AWGN [20]. Dadas las hipótesis del modelo continuo, los procesos gaussianos  $X(t)$  y  $N(t)$  pueden ser reformulados como secuencias discretas  $\mathbf{X}$  y  $\mathbf{N}$  respectivamente, y por tanto el modelo mismo puede ser considerado como de tiempo discreto [20, Cap. 2]. Al discretizar el canal AWGN, es posible representar la entrada como un alfabeto discreto  $\mathcal{X} \in \{a_1, a_2, \dots, a_Q\}$  cuya salida  $\mathcal{Y} \in (-\infty, \infty)$  es una probabilidad dada por [57, Cap. 1]:

$$(2.4) \quad p(y|a_k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-a_k)^2/2\sigma^2} \quad \text{para todo } y \in \mathbf{Y}$$

donde  $k = 1, 2, \dots, Q$ . La Figura 2.3 ilustra este modelo.

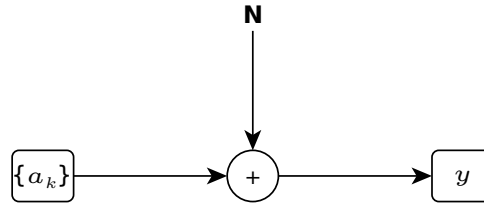


Figura 2.3: Canal AWGN con entrada discretizada.

La discretización de las secuencias de entrada y de ruido AWGN no implica que las salidas tomen valores discretos, como se observa en la Ecuación 2.4. Sin embargo, como se verá posteriormente, los esquemas de codificación se implementan en la práctica mediante circuitos digitales, por lo que es conveniente buscar una representación discretizada completa del canal AWGN, incluyendo sus salidas. Mediante la *cuantización* implementada en el dispositivo demodulador, la salida continua de este último es transformada a secuencias digitales. Dependiendo de la cantidad de secuencias de salida posibles (estados), la relación entre las secuencias discretas entrantes al canal y las de salida del demodulador definen el modelo de canal que equivale a una representación digital del canal AWGN [57, Cap. 2]. La Figura 2.4 esquematiza el canal equivalente, discretizado del modelo AWGN. Esta reformulación permite considerar el canal AWGN, el modulador y el demodulador como un único ente, cuya entrada proviene del *codificador de canal* y cuya salida accede al *decodificador de canal*. En este trabajo se considera que la relación entre las entradas y salidas conforman un canal DMC.

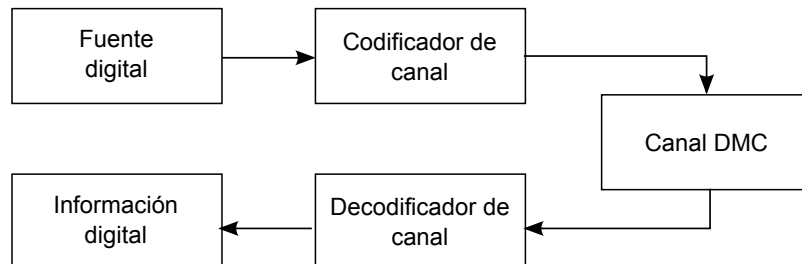


Figura 2.4: Representación discretizada del canal AWGN.



### 2.1.5. Capacidad de canal

Todo canal de transmisión introduce errores a la señal que se envía a través de él. Las características particulares de dicho canal lo hacen más o menos confiable para transmitir información. En una comunicación confiable, es posible reducir o limitar la cantidad de errores generados.

**Definición 2.1.3** La *capacidad del canal*  $C$  corresponde al límite de símbolos por unidad de tiempo que pueden ser transmitidos de forma confiable a través de dicho canal.

El concepto de capacidad de canal acota la tasa de transmisión que posibilita una comunicación confiable. En 1948, C. Shannon establece teóricamente la manera de mejorar la confiabilidad de un canal, relacionándola a la tasa de transmisión  $R$  empleada en un sistema de comunicaciones:

**Teorema 2.1.1** Si la tasa de transmisión  $R$  es menor a la capacidad de canal  $C$ , entonces es posible diseñar un esquema de codificación de canal, cuya probabilidad de error en los símbolos de salida sea tan pequeña como se desee [51].

De esta manera, si la tasa de transmisión  $R$  supera la capacidad del canal, no será posible limitar mediante codificación la tasa de errores.

En un canal binario simétrico, dada la probabilidad  $p$  de error, la capacidad es [44, Cap. 7]:

$$(2.5) \quad C_{BSC} = 1 + p \log_2(p) + (1 - p) \log_2(1 - p) \quad [\text{bps}].$$

En un canal continuo AWGN, la capacidad está dada por [45, Cap. 9]:

$$(2.6) \quad C = W \cdot \log_2(1 + SNR) \quad [\text{bps}].$$

## 2.2. Modulación digital

La modulación digital es un proceso que transforma una secuencia de símbolos digitales en una señal apta para ser transmitida en un medio físico analógico. Para distancias cortas, se usa modulación en banda-base, llamada a veces codificación de línea (*line coding*) [64, Cap. 2]. Para distancias largas, y transmisión inalámbrica en particular, se usa la modulación pasabanda, también conocida como modulación por portadora (*carrier modulation*) [64, Cap. 2]. Ambos esquemas se exponen a continuación.

### 2.2.1. Modulación en banda-base

Se entiende la modulación en banda-base como la representación de secuencias digitales mediante formas de onda adecuadas para la transmisión en canal de banda-base  $B = [0, W]$

para alguna frecuencia  $W > 0$ , sin realizar transformación en frecuencia. Existe una gran variedad de esquemas de modulación en banda-base, desarrollados para responder a propiedades deseadas tales como eficiencia en potencia y ancho de banda, o sincronización adecuada [64, Cap. 2].

En la Figura 2.5 se agrupan tres esquemas correspondientes a modulación NRZ (*Non-Return to Zero*). En el esquema NRZ-L, se emplean dos niveles de amplitud  $A$  y  $-A$  para identificar los dígitos binarios '1' y '0', respectivamente. En el esquema NRZ-M, un cambio de nivel de  $A$  a  $-A$  o viceversa es usado para señalar un '1', mientras que el '0' se representa sin realizar transición. El esquema es similar en NRZ-S, pero el cambio de nivel señala el bit '0'. Estas dos últimas formas de modulación son parte de las llamadas *codificaciones diferenciales*. En cuanto a aplicaciones, NRZ-L es de uso extensivo en lógica digital y NRZ-M se emplea principalmente en grabación en cinta magnética. La limitación principal de estos esquemas es la dificultad de recuperar los ciclos de reloj (*clock recovery*) cuando se transmiten secuencias largas de un mismo estado '0' ó '1'. Debido a esto, el uso de esquemas NRZ se restringe a redes de corto alcance, como es el caso de las redes de área local [64, Cap. 2].

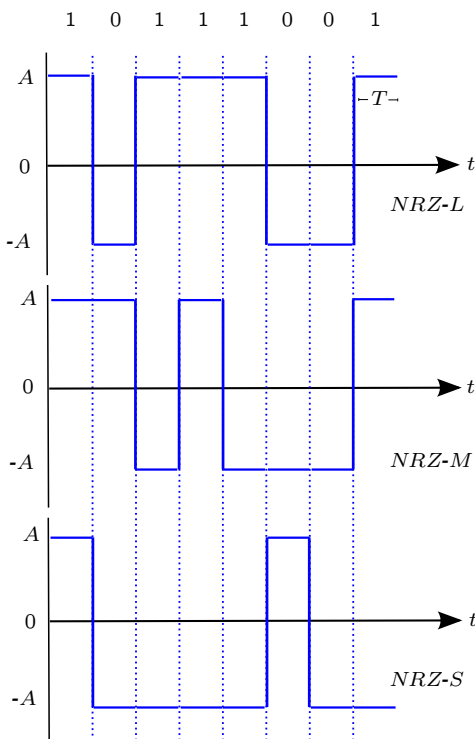


Figura 2.5: Modulaciones en banda-base NRZ de la secuencia **10111001**.

La dificultad de recuperar los ciclos de reloj en el caso de NRZ puede ser resuelta incluyendo más transiciones en la forma de onda. Esto se realiza empleando la denominada modulación RZ (*Return to Zero*). En su versión unipolar, el bit '1' se representa como un pulso de medio periodo de duración, y el '0' como el nivel cero en toda su duración. En el modo bipolar, tanto el '1' y el '0' se representan con niveles de amplitud no-nulos y de signo distinto, con medio periodo de duración, mientras la segunda mitad del periodo está en

nivel nulo. Este formato asegura dos transiciones por bit, mejorando las características de recuperación de ciclos de reloj, con la contraparte de que requiere el doble de ancho de banda que en el caso de NRZ [64, Cap. 2]. Este esquema se representa en la Figura 2.6.

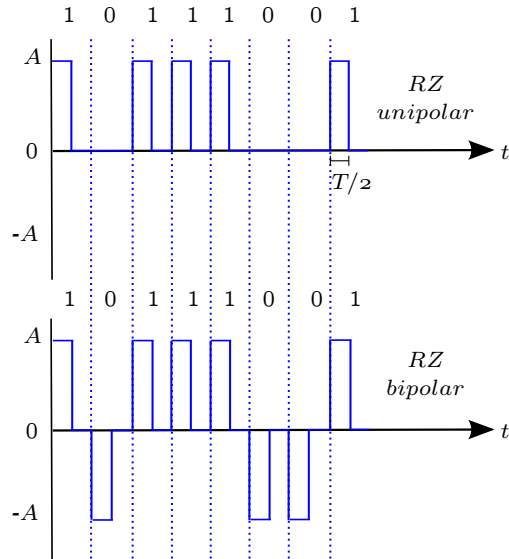


Figura 2.6: Modulaciones en banda-base RZ de la secuencia **10111001**.

## 2.2.2. Modulación en pasa-banda

### Esquemas básicos

En la modulación pasa-banda, una secuencia de símbolos digitales modifican los parámetros de una señal sinusoidal de alta frecuencia, denominada *portadora*. Dado que dicha señal tiene amplitud, frecuencia y fase, son estos parámetros los que pueden modificarse para modular la señal. De este modo, se originan tres esquemas básicos de modulación pasa-banda (Figura 2.7):

1. Modulación por desplazamiento de amplitud ASK (*Amplitude Shift Keying*), en la que cada símbolo '1' genera una ráfaga de la señal portadora de amplitud  $A$ , y un '0' genera una señal de amplitud  $B \neq A$ . Si una de ambas amplitudes es 0, el esquema se denomina OOK (*On-Off Keying*).
2. Modulación por desplazamiento de frecuencia BFSK (*Binary Frequency Shift Keying*), donde el símbolo '1' se representa por una ráfaga de una frecuencia  $f_1$  y el símbolo '0' por una ráfaga de frecuencia  $f_2 \neq f_1$ . Ambas frecuencias se calculan a partir de una frecuencia nominal  $f_c$ , dependiendo de las propiedades deseadas para este esquema (ver sección 2.2.2).
3. Modulación por desplazamiento de fase BPSK (*Binary Phase Shift Keying*), en la que el símbolo '1' se modula con una ráfaga con fase  $\theta^\circ$  y el '0' con una ráfaga con desfase de

$(\theta - 180)^\circ$ . Esta diferencia en fase minimiza la probabilidad de error de demodulación (ver Sección 2.3.2), aunque en un esquema PSK general puede tomar cualquier valor en el rango  $[0^\circ, 360^\circ]$ .

A partir de estos modos básicos de modulación pasa-banda, se han desarrollado otros esquemas con características particulares en fase, frecuencia y amplitud, con desempeños resultantes diversos en cuanto a eficiencia en potencia, eficiencia en ancho de banda y complejidad de implementación [64, Cap. 1].

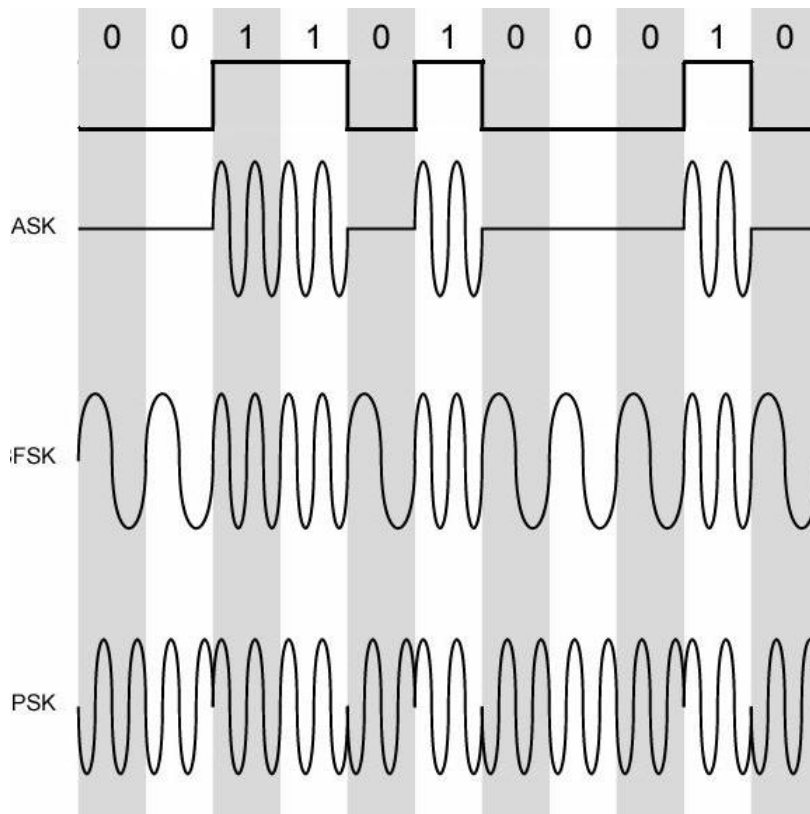


Figura 2.7: Modulaciones básicas pasa-banda.

Cabe señalar que la modulación de los símbolos se realiza, para cada esquema visto, de acuerdo al formato en que se representan en banda-base. En tal caso, si los símbolos son previamente modulados con codificación diferencial (en banda-base), la forma de onda modulada en pasabanda será distinta que si la codificación es no-diferencial, como en el caso de NRZ-L. Esto es de importancia al considerar características tales como eficiencia de potencia, ancho de banda y complejidad de implementación en el diseño de moduladores y demoduladores.

## Características de modulaciones BFSK y BPSK

En modulación BFSK, dos aspectos inciden en su comportamiento. El primero es la *coherencia*, definida como la igualdad de las fases de las dos frecuencias seleccionadas en torno a una frecuencia nominal  $f_c$ ; si tales fases son iguales, el esquema se dice *coherente*, y en caso contrario se denomina *no coherente*. En la modulación coherente, las transiciones de bits se realizan preservando la fase, y por lo tanto la forma de onda mantiene su continuidad en el tiempo.

El segundo aspecto es la *ortogonalidad* de frecuencias; en un esquema coherente, las dos frecuencias  $f_1$  y  $f_2$  son *ortogonales* si ambas son múltiplos de  $1/4T$ , donde  $T$  es el periodo de 1-bit, y además la separación entre ambas debe ser  $2\Delta f = f_1 - f_2 = k/2T$ , con  $k$  entero. En un esquema BFSK no coherente, las frecuencias ortogonales son múltiplos de  $1/2T$  y la separación entre ellas es múltiplo de  $1/T$ . Esto implica que se duplica el ancho de banda requerido para una transmisión no-coherente, considerando la misma tasa de datos  $R_b = 1/T$ . Un esquema FSK modulado en forma coherente, con frecuencias ortogonales con separación de frecuencias igual a  $1/T$  se denomina *FSK de Sunde* y corresponde a un caso especial de las modulaciones CPFSK (*continuous phase FSK*).

Finalmente, es posible demodular una señal coherente tanto de forma coherente como no-coherente, lo que depende de si se recuperan las fases de las frecuencias recibidas; una señal modulada de forma no-coherente, sin embargo, sólo puede ser demodulada de forma no-coherente. Esta última demodulación es la más usual debido a la dificultad de construir un recuperador de portadora que obtenga las fases [64, Cap. 3].

Para caracterizar el comportamiento de la modulación BPSK se utilizan conceptos distintos. En este esquema se modula el dígito binario '1' con una señal sinusoidal  $s_1(t)$  y el símbolo binario '0' con una señal  $s_2(t) = -s_1(t)$ ; dichas señales se denominan *antipodales*. La diferencia en signo de las señales  $s_1(t)$  y  $s_2(t)$  tiene por consecuencia un desfase de  $180^\circ$  entre ellas. La demodulación de una señal BPSK es *coherente* cuando se requiere recuperar la fase de la señal recibida, y sincronizarla con la señal de referencia del correlador respectivo; además debe satisfacerse que la frecuencia de la señal,  $f_c$ , sea múltiplo entero de la tasa de datos  $R_b$ . Aún así, esta condición puede relajarse si  $f_c \gg R_b$ , con una degradación despreciable en la tasa de errores de demodulación.

Una forma opcional es la dada por la modulación PSK *diferencial*, denotada por DEBPSK. En este formato, los bits de datos se codifican diferencialmente, y luego se modulan en fase de la forma descrita previamente. Esta codificación previa a la modulación permite emplear un demodulador diferencial, que toma el símbolo previo demodulado como referencia para calcular el símbolo actual, en lugar de requerir una señal de referencia sincronizada en fase y frecuencia. Un esquema con codificación diferencial en el modulador y demodulación diferencial se denomina DPSK <sup>2</sup>. Esta modulación no requiere la mencionada sincronización, pero su implementación es más compleja y el desempeño con respecto a BPSK coherente es

---

<sup>2</sup>El acrónimo DEBPSK alude sólo al formato codificado de modulación. El término DPSK incluye además la demodulación diferencial.

inferior <sup>3</sup>. En cualquiera de estos mecanismos de modulación, los datos binarios ingresan al modulador representados en formato banda-base NRZ [64, Cap. 4].

## Modulación MSK

El esquema MSK reúne características de modulación en fase, pues adecúa la forma de onda del esquema OQPSK<sup>4</sup>, y se visualiza además como un caso particular de modulación FSK en fase continua (CPFSK) [64, Cap. 5]. Visto de este último modo, MSK corresponde a la mínima separación entre frecuencias que permite generar señales ortogonales coherentes. Es por ello que este esquema es particularmente eficiente en ancho de banda. Existen varias formas de implementar un modulador MSK, basadas en su símil de OQPSK. El demodulador puede construirse en base al de este último esquema, para detectar la señal en forma coherente, o puede realizarse con un esquema de enmallado con algoritmo de Viterbi, puesto que MSK es además un tipo de modulación de fase continua (CPM). La tasa de errores (ver Sección 2.3.2) producidas en el demodulador coherente son idénticas al caso de la modulación BPSK y QPSK. Dado que MSK es una forma de FSK, puede también detectarse de forma no-coherente, con alrededor de 1 [dB] de pérdida de eficiencia en potencia. [64, Cap. 5]. Por las propiedades mencionadas, este esquema conjuga el buen desempeño en errores de BPSK con eficiencia en ancho de banda, y representa una alternativa para la construcción de un módulo de comunicaciones.

## 2.3. Indicadores de desempeño

Para evaluar el funcionamiento de un esquema de comunicaciones o alguna de sus partes, se requieren indicadores que permitan caracterizar su comportamiento y compararlo adecuadamente con otras alternativas. Los señalados a continuación son usuales en el estudio de sistemas de comunicación digital.

### 2.3.1. Energía-bit por densidad espectral

Se menciona en la Definición 2.1.2 la razón señal a ruido  $SNR$ , que en conjunto con el ancho de banda  $W$  caracterizan completamente un canal AWGN. Una manera similar de describir el efecto del ruido sobre señales discretas es la razón energía-bit por densidad espectral de ruido, denotada  $E_b/N_0$ , y corresponde a la energía portada por cada bit transmitido, en razón a la densidad espectral del ruido del canal. Se tiene que

$$(2.7) \quad \frac{E_b}{N_0} = \frac{S}{N_0 r_b} \quad [\text{dB}]$$

---

<sup>3</sup>Para una tasa de errores en demodulación de  $10^{-4}$ , DPSK requiere 0.8 [dB] más de potencia que en BPSK.

<sup>4</sup>OQPSK corresponde a la modulación en fase de 4 símbolos (QPSK), modificada de tal manera que la diferencia de fase en las señales de salida no supera  $\Delta\theta = 90^\circ$  [46, Ap. D].

donde  $r_b$  es la tasa de símbolos, medida en [bps], y  $S$  la potencia de la señal en banda-base portadora de 1 bit, de duración  $T \leq T_b = 1/r_b$  [11]. Es importante notar de la ecuación 2.7 que el nivel de  $E_b/N_0$  depende tanto de la potencia de la portadora, dependiente a su vez de la modulación banda-base usada para representar las secuencias binarias, como de la tasa de símbolos, siendo distinto al caso del  $SNR$  que depende de la potencia de la señal y del ancho de banda.

### 2.3.2. Probabilidad de error de símbolos

Se desprende de las definiciones 2.1.1 y 2.1.2 que al transmitir símbolos en un canal, existe una probabilidad asociada de que dicho canal los modifique, y por lo tanto el demodulador les asignará un valor digital incorrecto. Luego, otra forma de medir el desempeño es considerar los errores cometidos en la demodulación de una señal binaria [5].

En una transmisión de símbolos binarios en un canal AWGN, el demodulador consiste esencialmente de tres etapas: un filtro pasa-bajos, que elimina cierta parte del ruido de entrada, un módulo de muestreo sincronizado de la señal filtrada, y un módulo de decisión de símbolos. En el tercer módulo se decide si el símbolo recibido corresponde a un 1 ó un 0 binario; tal decisión se lleva a cabo comparando el valor de entrada de la señal  $y(t_k)$  con un valor denominado umbral  $U$  (ver Figura 2.8), el cual se escoge de modo que los errores cometidos sean equiprobables para ambos símbolos binarios. Este método de selección se conoce como *hard decision* y es válido sólo en condiciones de ruido gaussiano y con símbolos equiprobables. Si los símbolos no son equiprobables se utilizan otros métodos como máxima probabilidad a-posteriori (MAP) o estimación de máxima verosimilitud (MLE) [5]. En este contexto se establecen sólo las probabilidades de error en demodulación  $P_b$  para señales equiprobables y con energías iguales, en un canal AWGN.

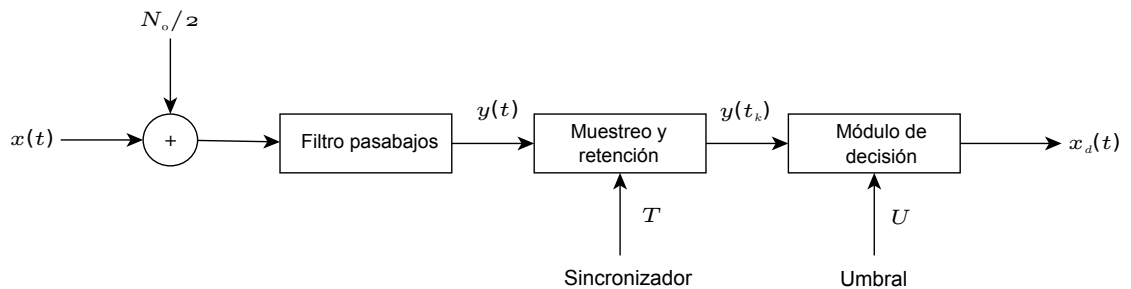


Figura 2.8: Demodulador de símbolos binarios.

La función  $Q(a)$ , relacionada con las probabilidades de error, se define como la probabilidad de que una variable aleatoria  $X$  sea mayor que un valor  $a$  bajo una distribución

gaussiana normal  $\mathcal{N}(0, 1)$  [5]:

$$(2.8) \quad Pr\{X > a\} = Q(a) = \frac{1}{\sqrt{2\pi}} \int_a^{\infty} \exp\left(\frac{-x^2}{2}\right) dx$$

Bajo las condiciones mencionadas, la probabilidad de error  $P_b$  dependerá de la demodulación. Dicha probabilidad también es afectada por la forma de modulación, en particular por la coherencia de fases en FSK y la discontinuidad de la forma de onda en PSK. De este modo, para un esquema PSK binario no diferencial, demodulado coherentemente <sup>5</sup>,  $P_b$  está dada para cada símbolo binario como [64, Cap. 4]

$$(2.9) \quad P_b \text{ (BPSK)} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Para un esquema FSK de Sunde, demodulado en forma coherente, la probabilidad de error es [64, Cap. 3]:

$$(2.10) \quad P_b \text{ (BFSK coh.)} = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

y bajo FSK con demodulación no-coherente y frecuencias ortogonales, dicha probabilidad es [64, Cap. 3]:

$$(2.11) \quad P_b \text{ (BFSK no coh.)} = \frac{1}{2}e^{-E_b/2N_0}$$

### 2.3.3. Tasa de errores de símbolos

Un parámetro empírico de importancia es la tasa de errores en el mensaje recibido en comparación con el mensaje enviado. Esta tasa se denota BER (*bit error rate*). Al recibir una secuencia de largo  $n$  en el receptor, el BER se obtiene dividiendo el número de bits errados en la transmisión por el largo total de la secuencia:

$$(2.12) \quad \text{BER} = \frac{\text{Número de bits erróneos}}{n}$$

Con los parámetros  $E_b/N_0$  y BER es posible comparar esquemas de modulación digital o codificación. A modo de ejemplo, una secuencia de tramas de bits moduladas en BPSK requiere un nivel de potencia  $E_b/N_0 = 4.30$  [dB] para obtener un BER promedio de  $10^{-2}$  (1 bit errado por cada 100 enviados); para lograr dicha tasa de errores con tales secuencias moduladas en BFSK, la potencia necesaria es de  $E_b/N_0 = 8.91$  [dB]. En conclusión, para

---

<sup>5</sup>En adelante se denomina BPSK al esquema PSK binario con modulación no diferencial y demodulación coherente.



obtener en promedio la misma cantidad de errores, modular la secuencia en BFSK redundante en un mayor consumo de potencia comparado a BPSK.

Es importante señalar que, bajo un sistema digital de comunicaciones sin codificación, el BER converge a la probabilidad de error  $P_b$  dada por el esquema de modulación empleado, a medida que aumenta la cantidad de transmisiones de secuencias binarias. Esto pues ambos indicadores son relativos al demodulador. En un esquema de comunicaciones con codificación esto no es así, puesto que el BER se mide entre la secuencia binaria inicial y la secuencia final decodificada; en tal caso se está midiendo el desempeño del decodificador y del demodulador en su conjunto.

### 2.3.4. Ganancia de código

Este indicador señala el ahorro energético consecuente al uso de un esquema de codificación comparado con otro, o con un esquema de transmisión no codificado, para una tasa de errores determinada. Para este último caso, con BER fijo  $P_e$  se tiene que la ganancia de código es:

$$(2.13) \quad \Delta G|_{\text{BER}=P_e} = \left[ \left( \frac{E_b}{N_0} \right)_{\text{No codificado}} - \left( \frac{E_b}{N_0} \right)_{\text{Codificado}} \right]_{\text{BER}=P_e}$$

Si se comparan dos esquemas de codificación en la ecuación anterior, la razón  $E_b/N_0$  no codificado será el del esquema codificado con mayor potencia requerida para  $P_e$ .

## 2.4. Comunicaciones satelitales

### 2.4.1. Descripción general

En el campo de las comunicaciones satelitales, se distinguen diversas órbitas con fines y aplicaciones diferentes. Estas órbitas varían principalmente en cuanto a la altitud de ubicación, desde los sistemas geoestacionarios GEO (*Geostationary Earth Orbit*) con altitud promedio de 36.000 [Km] desde la Tierra, hasta los sistemas de baja órbita LEO (*Low Earth Orbit*) con altitud variable entre 300 y 800 [Km]. Otras órbitas se diferencian por inclinación respecto del Ecuador, excentricidad y sincronía relativa a alguna referencia como el Sol. La órbita de interés del satélite *Cubesat* SUCHAI es una órbita LEO polar, es decir que cruza la Tierra con inclinación de alrededor de 90° respecto al Ecuador.

## Subsistemas de enlaces satelitales

A grandes rasgos, un sistema de comunicaciones satelital consiste de dos segmentos principales:

- **Estación terrestre:** Se ubica en la superficie terrestre, en zonas fijas o vehículos móviles, y su propósito es comunicarse con el satélite, mediante equipamiento como: antenas, amplificadores, equipos de procesamiento (modulación y demodulación, codificación y decodificación, ecualización), equipos de control, etc.
- **Satélite:** Dispone de varios subsistemas, con dimensiones limitadas a la del mismo satélite, que responden a las necesidades y requerimientos específicos del mismo. Los principales subsistemas son [41, Cap. 3]:
  1. Antenas: su función es recibir y transmitir señales en radiofrecuencia hacia o desde las zonas de cobertura deseadas.
  2. Comunicaciones: amplifican las señales recibidas, cambian su frecuencia y la entregan a la(s) antena(s) para su retransmisión a la Tierra. Puede incluir conmutación y procesamiento de señales (multiplexión/demultiplexión, codificación de fuente y/o canal, encriptación).
  3. Energía eléctrica: Suministra electricidad a todos los equipos, con los niveles adecuados de voltaje y corriente, bajo condiciones de iluminación solar o eclipse.
  4. Control térmico: regula la temperatura del conjunto, dada su gran fluctuación entre los periodos de iluminación y eclipse.
  5. Posición y orientación: determina y mantiene la posición y/u orientación del satélite. Estabiliza y orienta correctamente las antenas y paneles de celdas solares.
  6. Propulsión: proporciona incrementos de velocidad y torques para corregir las desviaciones en posición y orientación.
  7. Rastreo, telemetría y comando: intercambia información con la estación terrestre en Tierra para observar el funcionamiento del satélite. Monitoreo del estado del mismo.
  8. Estructural: aloja todos los equipos y le da rigidez al conjunto, en el lanzamiento y en su órbita de funcionamiento.

En el caso de satélites comerciales, en general se prescinde de procesamiento de señales, actuando sólo como repetidores entre dos o más estaciones terrestres. En satélites con fines de investigación, se incluye instrumentación adicional para experimentos, conocida genéricamente como *payload*<sup>6</sup>.

---

<sup>6</sup>También denominada *carga científica*.

## Subsistemas del satélite SUCHAI

El satélite SUCHAI no requiere mecanismos de propulsión, pues es eyectado en su órbita por un dispositivo conocido como P-POD, diseñado con ese objetivo. El subsistema de posicionamiento es opcional, y se analiza la inclusión de componentes que responden al magnetismo de la Tierra para la orientación, llamados *magneto-torquers*.

Se emplea una única antena dipolo, la cual se despliega en una de las caras del satélite tras un lapso de tiempo posterior a la eyección del CubeSat. Su fuente energética la constituyen celdas solares, funcionales en los ciclos donde el satélite recibe radiación solar, y baterías que reservan energía para su disponibilidad en los ciclos de oscuridad (ciclos llamados *eclipses*).

Se realiza también procesamiento de señales, centralizados en la unidad OBC (*On Board Computer*), concebida como un subsistema aislado de Comunicaciones, en contraste a lo señalado en la lista anterior. El subsistema de comunicaciones enlaza el procesamiento de datos con la transmisión y recepción mediante la antena, mediante un protocolo adecuado que incluyen corrección y detección de errores, modulación BFSK, y retransmisión de datos [3]. La telemetría y telecomandos constituyen parte de los subsistemas OBC y comunicaciones.

Se incluyen *payloads* para realizar diversos estudios. Uno lo constituye el estudio del plasma en la ionosfera, mediante el sensor *Langmuir*. Otro experimento está orientado al estudio del comportamiento de partículas en ambiente de *microgravedad*. Otros experimentos están en etapa de estudio para su posible inclusión en esta misión.

Finalmente, el cuerpo del CubeSat está constituido por una estructura cúbica de 10 [cm] de longitud, armado mediante la unión de aristas que dan forma a las caras y vértices del cubo; este armado es suficiente para dar la rigidez necesaria al sistema, además de reducir el peso y dejar huecos en las caras que permiten la colocación de los paneles solares, el subsistema de antena, y los *payloads*.

### 2.4.2. Descripción de enlaces de baja órbita

Un enlace satelital de baja órbita se distingue por las siguientes características:

1. El tiempo de enlace entre la estación terrestre y el satélite es bastante reducido, debido a que a menor altura la velocidad tangencial para mantener al satélite en la órbita deseada aumenta. En el caso de la órbita polar, se debe considerar que mientras el satélite se mueve longitudinalmente, la Tierra rota, por lo que la probabilidad de que pase sobre la estación terrestre, disminuye a una o dos veces al día, con tiempos de paso breve (10 minutos o menos) y, en general, con ángulos de elevación bajos, por lo que se empobrece el enlace en cuanto a niveles de potencia y cantidad de información enviada o recibida.
2. Debido a la alta velocidad orbital del satélite, se produce efecto Doppler en la señal de enlace. En la banda de frecuencias UHF (entre 300 y 3000 [MHz]), el desplazamiento en

frecuencia puede alcanzar los 10 [kHz] [32]. Este efecto se puede corregir si se conocen la altura y velocidad tangencial del satélite.

3. El tamaño del satélite impone restricciones a los subsistemas importantes como Energía y Comunicaciones. En el caso de Energía, la alimentación por energía solar se ve limitada al tamaño reducido de las placas solares; en el caso de las Comunicaciones, la potencia de la señal es limitada, y las antenas aptas para esta aplicación ofrecen baja ganancia, tanto por su tamaño como por su baja direccionalidad. Estas restricciones son propias de satélites pequeños, incluyendo *Cubesats*.
4. Dependiendo de la capa atmosférica donde el satélite está en órbita, existen diversos efectos que distorsionan la señal. El principal es el efecto Faraday que se produce en la ionosfera, que consiste en la rotación del plano de polarización. El vector de campo eléctrico puede girar hasta  $150^\circ$  en la banda UHF [41, Cap. 5]. Otro fenómeno presente es el centelleo (*scintillation*), fluctuaciones rápidas de amplitud, fase y ángulo de recepción de las ondas, producido por los cambios en el índice de refracción en la troposfera [41, Cap. 5]. La absorción atmosférica es de escasa importancia, pues las frecuencias de resonancia de elementos como agua, oxígeno y dióxido de carbono son importantes a partir de los 22 [GHz]. La lluvia, las nubes y la neblina son de importancia en frecuencias sobre 10 [GHz]; la degradación está ligada en tal caso al tamaño relativo de las gotas con respecto a la longitud de onda (en el caso del Cubesat  $\lambda \approx 68$  [cm] para 435 [MHz]) [41, Cap. 5]. Por último, si se compara un satélite en baja órbita con un sistema satelital a mayor altitud, el primero pierde menos potencia por dispersión.

Estos factores en conjunto inciden en la calidad de la transmisión, degradando la señal. Esto representa un problema en el caso de SUCHAI, pues puede perturbar tanto la transmisión hacia el satélite de telecomandos que ejecuten las acciones requeridas, como la recepción de señales en la estación terrena, que deben contener información fiable del estado del satélite y de los datos de los *payloads*. Es por ello que el dispositivo de comunicaciones implementa corrección de errores para los telecomandos, y detección de tramas erróneas en telemetría que efectúe la retransmisión de ellas.

El problema expuesto plantea la necesidad de implementar sistemas de control de errores, para enlaces de subida (*uplink*) o de bajada (*downlink*). A pesar de que esta misión agrega tal sistema, es deseable estimar su desempeño efectivo, y a la vez estudiar otras alternativas de corrección y/o detección de errores para un futuro dispositivo de comunicaciones.

## 2.5. Codificación para control de errores

### 2.5.1. Esquemas de control de errores

Debido a las características enumeradas en la sección 2.4.2, en un canal de comunicaciones de baja órbita puede ser una ventaja implementar algún sistema de protección

de la información que disminuya la tasa de errores en la transmisión o recepción. Esto es importante para la transmisión de datos experimentales hacia la estación terrestre, así como para evitar errores en los telecomandos.

Existen dos esquemas principales que manejan los errores introducidos por el canal. El primero consiste en un esquema de repetición, denominado ARQ (*Automatic Repeat reQuest*), y consiste en reenviar una trama de datos o telecomandos, si tras un tiempo de espera llamado *timeout*, el transmisor no ha recibido una trama específica conocida como *acknowledgment*, notificando que el receptor ha recibido correctamente el comando enviado. Esta situación se repite hasta que el transmisor reciba el *acknowledgment*. Otro mecanismo de control de errores es el uso de codificación de canal, denominado FEC (*Forward Error Correction*). Éste se basa en la extensión del mensaje agregando símbolos de redundancia en función de algún algoritmo implementado en el codificador. Al ser recibido, dicho mensaje extendido pasa por el decodificador, que verifica el mensaje original y corrige los errores introducidos por el canal, en función de su capacidad de corrección intrínseca. La ventaja del esquema FEC sobre el ARQ es que permite realizar corrección sin reenvío de información, y su capacidad de corregir errores puede incrementarse dependiendo del esquema de codificación. Estas razones justifican el amplio uso de este mecanismo de control de errores en enlaces satelitales.

## 2.5.2. Códigos lineales de bloques

Los códigos lineales de bloques corresponden informalmente a un conjunto de estrategias que forman parte del mecanismo de control de errores FEC. Tales estrategias reciben también la denominación de *esquemas de codificación*. Cada esquema en particular agrega redundancia al mensaje original mediante algún algoritmo específico, lo cual es aprovechado en el receptor para recuperar el mismo de la forma más fidedigna posible.

Todo esquema de codificación se desarrolla a partir de las propiedades del álgebra lineal y la teoría de campos finitos, por lo que es posible establecer sus cualidades específicas analizando su estructura y propiedades algebraicas, en el contexto dado por algunos conceptos propios del ámbito de Codificación, presentados más adelante. Esto debe incluir la modelación del canal, que permite estimar el comportamiento real de tales códigos. La teoría y propiedades del álgebra lineal, y la teoría de campos finitos con sus consecuencias en la constitución de códigos lineales, pueden ser revisados con detalle en la literatura [9], [11], [13], [40], [57]. La revisión dada en este Capítulo pretende sólo establecer los elementos necesarios para su aplicación posterior.

La siguiente es una definición operacional de los códigos lineales de bloques.

**Definición 2.5.1** *Un **código lineal** de bloques, de tamaño  $M$  sobre un alfabeto de  $q$ -símbolos es un conjunto de  $M$  secuencias  $q$ -arias de largo  $n$ , llamadas **palabras de código**, que son transmitidos a través de algún canal determinado.*

Usualmente  $M = q^k$ ,  $k \in \mathbb{Z}$  y  $q = 2$ , es decir el alfabeto es binario [9]. El espacio vectorial

algebraico donde residen estos códigos se denomina **campo de Galois**<sup>7</sup>, y denotado por  $GF(q)^n$  bajo la descripción anterior.

Bajo la definición previa, queda explícita la diferencia entre un código lineal, entendido como un conjunto de palabras de código, y un esquema de codificación, que corresponde a la estrategia de asignación de tales palabras de código para cada mensaje, y su posterior recuperación mediante un algoritmo de decodificación. Es posible hallar en la literatura el uso indistinto del término *código* para ambos conceptos, pero debe tenerse en cuenta esta diferencia conceptual expuesta.

Existen varios esquemas de códigos lineales, distinguidos por sus propiedades algebraicas y formas de construcción; esto conlleva diversos desempeños para cada uno de ellos que dependen también del canal donde son aplicados. Del mismo modo, cada uno se caracteriza matemáticamente de forma distinta, pero siempre se podrán representar equivalentemente como operaciones matriciales y vectoriales, como se indica posteriormente.

Bajo la definición 2.5.1, para el caso de un alfabeto binario, una palabra de código lineal de largo  $n$  representa un bloque de información de largo  $k$  codificado. El conjunto de todas las palabras de código conforman un conjunto denotado  $\mathcal{C}(n, k)$ . Es decir, un mensaje de bits de largo  $k$  es proyectado en una secuencia de largo  $n$ , la cual se transmite por el canal. En el extremo receptor, el decodificador recibe la secuencia alterada por el canal, y es decodificado en un mensaje de largo  $k$  que es una estimación del original. Esto se muestra gráficamente en la Figura 2.9. Además, dados los parámetros  $n$  y  $k$ , se describe un código de bloque por la llamada **tasa de codificación**  $R = k/n$ ; con ella se conoce la proporción del mensaje en la secuencia transmitida con respecto al largo total de esta última. Por último, si la transmisión de símbolos es a una tasa de  $1/T$  [símbolos/segundo] (baud rate) en un sistema codificado, la información comunicada tiene una tasa de datos de  $R/T$  [símbolos/segundo].

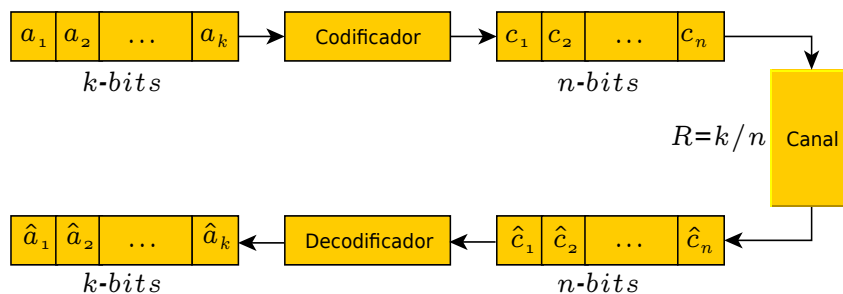


Figura 2.9: Codificación de bloques.

La tabla 2.1 ilustra las operaciones en el alfabeto binario, el cual se denota por  $GF(2)^n$ , siendo  $n$  el largo de las palabras de código. Las operaciones se llevan a cabo bit a bit. Una excepción importante la conforman los códigos lineales llamados Reed-Solomon [48], cuyos alfabetos tienen más de dos elementos por definición, y por tanto su operatoria algebraica es diferente.

<sup>7</sup>Traducido del inglés *Galois field*

+	0	1
0	0	1
1	1	0

x	0	1
0	0	0
1	0	1

Tabla 2.1: Operaciones en  $GF(2)$ .

**Definición 2.5.2** La **distancia de Hamming** entre dos palabras de código corresponde al número de símbolos en que ambas difieren. La **distancia mínima** de un código,  $d_{min}$ , es la mínima distancia entre dos palabras del código [9].

Como ejemplo, si se tienen las palabras de código  $c_1 = [0\ 1\ 1\ 1\ 0]$  y  $c_2 = [0\ 1\ 0\ 0\ 0]$ , ambas difieren en el tercer y cuarto símbolo; por lo tanto, la distancia Hamming entre ambas palabras es  $d = 2$ . Notar que la distancia de Hamming es una métrica y, por consiguiente, cumple las propiedades respectivas (no-negatividad, simetría, desigualdad triangular).

**Definición 2.5.3** El **peso de Hamming** de una palabra de código  $v$  es el número de símbolos distintos de '0' en  $v$ . El **peso mínimo** corresponde al menor peso de Hamming entre las secuencias pertenecientes a un código lineal, excluyendo aquella que contiene sólo símbolos '0'.

Los conceptos de *distancia* y *peso* permiten caracterizar cualquier código de bloques. Más aún, son indistinguibles en el caso de distancia y peso mínimos, como establece el siguiente Teorema.

**Teorema 2.5.1** La mínima distancia de Hamming en un código lineal de bloques es igual al peso mínimo de sus palabras de código [13, Cap. 3].

**Definición 2.5.4** La **capacidad de corrección** es una medida de cuántos errores es posible corregir en una palabra de código. En general para un código lineal  $C(n, k)$ , la capacidad  $t$  será un entero que satisfaga [11, Cap. 2]:

$$(2.14) \quad 2t + 1 \leq d_{min} \leq 2t + 2$$

Donde  $d_{min}$  es la distancia mínima de Hamming.

De la ecuación 2.14 se infiere que la capacidad de corrección se incrementa con la distancia mínima. Esto es intuitivo si se observa que cuanto mayor es esta distancia, es más fácil diferenciar entre palabras de código, descontando el efecto alterador del canal.

En algunas aplicaciones puede ser de interés sólo detectar las secuencias erróneas, solicitando al transmisor el reenvío de las mismas. En tal caso, se requiere conocer la sensibilidad del receptor para detectar los errores en los códigos enviados. El concepto definido a continuación permite cuantificar dicha sensibilidad.

**Definición 2.5.5** La **capacidad de detección** mide cuántos errores es posible detectar en una palabra de código. En un código lineal  $\mathcal{C}(n, k)$ , la capacidad de detección  $d_{det}$  es [13, Cap. 3]:

$$(2.15) \quad d_{det} = d_{min} - 1$$

Donde  $d_{min}$  es la distancia mínima de Hamming.

De las ecuaciones 2.14 y 2.15 se infiere que en los códigos lineales de bloque la capacidad de detección  $d_{det}$  es al menos dos veces mayor que la capacidad de corrección  $t$ . Por consiguiente, si se emplea un esquema con una capacidad de corrección adecuada para un canal determinado, puede ser de utilidad añadir un sistema de retransmisión que aproveche la detección de secuencias erradas, si la situación lo amerita.

### Descripción matricial de códigos lineales

Los códigos lineales pueden ser representados por operaciones algebraicas con vectores y matrices. De este modo, si en un codificador ingresa una palabra de datos de largo  $k$  bits, se representa tal palabra mediante un vector fila  $a$  de  $k$ -componentes, y se mapea dicho vector a una palabra de código  $c$  de  $n$ -componentes mediante la denominada **matriz generadora**  $\mathbf{G}$ , de dimensión  $k \times n$ . Luego la operación ejecutada en el codificador es

$$(2.16) \quad c = a\mathbf{G}$$

Todo código lineal puede ser descrito por la matriz  $\mathbf{G}$ . Esto tiene la ventaja concreta de poder construir codificadores a partir de bloques lógicos digitales.

Análogamente, en el extremo receptor, la llamada **matriz de chequeo de paridad**  $\mathbf{H}$  es tal que permite comprobar si la palabra de código recibida  $v$  corresponde en efecto a la original transmitida, mediante el producto

$$(2.17) \quad v\mathbf{H}^T = z$$

El vector  $z$  obtenido de esta operación se denomina **síndrome**, y es el mecanismo más básico para detectar y/o corregir errores en las secuencias recibidas. Tal mecanismo se llama *detección de síndrome*.

Cuando el síndrome es no nulo se tiene que la palabra de código ha sufrido perturbaciones en el canal. Si el producto es nulo implica que  $v$  es una palabra de código que pudo ser generada por la matriz  $\mathbf{G}$ , aunque puede ser una palabra erróneamente producida por el canal mismo, a partir de una palabra original distinta (ver Figura 2.10). Esto implica que toda secuencia errónea que tenga la forma de una palabra de código queda fuera de la detección y por lo tanto no se puede corregir mediante chequeo de síndrome.



Si el síndrome es nulo y corresponde en efecto a una palabra de código, se cumple la siguiente relación:

$$(2.18) \quad z\mathbf{H}^T = c\mathbf{H}^T = 0$$

De la ecuación 2.16 se tiene que

$$(2.19) \quad c\mathbf{H}^T = a\mathbf{G}\mathbf{H}^T = 0$$

Esto implica que (asumiendo  $a \neq 0$ ):

$$(2.20) \quad \mathbf{G}\mathbf{H}^T = 0$$

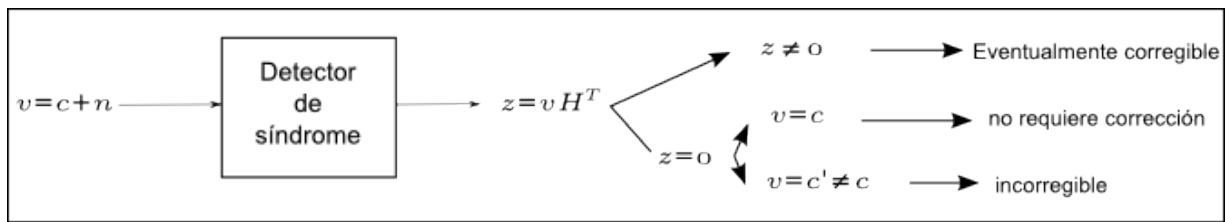


Figura 2.10: Detección de síndrome en el decodificador.

Cuando un código lineal es representado por una matriz generadora  $\mathbf{G}$ , existen muchas matrices  $\mathbf{H}$  posibles. Una manera de obtener fácilmente una matriz de chequeo, dada una matriz generadora, es mediante una forma equivalente de representar los códigos lineales; la matriz  $\mathbf{G}$  puede ser re-escrita como

$$(2.21) \quad \mathbf{G} = [\mathbf{P} \quad \mathbf{I}_k]$$

donde  $\mathbf{I}_k$  es una matriz identidad de  $k \times k$ , y  $\mathbf{P}$  es de  $k \times (n - k)$ . Esta reformulación de  $\mathbf{G}$  es posible mediante operaciones fila elementales o permutación de columnas, y la matriz resultante produce un código equivalente al original [9, Cap. 3]. Esta forma de  $\mathbf{G}$  se denomina **forma sistemática** de la matriz generadora. Un codificador que genera palabras de código con la matriz  $\mathbf{G}$  en su forma sistemática se denomina **codificador sistemático**; y en caso contrario será **no-sistemático**.

Dada la ecuación previa, se deduce fácilmente una expresión para  $\mathbf{H}$ , pues

$$(2.22) \quad \mathbf{G}\mathbf{H}^T = [\mathbf{P} \quad \mathbf{I}_k][\mathbf{I}_{n-k} \quad \mathbf{P}^T] = -\mathbf{P} + \mathbf{P} = 0$$

de donde

$$(2.23) \quad \mathbf{H} = [\mathbf{I}_{n-k} \quad \mathbf{P}^T]^T$$

Como consecuencia inmediata de la codificación sistemática, cuando una palabra de datos es codificada mediante dicha forma matricial, la palabra de código resultante contendrá los

bits de redundancia en los primeros  $n - k$  bits, seguida de la misma palabra de datos en los  $k$  bits posteriores <sup>8</sup>. Por otra parte, es conocido que todo código lineal es equivalente a otro cuyo codificador es sistemático [9]; por lo tanto, si dicho equivalente es conocido, se reduce la complejidad del codificador al asignar palabras de código mediante la sola concatenación de símbolos redundantes, además de simplificar el hallazgo de la matriz de chequeo  $\mathbf{H}$ .

### 2.5.3. Esquemas de codificación más conocidos

#### Códigos Hamming

Este esquema constituye uno de los primeros códigos lineales estudiados. Fueron descubiertos por R. Hamming en 1950, quien junto a C. Shannon estableció las bases de la teoría del control de errores [27]. Un código Hamming, cuyos mensajes son de largo  $k$  y las palabras de código son de largo  $n$ , debe satisfacer que  $n = 2^m - 1$  y  $k = 2^m - m - 1$ , con  $m \geq 3$ . La decodificación se realiza mediante detección de síndrome. La capacidad mínima de corrección de los códigos Hamming es  $t = 1$ , para  $m = 3$ . Una aplicación usual de estos códigos es el Hamming extendido, que añade un bit de redundancia al código Hamming(7, 4), convirtiéndolo en Hamming(8, 4) capaz de corregir un error y detectar dos, método denominado SECDED (*Single error correction, double error detection*) y utilizado en memorias D-RAM [42].

#### Códigos cíclicos

Descubiertos por Prange en 1957 [43], un código cíclico es generado al rotar las posiciones de una palabra de código dada, es decir, si se tiene una palabra de código  $c = (c_0, c_1, \dots, c_{n-1})$ , la rotación  $i$ -ésima del mismo dada por  $c^{(i)} = (c_{n-i}, c_{n-i+1}, \dots, c_{n-1}, c_0, c_1, \dots, c_{n-i-1})$  es también una palabra de código. Es usual describir estos códigos como polinomios en una variable implícita  $X$ , donde el coeficiente  $i$ -ésimo corresponde al bit  $i$ -ésimo de la palabra de código, y el exponente de la variable es la posición de tal bit.

Como ejemplo, si se tiene la palabra de código  $c = (1, 1, 0, 1)$ , ésta se representará polinomialmente como  $p(X) = 1 + X + X^3$ . El grado máximo del polinomio representante de una secuencia es  $n - 1$ , donde  $n$  es el largo de dicha secuencia [11].

Al rotar las posiciones de una misma palabra de código, en algún momento se genera un polinomio equivalente que tiene el mínimo grado entre todas las rotaciones. Dicha secuencia es el **generador polinomial** del código, y basta multiplicarlo por  $X^i$  para obtener la rotación  $i$ -ésima del mismo. Siendo códigos de bloque, los cíclicos se pueden describir con una matriz generadora  $\mathbf{G}$  y una de chequeo de paridad  $\mathbf{H}$ , donde la matriz  $\mathbf{G}$  estará formada por los generadores polinomiales rotados de posición. Estas matrices también pueden escribirse de

---

<sup>8</sup>En un canal con errores de ráfaga, la transmisión de una trama codificada en forma sistemática podría dañar sucesivamente varios bits de redundancia o de información, produciendo una tasa mayor de errores. Esto puede evitarse añadiendo un *entrelazador* que distribuya uniformemente los bits en la secuencia antes de ser transmitidos.

forma sistemática. En cuanto a la decodificación puede utilizarse la detección de síndrome, pero existen mejores algoritmos que aprovechan la estructura cíclica del código como el algoritmo de Meggit [37] o el de captura de error [13].

Este esquema es utilizado en el estándar Ethernet, en el llamado código de redundancia cíclica (CRC), que forma parte del *data frame* del protocolo; tal mecanismo consiste en añadir al bloque de datos una redundancia de 32 bits mediante un polinomio generador conocido [11, Cap. 3].

## Códigos BCH

El esquema BCH (Bose, Chaudhuri [10], Hocquenghem [28]) lo constituyen códigos cíclicos cuya característica distintiva es la mejora en la capacidad de corrección de errores, tanto con alfabetos binarios o no-binarios, es decir con dos o más símbolos. En este sentido, son una generalización de los códigos Hamming, pues es posible seleccionar la capacidad de corrección mediante la variación de la tasa de código. Los esquemas BCH deben satisfacer que para todo  $m \geq 3$ ,  $n = 2^m - 1$ . En la Tabla 2.2 se muestra la capacidad de corrección de un código BCH con  $m = 6$ :

$k$	$t$	$R$
57	1	0.90
51	2	0.81
45	3	0.71
39	4	0.62
36	5	0.57
30	6	0.48
24	7	0.38
18	10	0.29
16	11	0.25
10	13	0.16

Tabla 2.2: Capacidad de corrección  $t$  de códigos BCH( $n, k$ ) en función de  $R = k/n$  con  $n = 63$ . Tabla extraída de [42]

Estos códigos se extienden a diversas aplicaciones tales como televisión satelital y protección de datos en medios de almacenamiento de estado sólido (discos duros, DVDs, códigos de barra). En particular, en aplicaciones satelitales ha formado parte de estándares de comunicaciones de organizaciones como el CCSDS<sup>9</sup> [18]. El dispositivo de comunicaciones del proyecto SUCHAI emplea un código BCH (63,56) [3], es decir, un BCH (63,57) al cual se le ha eliminado un bit de información, técnica conocida como *acortamiento*<sup>10</sup>. A la trama resultante de 63 bits se añade un bit '0' que completa una secuencia de 64 bits (8 bytes) [2]. La

<sup>9</sup>La revisión del estándar de 2010 ha sido la última donde se incluye este esquema. En revisiones posteriores se ha optado por técnicas más recientes para control de errores.

<sup>10</sup>Del inglés *shortening*.

mejora que introduce esta modificación es la reducción a la mitad de los mensajes posibles en el código original, disminuyendo la cantidad de unidades digitales requeridos en el codificador, tales como *flip-flops* y sumadores, y requiriendo menor tamaño de *buffer* para decodificación. Sin embargo, la tasa de código cae de  $R = 0.9$  a  $R = 0.88$ , disminuyendo la cantidad efectiva de datos por trama en la transmisión. De la Tabla 2.2 se aprecia que la capacidad de corrección no aumenta, pues habría que reducir el largo del mensaje a  $k = 51$  bits para que  $t = 2$ . De las Ecuaciones 2.14 y 2.15, no es posible concluir directamente si la capacidad de detección  $d_{\text{det}}$  se incrementa, dado que  $t$  sólo la acota inferiormente<sup>11</sup>.

## Códigos Reed-Solomon

Los códigos Reed-Solomon [48] corresponden a un subconjunto de los BCH, y emplean alfabetos no-binarios, es decir son un subconjunto del espacio de  $GF(q)$  con  $q = p^m$ ,  $m \in \mathbb{N}$  y  $p$  primo. Como tales, satisfacen las mismas propiedades algebraicas que los BCH y, por ende, los cíclicos. Con frecuencia,  $p = 2$ , lo que permite una representación binaria de los símbolos (usando más de un bit por símbolo). Una propiedad interesante de estos códigos es que son aptos para ser acortados (*shortened*), es decir, la palabra de código puede ser acortada eliminando símbolos del mensaje y de redundancia manteniendo la tasa  $r = k/n$  o pueden ser perforados (*punctured*), donde símbolos de redundancia se eliminan acortando el código y aumentando la tasa  $r$  [42]. Los decodificadores para códigos Reed-Solomon son similares a los de BCH, dada su similaridad estructural. Un esquema habitual de decodificación para ambos casos es el algoritmo de Berlekamp-Massey [6], [35]. Estos códigos tienen un buen desempeño para errores producidos en ráfaga. En aplicaciones prácticas se suele implementar en serie con esquemas convolucionales, conformando un codificador concatenado de uso amplio en comunicaciones satelitales, como el caso de la sonda espacial *Voyager* [14]. En el contexto de satélites de baja órbita, la compañía GomSpace ofrece un transpondedor que emplea un esquema Reed-Solomon, que codifica secuencias 256-arias de datos de largo  $k = 223$  bits en tramas de  $n = 255$  bits [24], denotado por RS(255,223). El mismo dispositivo ofrece un esquema convolucional (ver siguiente Sección) con características similares a las ofrecidas por el transpondedor del proyecto SUCHAI [3]. Cabe señalar que el dispositivo de GomSpace tiene herencia de vuelo, es decir, su diseño ha sido empleado en satélites ya lanzados; éstos son el AAUSAT y SSETI-Express [3].

## Códigos convolucionales

Estos códigos, a diferencia de los previamente mencionados, no se basan en bloques de largo fijo  $n$ , sino que se caracterizan por codificar la secuencia completa, de manera continua. En este caso, los parámetros  $k$  y  $n$  corresponden al número de entradas y salidas del codificador, respectivamente. En este esquema cobra especial importancia el codificador, pues su implementación se basa en unidades de memoria o *delays*, unidas por sumadores y multiplicadores. La cantidad de *delays* se denota por  $K$ , e influye en la capacidad de

---

<sup>11</sup>En tal caso se requiere conocer la distancia mínima de Hamming, o equivalentemente, el peso mínimo del código.

corrección del código. De este modo un código convolucional se describe por  $\mathcal{C}(n, k, K)$ . El circuito del codificador constituye lo que se conoce como máquina secuencial de estado finito (FSSM).

La manera más práctica de describir estos códigos es mediante su representación *trellis* (enmallado) donde se describen los cambios de estado de los *delays* a través del tiempo, incluyendo las entradas y salidas del circuito codificador. El algoritmo de decodificación desarrollado por A. Viterbi [56] se basa en la comparación del código recibido con las secuencias del *trellis*, con lo que descarta desde el principio las opciones de código menos probables para estimar el mensaje original. Heurísticamente, un proceso de decodificación cuyo enmallado es de largo  $5K$  es suficiente para una decodificación con alta probabilidad de ser correcta [11]. Esta cualidad del decodificador Viterbi es consecuencia de que el algoritmo de decodificación sea de máxima verosimilitud (MLD). Para códigos con mayor cantidad de *delays* existen otros algoritmos, tales como los secuenciales [60] con mejor desempeño aunque complejidad superior [11].

## 2.6. Diseño de hardware digital

### 2.6.1. Field Programmable Gates Array

Un arreglo de compuertas de campo programable, conocida ampliamente como FPGA por su sigla en inglés, corresponde a un dispositivo programable para múltiples propósitos. Como dispositivo digital, cualquier implementación corresponderá en la práctica a un arreglo de compuertas lógicas, programable tanto por código directo ó por arreglo de compuertas mediante interfaz gráfica.

El elemento minimal común a las FPGA consiste en una celda lógica, a su vez constituida por una tabla de valores LUT (*Look-Up Table*) y un elemento de memoria. Depende de cada fabricante la distribución espacial y organización de estas unidades lógicas.

La plataforma utilizada en este trabajo es una Spartan 3E, del fabricante Xilinx. La Figura 2.11 muestra el esquema organizativo de las unidades lógicas que esta compañía implementa. En este caso, la unidad funcional lógica es el *slice*; cuatro de éstas constituyen un bloque lógico configurable CLB, siendo este último replicado a lo largo y ancho de la FPGA. Se añaden bloques de entrada/salida (I/O Logic), de memoria RAM y de control de reloj digital.

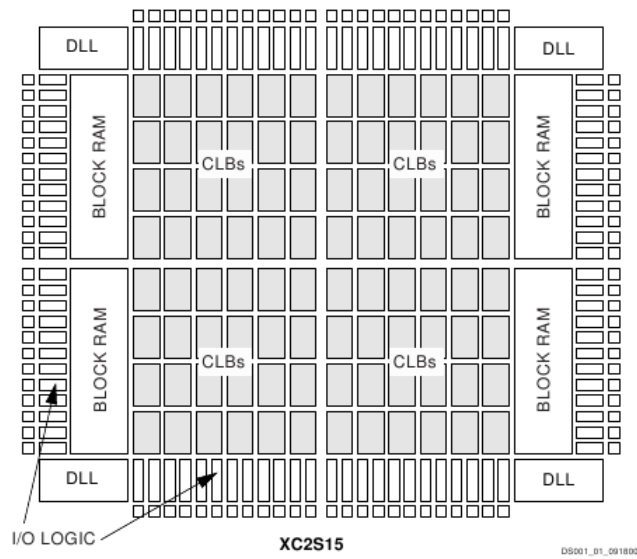


Figura 2.11: Bloques funcionales de FPGA Xilinx.

La FPGA se comercializa como chip para ser integrado según lo estime el usuario, o como parte de tarjetas de desarrollo que añaden múltiples funcionalidades, como puertos de entrada y salida (puerto USB, VGA, RS-232, Ethernet), y otros chips con funcionalidad similar a la FPGA, tales como CPLD, DSP, etc. La Figura 2.12 muestra un modelo de FPGA idéntico al que se utiliza en este trabajo. Para referencias más detalladas, ver manuales de configuración y de FPGA en [62], [63].

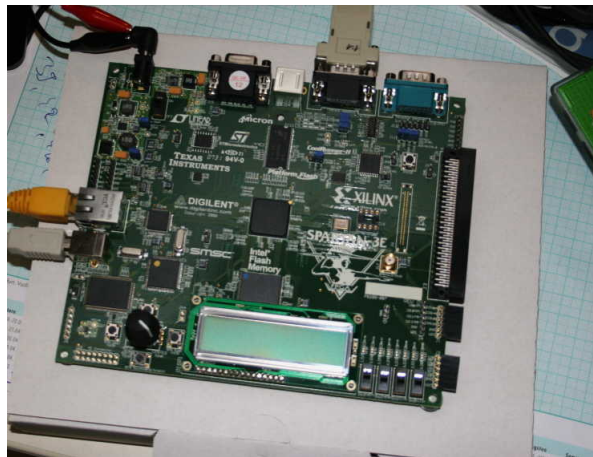


Figura 2.12: Tarjeta Spartan 3E.

Es posible programar el dispositivo mediante lenguajes de programación, conocidos como lenguajes de descripción de hardware (HDL), tales como VHDL o Verilog. Se puede programar también realizando esquemáticos de compuertas lógicas. Este diseño del dispositivo a implementar debe incluir la asignación de sus puertos de entrada/salida a los puertos disponibles en la tarjeta de desarrollo o placa del usuario. Tras esta etapa de diseño,

el programa ISE realiza varias compilaciones, que comprenden la correctitud del mismo, factibilidad de implementación en el chip y asignación de puertos, y optimización cuando es posible. En este procedimiento se generan varios archivos que contienen los atributos del diseño. Finalmente, se implementa el diseño mediante la transferencia de un archivo (*bitstream*) al chip, mediante un puerto JTAG.

## 2.7. Estado del arte

### 2.7.1. Codificación en canales satelitales

Los esquemas de codificación revisados en la Sección 2.5.3 corresponden a métodos clásicos de control de errores. En varias misiones de exploración espacial, el esquema preponderante usado ha sido el convolucional, inicialmente con decodificación secuencial, y posteriormente con decodificación de Viterbi [14, p. 5]. Este último esquema es adoptado en el estándar CCSDS [18], que agrupa los estándares utilizados por las agencias espaciales NASA (estadounidense) y ESA (europea) [14, p. 5]. Otros esquemas de uso extendido en misiones espaciales son los concatenados, conformados por un código interno (*inner-code*) convolucional y un código externo (*outer-code*) Reed-Solomon [19]. Este esquema de concatenación también es actualmente incluido en el CCSDS, y ha sido objeto de múltiples mejoras [14, p. 7]. El año 1993, Berrou, Glavieux y Thitimajshima publican el hallazgo de una nueva especie de codificación, denominados *códigos Turbo*, caracterizados por ser esquemas convolucionales concatenados en paralelo, en combinación con un entrelazador pseudoaleatorio [7]. Este esquema tiene la característica de lograr desempeños bastante más cercanos al *límite de Shannon* [14, p. 8], y forma parte del estándar CCSDS [18].

Una familia de codificación que suscita interés actualmente la constituyen los códigos de chequeo de paridad de baja densidad (LDPC), que pueden alcanzar desempeños muy cercanos al límite de Shannon, a medida que el largo del bloque transmitido se incrementa. Estos fueron originalmente desarrollados por R. Gallager en 1962 [23], aunque probablemente no considerados debido a la superioridad práctica de los códigos concatenados [34, p. 1]. Estos códigos fueron redescubiertos en los trabajos de MacKay y Neal en 1996, quienes estudiaron su desempeño en canales Gaussianos [33], [34], y desde entonces han sido objeto de bastante análisis, reflejado en múltiples publicaciones IEEE, y se ha incorporado en estándares de televisión digital *DVB*, redes inalámbricas de área local (Wi-Fi) y en comunicaciones satelitales [18].

Debido a la escasez de entradas no nulas de la matriz  $\mathbf{H}$ , la decodificación de los códigos LDPC es de baja complejidad, en contraste a los esquemas clásicos. Se han desarrollado subclases de LDPC que permiten además simplificar la implementación del codificador [22], [31], [30], [49], [52], [65] y otras que mejoran el desempeño [12]. También se ha estudiado el desempeño de este código concatenado con BCH, con ganancias obtenidas de hasta 8 [dB] en sistemas de geoposicionamiento [38].

Basado en la subclase de los denominados *LDPC cuasi-cíclicos* (QC-LDPC), F. Olivier

plantea en 2010 su empleo en satélites de baja órbita, simulando diversos parámetros e implementando un prototipo en FPGA [42], y a la fecha está siendo integrado al satélite ISHS-2, desarrollado para aplicaciones de detección en agricultura [59].

### 2.7.2. Esquemas de codificación en FPGAs

Existe una gran variedad de implementaciones de esquemas de codificación en hardware, en particular para FPGAs, debido a su versatilidad y tamaño, fácilmente adaptable a una placa de pequeñas dimensiones, y por lo tanto apta para su empleo en satélites pequeños. Por otra parte, a causa de su costo relativamente bajo se ha popularizado su uso en diseños de comunicaciones digitales y procesamiento de señales.

Para esquemas convolucionales, el esfuerzo se ha enfocado en reducir la complejidad del algoritmo de Viterbi [25], [47], [66]. También se han implementado esquemas concatenados, variando los códigos internos y externos [15], [39], [54]. Del mismo modo, códigos Turbo [61] y LDPC [53], [58] han sido diseñados en FPGA. Existe bastante documentación en páginas web de desarrollo en HDL, y algunos fabricantes de FPGA tales como Xilinx y Altera ofrecen licencias comerciales para el uso de implementaciones hechas por ellos.

En cuanto a la factibilidad de emplear las FPGA para aplicaciones satelitales, esta alternativa ya ha sido estudiada con resultados favorables [26], [50]. Por otro lado, los fabricantes han incorporado la creciente exigencia de FPGAs aptas para el uso espacial, lo que significa mayor resistencia a la radiación y fluctuaciones de temperatura. Entre los componentes desarrollados con este fin están las series Virtex de Xilinx<sup>12</sup>, RTAX de Actel<sup>13</sup> y AT de Atmel<sup>14</sup>. El código LDPC estudiado por Olivier fue implementado en un modelo Virtex-5 de Xilinx [42, p. 69].

---

<sup>12</sup>Ver <http://www.xilinx.com/esp/aerospace-defense/space/index.htm>

<sup>13</sup>Ver <http://www.actel.com/products/milaero/rtaxs/default.aspx>

<sup>14</sup>Ver [www.atmel.com/Images/doc4015.pdf](http://www.atmel.com/Images/doc4015.pdf)



# Capítulo 3

## Simulaciones computacionales

### 3.1. Simulaciones canal binario simétrico

#### 3.1.1. Planteamientos previos

En primera instancia, se realizan simulaciones de diversos códigos lineales para obtener datos numéricos y así fijar una base adecuada que permita comparar cuantitativamente los diversos esquemas de codificación, bajo las hipótesis y parámetros establecidos a continuación:

- **Modelo de canal:** En principio, el modelo de canal más sencillo y adecuado es el AWGN, debido a que representa bien las fuentes de ruido térmico del receptor y del medio de transmisión<sup>1</sup>, considera una complejidad de decodificación ilimitada, y es el canal estándar para la comparación de distintos esquemas de codificación [21]. Sin embargo, como es mencionado en la Sección 2.1.4, la cuantización de la salida del demodulador permite representar el modelo AWGN como un canal discreto DMC, con un número de salidas equivalentes al número de estados del cuantizador. Un *cuantizador duro*<sup>2</sup> tiene sólo dos salidas, que equivalen a los dígitos binarios '0' y '1'. Este modelo de canal fue expuesto en la Definición 2.1.1 (canal BSC). Luego, es válido y conveniente representar el modelo AWGN mediante un BSC, eliminando la complejidad involucrada al simular esquemas de modulación y la transmisión en AWGN. Por lo tanto, se opta por simular el desempeño de los esquemas de codificación transmitiendo las secuencias por el canal BSC.
- **Modulación:** Del item previo, se deduce que es necesario seleccionar esquemas de modulación, pues las probabilidades de error del canal BSC asociadas a ellos corresponden a las probabilidades de error de detección. Por su aplicación extendida

---

<sup>1</sup>La densidad de electrones libres e iones en la ionosfera afecta la propagación de ondas de radio, y puede considerarse como fuente de ruido térmico adicional, si bien su potencia es poco apreciable en frecuencias UHF [29].

<sup>2</sup>Del inglés *hard quantizer*.

en la comunicación de pequeños satélites, el esquema BFSK con demodulación no-coherente es escogido; se selecciona además el esquema BPSK, debido a sus buenas cualidades de eficiencia en potencia y ancho de banda [64]. La elección de dos esquemas también permite estimar su influencia en el desempeño de los códigos y compararlos entre sí. Por otra parte, el esquema MSK, cuyas probabilidades de error de demodulación son idénticas al caso de BPSK según lo descrito en 2.2.2, está representado en los resultados de este último esquema.

- **Indicadores de desempeño:** Se fija como parámetro el rango de  $E_b/N_0$ , y en función del esquema de modulación simulado, quedan fijas las probabilidades de error de símbolos, al depender directamente del valor de  $E_b/N_0$ . De esta manera la tasa de errores BER es la variable dependiente del esquema de codificación. En particular, el rango simulado de  $E_b/N_0$  es [0,10] [dB] para el esquema BPSK, y [0,15] [dB] para BFSK. La diferencia de rangos se justifica pues las potencias requeridas por BPSK para obtener las tasas de error de interés (señaladas más adelante) se sitúan en el rango escogido; el mismo razonamiento es aplicado en el caso de BFSK. Cabe señalar que al fijar niveles de  $E_b/N_0$  se asume que tanto la modulación banda-base como la tasa de los símbolos es la misma para todas las simulaciones. En los análisis realizados posteriormente, a modo de criterio, se selecciona un BER de  $10^{-3}$  como valor mínimo para una transmisión confiable, y como máximo de interés un BER de  $10^{-5}$ . Estas tasas se seleccionan de acuerdo a los valores empleados por los fabricantes de dispositivos de comunicaciones; en particular, para el transpondedor empleado en el proyecto SUCHAI se usa  $BER=10^{-3}$ .
- **Esquemas de codificación:** Se simulan el esquema BCH, debido a su utilización en el dispositivo de comunicaciones del satélite en proyecto; Reed-Solomon, por su amplio uso en comunicaciones satelitales y reconocida capacidad de corrección; y convolucional por las mismas razones anteriores [18].
- **Número de realizaciones:** Es la cantidad de usos que se da al canal, a través del envío de una secuencia de datos, para obtener estadísticos con los cuales analizar el comportamiento de cada esquema de codificación. Depende de la tasa de errores mínima esperada; por ejemplo, si se impone un mínimo de 1 error por cada  $10^3$  bits, es conveniente simular al menos  $10^3$  realizaciones, para maximizar la probabilidad de que se obtenga dicha tasa de error. En este caso, se considera un BER mínimo esperado de  $10^{-6}$ , por lo tanto se simulan  $10^6$  realizaciones.

Otros posibles parámetros quedan definidos según el esquema particular de codificación a simular.

Las simulaciones se realizan del siguiente modo: primero se simulan usos de canal con tramas de bits sin codificación, moduladas en BPSK y BFSK, y se contrastan con las curvas teóricas para verificar el ajuste entre teoría y simulación. Luego, se evalúan los códigos seleccionados y descritos en detalle en las secciones respectivas para ambos tipos de modulación escogidos. Con los datos obtenidos de estas simulaciones se grafican curvas de desempeño con las tasas de errores promedio, cuantiles 0.1, 0.9 y 0.999 (especificados en los gráficos como cota 10 %, 90 % y 99.9 % respectivamente), y el valor máximo, obtenidas para

cada nivel de  $E_b/N_0$ . La decisión de graficar estos cuantiles y la tasa máxima se justifica por la variabilidad desconocida de las tasas de error; sus valores numéricos se escogen por criterios de representatividad y visualización gráfica. Los resultados obtenidos para cada esquema se discuten tras exponer los gráficos en cada caso. En algunos casos, ciertas curvas de cuantiles terminan abruptamente; esto ocurre pues los puntos sucesivos tienen BER nulo, no visible en los gráficos debido a la escala logarítmica del eje vertical.

### 3.1.2. Simulación sin codificación

Es necesario tener una referencia con la cual contrastar todos los esquemas de codificación a evaluar. La comparación natural corresponde a realizar la misma simulación sin corrección de errores, transmitiendo directamente los datos generados al canal BSC. Esta prueba no codificada se contrasta con cada código lineal evaluado, para visualizar la mejora añadida por este último. La figura 3.1 muestra el desempeño de una simulación que envía una trama de bits de largo  $L = 1000$ , a través del canal binario con las probabilidades de error de símbolos señaladas en la sección 2.3.2, para los casos de BPSK, BFSK coherente y BFSK no coherente. Este procedimiento se realiza  $n = 10^4$  veces para cada esquema. Las curvas son contrastadas con las curvas teóricas que **MatLab**® incluye en su Toolbox de Comunicaciones. De esta manera se comprueba que el uso del canal BSC entrega resultados correspondientes a la forma teórica esperada de las curvas.

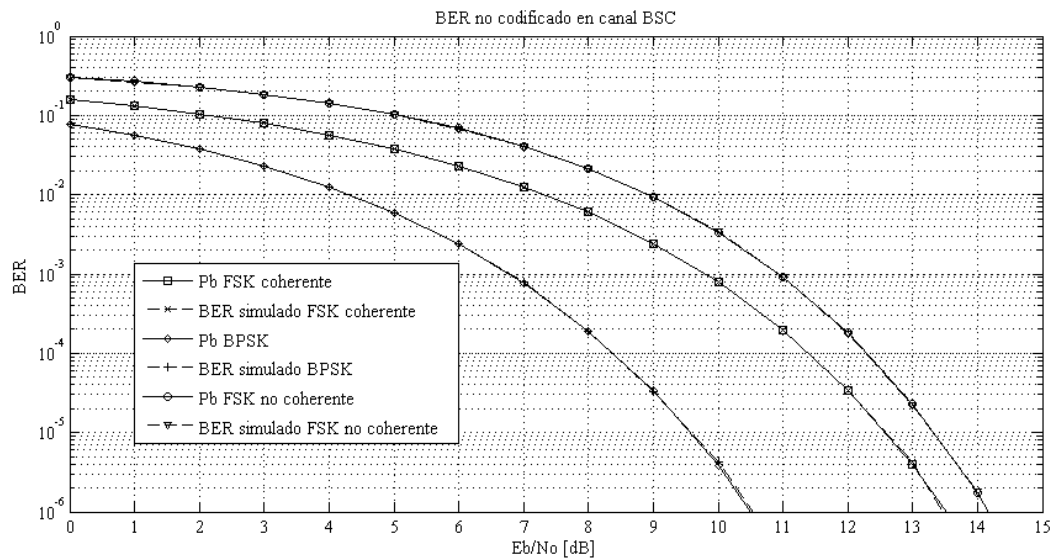


Figura 3.1: BER sin codificación, casos teórico y simulación en canal BSC, para modulación en BPSK, BFSK demodulado en forma coherente y no coherente.

### 3.1.3. Discusión de caso sin codificación

Se observa que existe un buen ajuste entre los valores teóricos y simulados del BER para ambos esquemas de modulación, con una diferencia máxima de  $\Delta p \approx 10^{-7}$ . Es decir, la tasa promedio de errores al transmitir secuencias en un canal BSC converge rápidamente a las probabilidades de error teóricas dadas por los esquemas de modulación empleados.

Es importante también señalar las ganancias de la modulación BPSK con respecto a BFSK con detección no coherente<sup>3</sup>. De este modo se distinguen las ganancias dadas por modulación de las obtenidas por el uso de corrección de errores. Tales ganancias se muestran en la Tabla 3.1:

BER	$10^{-3}$	$10^{-4}$	$10^{-5}$
$\Delta G_M$ [dB]	4.20	3.92	3.78

Tabla 3.1: Ganancias de modulación BPSK respecto a BFSK no coherente, para tasas de error de  $10^{-3}$ ,  $10^{-4}$  y  $10^{-5}$ .

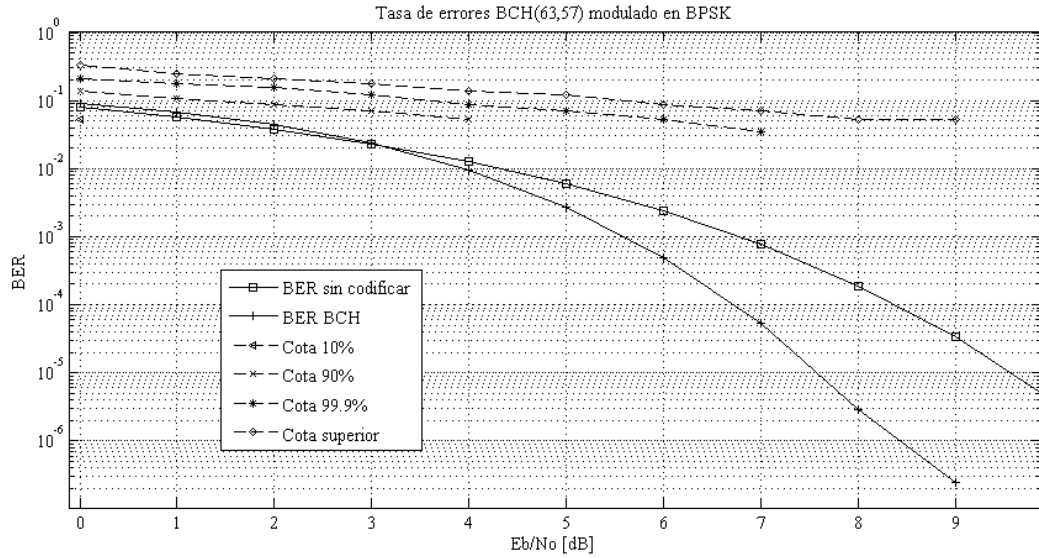
En las posteriores simulaciones la cantidad de realizaciones se incrementa a  $10^6$ , por lo que en lugar de contrastar transmisiones codificadas con envíos simultáneos de datos no codificados, se compara con la curva teórica dada por la modulación utilizada, para mayor simplicidad de implementación y ejecución computacional. Esto es válido dados los resultados expuestos previamente.

### 3.1.4. Simulación codificación BCH

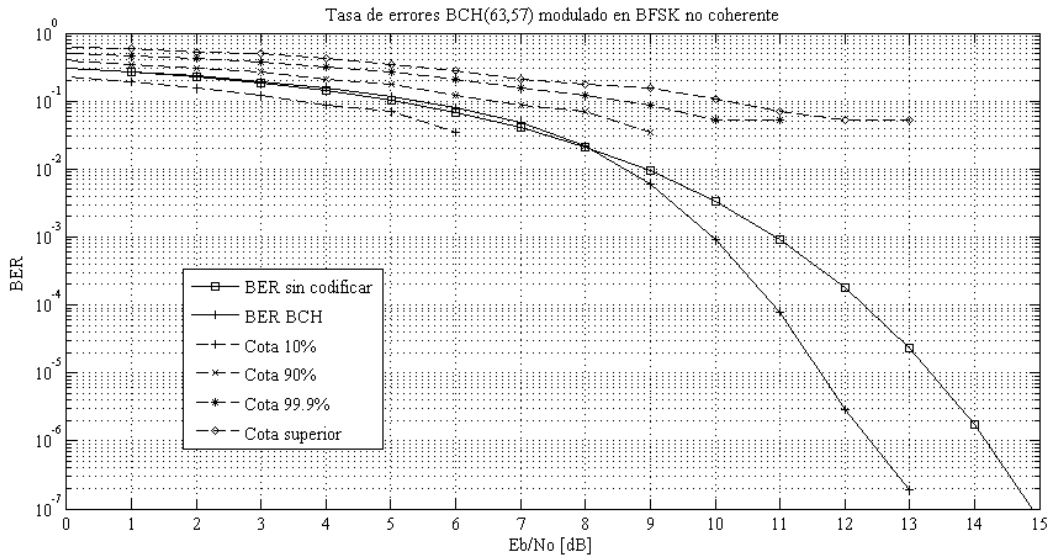
Se simula un código BCH con modulaciones BPSK y BFSK con demodulación no-coherente. El largo de las secuencias de datos es  $K = 57$  bits y el de las secuencias codificadas es  $N = 63$ . Por lo tanto, según la Tabla 2.2, la capacidad de corrección es de  $t = 1$  bit erróneo por secuencia codificada. El polinomio generador es  $g(x) = 1 + x + x^6$ . El método de decodificación usado es tabla de síndrome. Los resultados se muestran en la Figura 3.2. El número de realizaciones del canal es  $n = 10^6$ .

---

<sup>3</sup>El caso de BFSK con detección coherente no se abordará nuevamente, debido a su empleo infrecuente como ya se ha señalado.



(a)



(b)

Figura 3.2: Desempeño codificador BCH(63,57) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK. Nota: el gráfico (a) se ha desplazado levemente a la derecha para apreciar el único valor no nulo en la cota de 10% (BER=0.0526 para  $E_b/N_0 = 0$  [dB]).

### 3.1.5. Discusión codificación BCH

En el caso del esquema BCH, modulación BFSK (Figura 3.2b), se aprecia que sobre los 8 [dB] se mejora el BER promedio obtenido sin codificar, aunque con un valor de 0.02, cifra no confiable según el criterio escogido en 3.1.1. Para potencias desde 10 [dB], la tasa de errores es nula en un 90% de los casos, y un BER promedio de  $10^{-3}$  se obtiene para un nivel de

$E_b/N_0 \approx 10$  [dB], con una ganancia de código de  $\Delta G = 1$  [dB] respecto a la transmisión sin corrección de errores. Por otra parte, observando la cota de 90 %, se infiere que bajo 11 [dB] de potencia no se asegura que el código tenga mejor desempeño que el caso sin codificar, y que incluso puede tener peor comportamiento en cuanto a errores. Estos elementos presentan una desventaja si se desea emplear el esquema con niveles de potencia  $E_b/N_0$  menores a 10 [dB].

En cuanto a variabilidad de BER, ésta se amplía desde 6 [dB], con tasas de error cuyos valores pueden superar al esquema sin codificación, o por el contrario situarse en un BER nulo (decodificación exitosa), explicando los cortes abruptos en las curvas de cotas. Estos cortes se observan para todos los deciles desde 0.1 a 0.9, no todos expuestos en el gráfico 3.2b. De estos últimos se aprecia que para niveles de  $E_b/N_0$  entre 6 y 8 [dB] la tasa de error de los cuantiles comienza a caer aceleradamente, aunque una cantidad decreciente de palabras de código contienen un número de errores que supera la capacidad de corrección del esquema, y por lo tanto su BER asociado es elevado. Esta aparición de secuencias decodificadas incorrectamente se extiende hasta 13 [dB], y para potencias superiores el esquema corrige todas las secuencias transmitidas.

Conviene observar que los BER más altos (cotas 99.9 % y máxima), aunque pocos en comparación a la tasa promedio, pueden ser de órdenes de magnitud superior a esta última; por ejemplo, para un BER de  $10^{-3}$ , la tasa máxima observada es aproximadamente  $10^{-1}$ . Esta diferencia incrementa el valor promedio de la tasa de errores, y debe tenerse en cuenta al juzgar el desempeño del esquema de codificación debido a la distorsión que introduce. Esta situación de "polarización" de tasas de error se comprueba al observar sus histogramas (ver Apéndice A) y se repite en mayor o menor medida en los esquemas restantes, para ambas formas de modulación.

Se menciona en la sección 2.5.3 que el transceiver de SUCHAI utiliza este esquema modificado con un bit menos de mensaje. A pesar de implementar distintos polinomios generadores, las capacidades de corrección del código simulado y del transceiver son las mismas, por lo que es posible que la decodificación produzca curvas de BER similares a la obtenida, requiriendo los niveles de  $E_b/N_0$  antes mencionados para una corrección relativamente confiable. Aún así, el dispositivo cuenta con un protocolo adecuado que permite la detección de tramas defectuosas para su retransmisión en telecomandos (enlace de subida), utilizando códigos de redundancia cíclica [3].

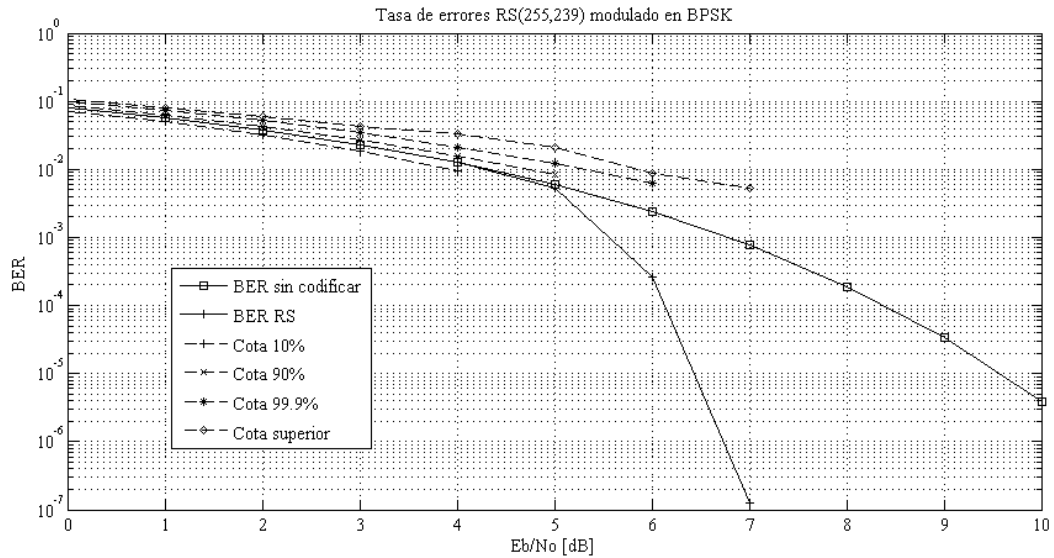
Para el caso de modulación BPSK, Figura 3.2a, los resultados son similares al caso de BFSK, con la diferencia que las caídas de curvas cuantiles se producen a menores niveles de  $E_b/N_0$ , debido a las menores probabilidades de error que el canal asigna para las mismas potencias. La mayor parte de las tasas cae aceleradamente en un rango acotado de potencia, en este caso entre 0 y 2 [dB], expresión del mismo fenómeno de "polarización" mencionado antes (nuevamente se han omitido varios cuantiles en el gráfico respectivo). Valores confiables de BER promedio se logran desde  $E_b/N_0 = 5.58$  [dB], con más del 90 % de secuencias decodificadas de forma correcta. Es posible entonces obtener un enlace con una tasa de error menor a  $10^{-3}$  con 6 [dB], y con una alta tasa de código ( $R = 0.9$ ). La tasa puede reducirse aún a valores bajo  $10^{-6}$  con alrededor de 9 [dB].

Contrastando resultados de ambos esquemas, en BPSK se obtienen tasas de error promedio para transmisión confiable a valores de  $E_b/N_0$  menores al caso de BFSK, con una ganancia de  $\Delta G = 1.19$  [dB] para un BER de  $10^{-3}$  en BPSK, superior en 0.19 [dB] a la ganancia señalada bajo la misma tasa en BFSK; para tasas de  $10^{-4}$  y  $10^{-5}$  la diferencia de ganancias no es superior a 0.27 [dB] (ver Tablas 3.4 y 3.5). Esta leve diferencia indica que la ganancia aportada por el codificador es muy poco dependiente de la modulación empleada, para las tasas de error analizadas.

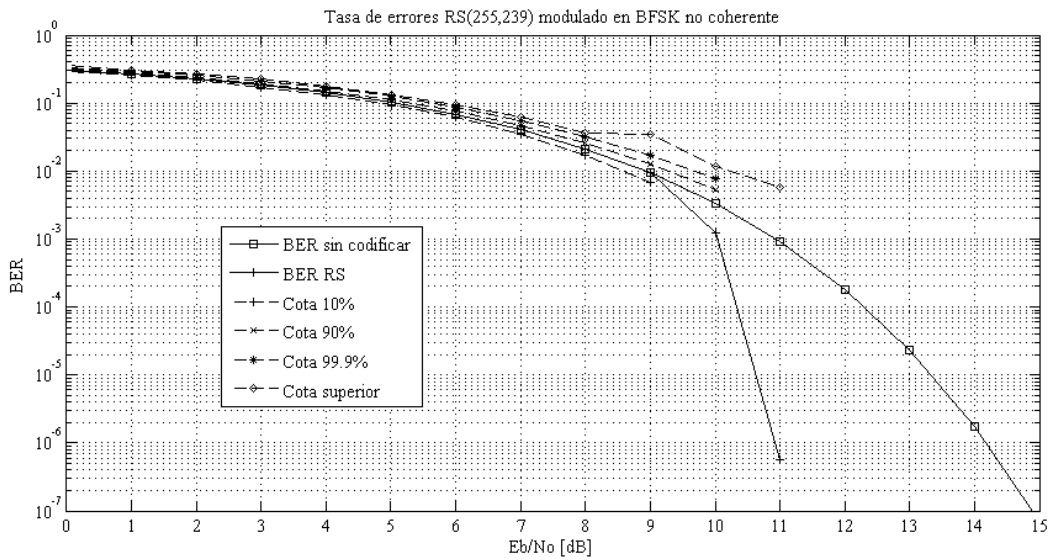
Se esperaría un mejor desempeño de este código al incrementar la capacidad de corrección, con el costo de menor transferencia de datos efectivos en la transmisión. Debe señalarse además que para niveles bajos de potencia ([0-8] [dB] en BFSK, [0-4] [dB] en BPSK) el promedio de errores en ambos casos es similar a los esquemas sin codificar, lo cual constituye el peor comportamiento esperable del código BCH.

### 3.1.6. Simulación codificación Reed-Solomon

Se simula un código Reed-Solomon (255,239), es decir la trama de datos es de  $k = 239$  símbolos y la trama codificada es de  $n = 255$  símbolos. Este esquema se presta para transmisión con más de dos símbolos en modulación, sin embargo en este caso se mantiene el esquema de modulación con dos símbolos, para lo cual el alfabeto de la trama codificada (de 256 símbolos) es convertido a una trama binaria de mayor longitud. La decodificación es realizada con el algoritmo Berlekamp-Massey. La Figura 3.3 grafica el desempeño de este esquema en ambas modulaciones:



(a)



(b)

Figura 3.3: Desempeño codificador RS(255,239) en canal binario simétrico.(a) Modulación BPSK, (b) Modulación BFSK.

### 3.1.7. Discusión codificación Reed-Solomon

En modulación BPSK, un comportamiento con escasa variabilidad se aprecia hasta 5 [dB], donde el desempeño es similar al caso sin codificación. Es distintivo de este caso particular la caída abrupta en el BER promedio a partir de 5 [dB], desde donde las tasas presentan mayores variaciones, produciéndose un evento similar de concentración de tramas decodificadas con BER=0 y algunas con BER superior a sin codificación. Para un BER de



$10^{-5}$ , con  $E_b/N_0 = 6.43$  [dB], la ganancia de código es de 3.13 [dB], superior a la alcanzada en ambas modulaciones con BCH, aunque a un mayor costo energético, o menor tasa de datos  $r_b$ . Se resalta la baja tasa que puede lograrse con un incremento moderado de potencia, pues desde un tasa de errores de  $10^{-3}$  con 5.56 [dB] se disminuye hasta  $10^{-7}$  con 7 [dB]. Es decir, con un aumento de 1.44 [dB] se reduce la tasa en 4 órdenes de magnitud.

Bajo modulación BFSK, el desempeño promedio del código es prácticamente idéntico al de una transmisión sin codificación, con pocas variaciones de modo análogo a BPSK en este esquema. Esta tendencia se mantiene hasta los 9 [dB], y para el rango de [10,11] [dB] se produce una fuerte disminución de la tasa de errores, cayendo desde  $10^{-3}$  hasta menos de  $10^{-6}$ . En el mismo intervalo se aprecia la alta cantidad de decodificación exitosa, pues sólo la cota máxima persiste. La decodificación es correcta para todas las secuencia transmitidas desde 11 [dB]. Las ganancias de código para este esquema pueden verse en la Tabla 3.5, siendo la mayor de ellas de 2.71 [dB] para un BER de  $10^{-5}$ .

Para ambos casos, la estrecha variación de las cotas en torno a la curva de BER promedio, cuando ésta es similar al caso no codificado, podría explicarse por la conversión de la trama codificada 256-aria a una secuencia más larga formada por símbolos binarios, para ajustar la transmisión por el modelo de canal fijado. Posiblemente la capacidad de corrección de este código se vea perjudicada pues la inversión de los símbolos en el canal inducirían una reconstrucción de la palabra 256-aria original menos confiable que la transmisión de los símbolos completos o con menos segmentación, a través de modulaciones  $M$ -PSK o  $M$ -FSK,  $M > 2$  para ambos casos. Recuérdese que la capacidad de este código es de  $t = 8$  símbolos 256-arios en una trama de largo  $n = 255$ , pero las secuencias enviadas son de largo  $n' = 2040$  bits y sus símbolos son binarios, por lo que las circunstancias necesarias para dicha corrección no están dadas *a priori*. Aún así, se piensa que un desempeño similar es esperable al emplearlo como parte de un esquema de comunicaciones satelitales, pues la manera usual de modular señales, en enlaces móviles, es O-QPSK [41, Cap. 4]. En este sentido, debería percibirse una mejora en el desempeño del esquema si la modulación se amplía a un esquema con más símbolos, al menos para el caso de  $M$ -FSK<sup>4</sup>.

Comparando las ganancias respecto a modulación no codificada, la máxima diferencia entre las obtenidas para BPSK y BFSK es de 0.42 [dB] bajo una tasa de errores de  $10^{-5}$ , por lo cual el desempeño dado por el esquema Reed-Solomon es prácticamente independiente de la modulación aplicada.

Por último, se rescata el hecho de que el peor comportamiento esperado para ambas modulaciones codificadas es similar a los casos sin codificar, y por lo tanto, el esquema Reed-Solomon ofrece un desempeño igual o mejor a este último, en el rango completo de potencias evaluadas ([0-15] [dB]).

---

<sup>4</sup>Para modulación PSK, el aumento de símbolos degrada el desempeño del esquema sin codificar [64, Cap. 4], aunque no se prevé el comportamiento exacto al emplearlo con codificación Reed-Solomon.

### 3.1.8. Simulación codificación convolucional

Se simula un código convolucional con las modulaciones usadas en los casos anteriores. El largo de la trama es de  $L = 64$  bytes, escogido de acuerdo al largo de los telecomandos del transceiver satelital en el proyecto SUCHAI. En la Figura 3.4 se muestra el diagrama circuital del codificador, extraído de la recomendación de la CCSDS [18]. Este esquema particular es en la actualidad parte de un estándar *de facto* en la industria de comunicaciones satelitales y móviles [14]. Para fines del análisis del desempeño del código convolucional, no se considera el inversor añadido en la salida de la trama  $c_2$ , cuya finalidad es sólo asegurar sincronización. La memoria del codificador es  $K = 7$ , la tasa de código es  $r = k/n = 1/2$ , y los vectores de conexiones son

$$(3.1) \quad G_1 = [1111001] \quad G_2 = [1011011]$$

donde un '1' en la posición  $i$ -ésima del vector es una conexión de la entrada del *delay*  $i$ -ésimo al sumador módulo-2, equivalente a un operador lógico XOR. Estos vectores se pueden interpretar como números binarios, y se abrevian en forma octal. De este modo una forma resumida de caracterizar las conexiones es  $G_1 = 171_8$ ,  $G_2 = 133_8$ . Esta notación es además la utilizada por *MatLab* para describir un código convolucional. Este codificador es no-sistemático. El esquema de decodificación empleado es el algoritmo de Viterbi. En el Capítulo 4 se exponen más detalles de este decodificador de características particulares.

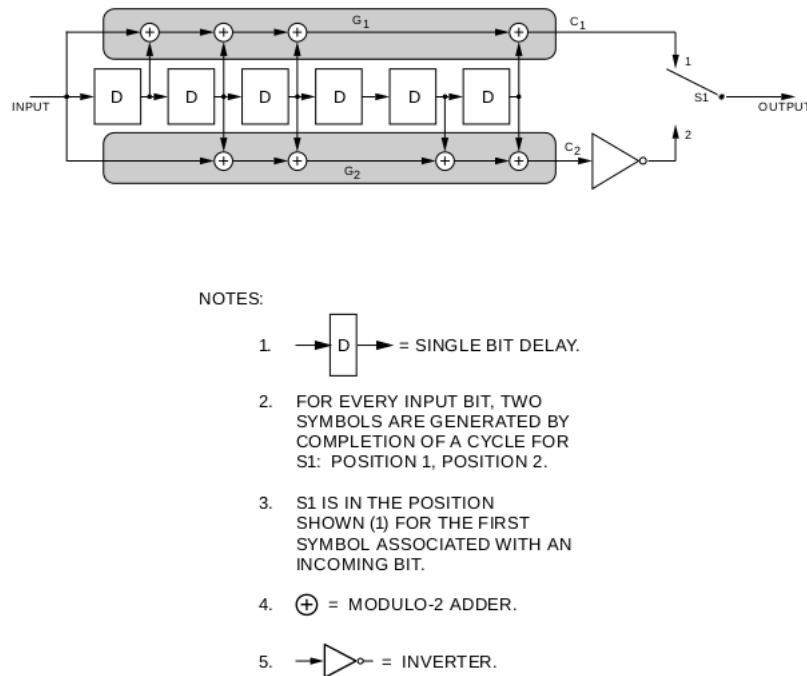
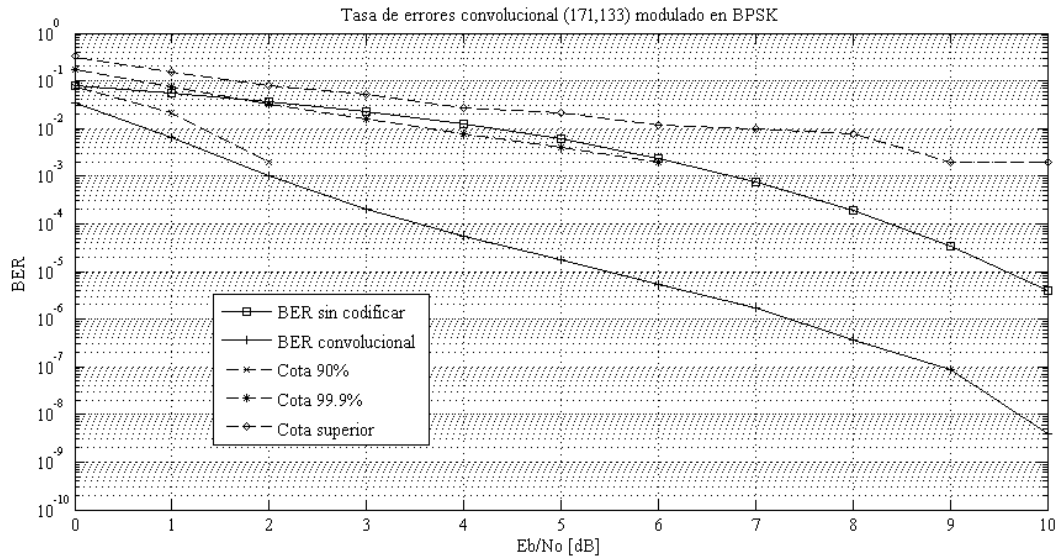
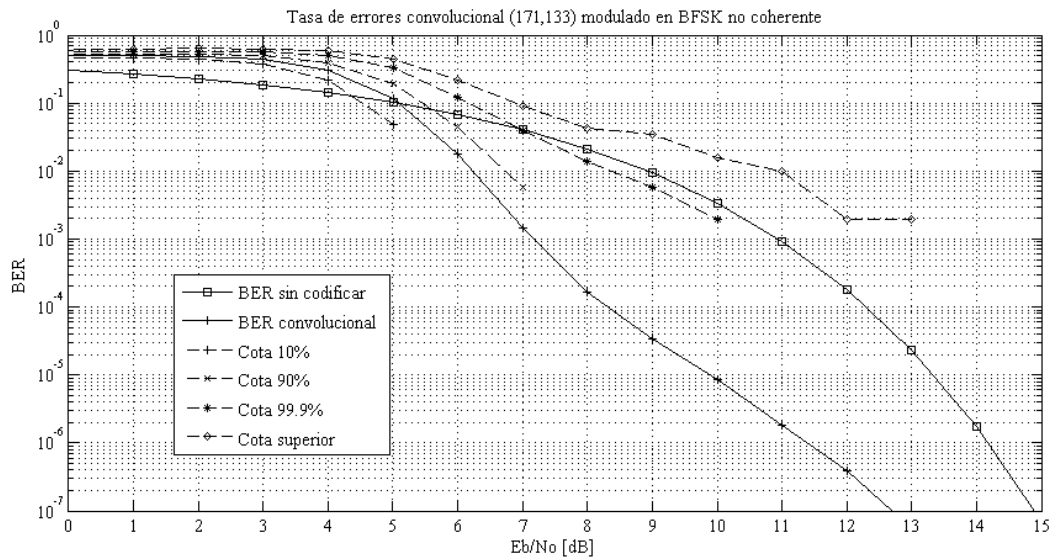


Figura 3.4: Diagrama del codificador convolucional (171,133). Figura extraída de [18].



(a)



(b)

Figura 3.5: Desempeño codificador convolucional (171,133) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK.

### 3.1.9. Discusión codificación convolucional

Para la simulación de código convolucional, el desempeño en modulación BFSK es pobre con un nivel de potencia bajo; se aprecia que prácticamente todos los casos, con seguridad el 90% como se infiere de la cota de 10%, son erróneamente decodificados bajo los 5 [dB]; en otras palabras, el desempeño promedio del esquema codificado empeora con respecto al caso no codificado. A potencias mayores, la tasa de errores promedio mejora al caso no

codificado; en un 90% de los casos el BER es nulo sobre los 7 [dB], pero para las cotas superiores la tasa de errores se mantiene alta en torno a las tasas sin codificación. Desde 7.2 [dB] de potencia puede aseverarse que el código tiene un comportamiento confiable, pues en promedio las tasas de error son inferiores al límite de confiabilidad establecido, y en un 99.9% el desempeño mejora al caso sin codificación en cuanto a BER. La ganancia de código con respecto a BFSK sin corrección de errores es  $\Delta G_{BFSK} = 3.8$  [dB] para una tasa de  $10^{-3}$  (Tabla 3.5).

En modulación BPSK, para un  $E_b/N_0$  sobre 2 [dB] la tasa de errores es nula en el 90% de los casos, y sobre 6 [dB] es nula en el 99.9% de los casos. Además el nivel de potencia desde el cual el código tiene un comportamiento promedio confiable ( $BER=10^{-3}$ ) es de 2 [dB], con una ganancia de código de  $\Delta G_{BPSK} = 4.76$  [dB]. Debe notarse sin embargo que existirán tramas con alto contenido de error en el rango completo de potencias, como se aprecia en la curva de la cota máxima, conducta observada en BFSK también. De modo similar a los esquemas previos de codificación, la cantidad de estas tramas es muy inferior a aquellas decodificadas exitosamente, aún cuando su tasa de errores puede superar por órdenes de magnitud al promedio (para  $E_b/N_0 \approx 7.3$  [dB], la tasa promedio es  $10^{-6}$  y la máxima es  $\approx 10^{-2}$ ).

Es preciso tener en cuenta que el decodificador convolucional con algoritmo de Viterbi, a diferencia de los decodificadores de los esquemas previos, corrige la trama completa en lugar de bit a bit, escogiendo la secuencia más probable. Esto significa que, cuando los errores producidos en la trama no superan la capacidad de corrección del código, la secuencia es decodificada correctamente siendo la más probable; en el caso que la capacidad es superada, el decodificador interpreta erróneamente una secuencia como la más probable, decodificando incorrectamente. Este algoritmo puede entonces contribuir con algunas tramas mal decodificadas, las que eventualmente tendrán un BER mayor al de la transmisión sin codificar.

Finalmente, la máxima diferencia entre las ganancias obtenidas para BPSK y BFSK se sitúa para una tasa de errores de  $10^{-3}$ , y es de 0.96 [dB], siendo entonces de moderada incidencia la modulación aplicada al mensaje codificado.

### 3.1.10. Comparación entre esquemas

Expuestos los resultados individuales de desempeño de los esquemas revisados, una comparación gráfica da cuenta del desempeño relativo entre ellos, para ambas modulaciones seleccionadas.

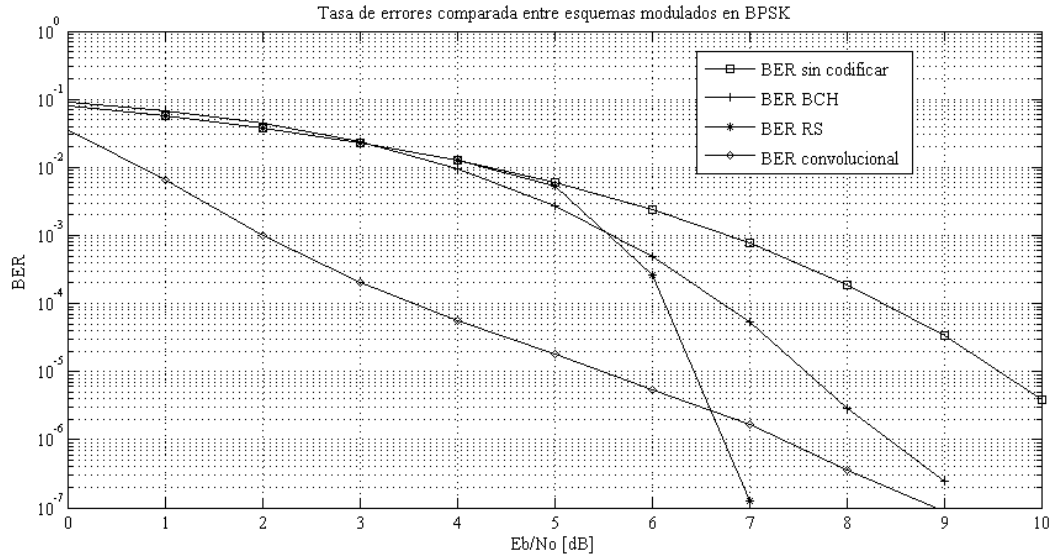


Figura 3.6: Desempeño de códigos BCH(63,57), RS(255,239) y Convolutacional(171,133) en canal BSC con modulación BPSK.

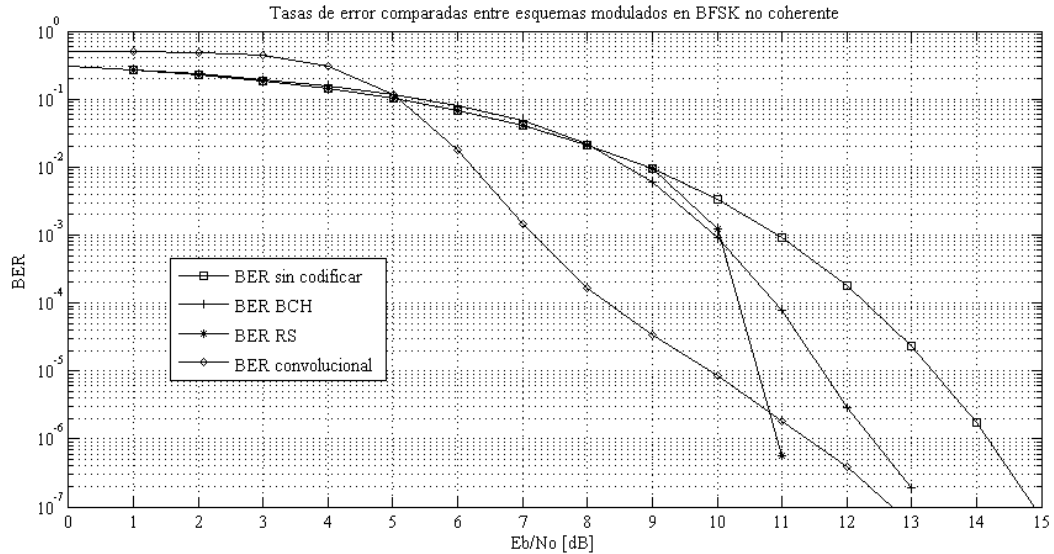


Figura 3.7: Desempeño de códigos BCH(63,57), RS(255,239) y Convolutacional(171,133) en canal BSC, modulación BFSK con detección no coherente.

### 3.1.11. Discusión comparativa entre codificaciones

De las curvas obtenidas mediante simulación, se aprecia que el desempeño de los esquemas revisados disminuye la tasa de errores para ambas modulaciones empleadas. Las Tablas 3.2 y 3.3 muestran la potencia necesaria para cada esquema de codificación, comparada

con la transmisión no codificada, para BPSK y BFSK.

Cabe señalar que para un código con menor capacidad de corrección es esperable una menor mejora en desempeño, como se observa en el caso de BCH. Sin embargo, debe considerarse que las secuencias son de distinta longitud en bits para cada caso; de este modo, el esquema BCH puede corregir 4 errores separados en 4 secuencias, de largo  $N_{BCH} = 63$  bits cada una, y el código Reed-Solomon corrige 8 símbolos en una trama de longitud  $N_{RS} = 255$ . Luego, si se codifican 4 secuencias de 63 bits en BCH y se transmiten como un único tren de largo  $63 \times 4 = 252$  bits, la capacidad de corrección de BCH es de  $t = 4$ , suponiendo que en cada sub-secuencia de largo  $N_{BCH} = 63$  bits sólo se produce 1 error. Esta apreciación debe tenerse en cuenta al comparar entre sí los esquemas de codificación.

Modulación PSK	Bit Error Rate		
Codificación	$10^{-3}$	$10^{-4}$	$10^{-5}$
Sin codificación	6.77	8.37	9.56
BCH(63,57)	5.58	6.72	7.58
Reed-Solomon(255,239)	5.56	6.14	6.43
Convolutional(171,133)	2.01	3.54	5.55

Tabla 3.2: Comparación de potencias requeridas para esquemas de codificación con modulación BPSK, en [dB].

Modulación FSK	Bit Error Rate		
Codificación	$10^{-3}$	$10^{-4}$	$10^{-5}$
Sin codificación	10.97	12.29	13.34
BCH(63,57)	9.97	10.90	11.63
Reed-Solomon(255,239)	10.04	10.34	10.63
Convolutional(171,133)	7.17	8.32	9.87

Tabla 3.3: Comparación de potencias requeridas para esquemas de codificación con modulación BFSK, en [dB].

## Modulación BPSK

En el análisis del gráfico 3.6 pueden considerarse dos criterios para la elección de un esquema superior en desempeño. En primer lugar puede ser deseable aquel sistema de codificación que permita obtener el BER más bajo con potencia aceptable, sin ser esta última la limitación principal, en cuyo caso el esquema RS(255,239) supera al esquema convolutional para  $BER \leq 10^{-7}$ , requiriendo un máximo de 7 [dB], mientras que el convolutional necesita más de 9 [dB]. Sin embargo en el rango de interés, con BER situado entre  $10^{-3}$  y  $10^{-5}$ , el esquema convolutional requiere como mínimo 1 [dB] menos para obtener dichas tasas de error. El otro criterio es escoger el código con BER aceptable, exigiendo el uso mínimo de energía como condición principal. En tal caso, el esquema convolutional logra BER inferiores a  $10^{-5}$  en un rango de potencia entre 2-6 [dB], superando con facilidad a los otros esquemas.

Consecuentemente, bajo dos criterios diferentes, tomando en cuenta sólo el rango de BER de interés, el esquema convolucional supera en desempeño a los esquemas Reed-Solomon y BCH. Se debe tomar en cuenta además que el código convolucional supera en todo el rango de potencia el desempeño de la modulación no codificada, mientras en los otros casos el desempeño es similar a esta última para potencias más bajas. Además, de las discusiones de los esquemas es directo que el código convolucional tiene ganancia de código superior a los demás esquemas; esto se expone en la Tabla 3.4.

Bit Error Rate	$10^{-3}$	$10^{-4}$	$10^{-5}$
$\Delta G$ BCH (63,57)	1.19	1.65	1.98
$\Delta G$ Reed-Solomon (255,239)	1.21	2.23	3.13
$\Delta G$ Convolucional (171,133)	4.76	4.83	4.01

Tabla 3.4: Ganancias de código con modulación BPSK, en [dB].

## Modulación BFSK

Bajo una óptica similar al caso de BPSK, pueden considerarse dos criterios de decisión para el código con desempeño superior en el caso de modulación BFSK, gráfico 3.7. Bajo el criterio de menor BER, el esquema convolucional se muestra como opción indiscutible, pues alcanza los BER de interés en el rango de potencia de 7-10 [dB]. Si se define una potencia reducida a aquella comprendida entre [0-7] [dB] y se exige que la corrección de errores tenga un desempeño confiable en dicho rango, entonces ninguno de los esquemas satisface este requerimiento. Por lo tanto el código convolucional sólo cumple con un tipo de criterio en cuanto a desempeño. Sin embargo, este esquema presenta la desventaja de que para el rango de potencia entre 0-5 [dB], su desempeño promedio empeora respecto al caso sin codificación, y no parece útil emplearlo bajo condiciones de potencias bajas, como las señaladas. Por su parte, el esquema Reed-Solomon puede adquirir tasas de error bastante bajas, aunque a un costo energético más alto. Por último, el código BCH requiere potencia similar a Reed-Solomon para obtener un BER de  $10^{-3}$ , pero requiere mayores potencias para obtener tasas más bajas en contraste a dicho esquema, y por lo tanto es la alternativa menos conveniente en todos los términos para modulación en BFSK. De todos modos debe recordarse que su desempeño puede mejorarse mediante la disminución de la tasa de código.

La Tabla 3.5 resume las ganancias de la modulación BFSK codificada en contraste al mismo esquema sin corrección de errores.

Bit Error Rate	$10^{-3}$	$10^{-4}$	$10^{-5}$
$\Delta G$ BCH (63,57)	1.00	1.39	1.71
$\Delta G$ Reed-Solomon (255,239)	0.93	1.95	2.71
$\Delta G$ Convolucional (171,133)	3.80	3.97	3.47

Tabla 3.5: Ganancias de código con modulación BFSK, en [dB].

## Discusión adicional

De los gráficos expuestos a lo largo de este capítulo, es visible la similitud de las curvas de modulación BFSK y BPSK para cada esquema de codificación particular, aunque trasladadas en potencia. Esto se aprecia de igual manera en las curvas sin codificación. Esto es consecuencia de que, en el canal BSC, la única diferencia entre las modulaciones FSK y PSK binarias son las probabilidades de error asignadas para los mismos niveles de  $E_b/N_0$ . Para el rango simulado de potencia, las probabilidades de error del canal se sitúan en el intervalo  $[0.00337, 0.303]$  para BFSK, y en el caso de BPSK están en el intervalo  $[0.0000039, 0.0786]$ . Se ve entonces que las probabilidades de error asociadas a BPSK son por lo menos un orden de magnitud menores que las de BFSK, lo que justifica el peor desempeño de este esquema con respecto al primero. Se aprecia además que los códigos mejoran levemente más la modulación en fase, medido en las diferencias entre ganancias de código, siendo el convolucional donde más notorio es este efecto, con una diferencia de casi 1 [dB] entre ganancias de BPSK y BFSK codificados.

Con respecto al desempeño de los esquemas evaluados, hay varios aspectos que inciden en ello. El enfoque del código convolucional, de estimar el mensaje de forma probabilística y en la trama completa, a diferencia de los dos restantes enfocados en la detección y corrección de errores bit a bit, puede ser una causa de la notoria mejora del comportamiento del esquema en todo el rango de potencia para BPSK, y sobre los 5 [dB] en BFSK. Como desventaja, esta superioridad tiene el costo no despreciable de una baja tasa de código  $R_{conv} = 0.5$ , en contraste a los esquemas BCH y Reed-Solomon (ver Tabla 3.6), además de requerir incrementos mayores de potencia para disminuir el BER. La tasa de código de un convolucional puede incrementarse mediante un *perforador*<sup>5</sup> que elimina algunos símbolos de salida del codificador, intentando recuperarlos en el decodificador. El desempeño del esquema con perforación empeora en cuanto a BER, aunque puede mantenerse mejor que los otros esquemas<sup>6</sup>.

Un aspecto no menor lo constituye la complejidad de implementación, tanto del codificador como del decodificador. Para los códigos BCH y convolucional, el codificador es de baja complejidad, pues son circuitos secuenciales con pocos elementos, además de la ventaja que las operaciones de suma y multiplicación en  $GF(2)$  no generan *carries*, e incluso sólo pueden requerir circuitos combinatoriales con operaciones XOR y AND. El caso de Reed-Solomon es distinto al tener un alfabeto no-binario; se requiere entonces diseñar circuitos con registros que realicen las operaciones en el conjunto no-binario definido, en este caso de 256 símbolos [13]. Es posible aumentar la constelación de símbolos en el modulador, aunque en el caso de FSK implica el uso de mayor ancho de banda, y en PSK aumenta la tasa de errores [64, Caps. 3,4].

En la decodificación, todos los códigos incrementan su complejidad, de manera especial el convolucional, cuyo circuito requiere una cantidad exponencialmente creciente de *flip-flops* con el incremento lineal de *delays* en el codificador. El decodificador de Reed-Solomon, que

---

<sup>5</sup>Del inglés *puncturer*

<sup>6</sup>Se ha evaluado que mediante perforación, aumentando la tasa de código a  $R = 0.75$ , el esquema mantiene tasas más bajas que empleando BCH(63,57) [16].



puede ser el mismo de BCH extendido al alfabeto no-binario, consiste de varias etapas cuya implementación puede variar en complejidad, existiendo un compromiso entre rapidez, costo y finalidad de aplicación, dado que puede llevarse a cabo por hardware o software [13, Cap. 6].

Finalmente, la decodificación de síndrome es la más sencilla, pues requiere una única etapa de *buffer* para la recepción del mensaje codificado, seguido de un circuito combinacional de cálculo de síndrome y detección-corrección de error. Este algoritmo es sólo recomendado para códigos con bajo valor de  $n-k$ , por lo cual es una alternativa válida para el decodificador del transceiver satelital [13].

	BCH	RS	convolucional
$R = k/n$	0.905	0.937	0.500

Tabla 3.6: Tasas de código para esquemas de codificación.

## 3.2. Conclusión

Se comprueba que los esquemas de codificación tienen comportamientos individuales particulares, justificando que cada cual se aplique a diferentes situaciones, según el nivel de potencia manejado o nivel de errores máximo requerido. Sin duda, en la aplicación que concierne a este trabajo, la exigencia puede provenir de ambos requisitos, puesto que el satélite maneja una potencia limitada, y se exige una tasa de errores muy baja en las tramas del enlace de subida (*uplink*) puesto que ellas constituyen los telecomandos. Del análisis previamente realizado, se infiere que la mejor implementación consiste en un código convolucional con modulación BPSK, puesto que requiere baja potencia para obtener tasas de error bastante bajas, y la complejidad del codificador es la más baja comparado a los otros códigos. Sin embargo, debido a la complejidad del decodificador, en principio su diseño estaría sólo orientado a funcionar en tierra, con lo cual el código se aplicaría sólo para enlaces de bajada (*downlink*), como control de errores para telemetría. Aún así, se ha mencionado en la sección 2.7.2 el esfuerzo existente por reducir la complejidad de este algoritmo, por lo que no se descarta la posibilidad de la inclusión del decodificador en el satélite, para corrección de errores en *uplink*. Por otra parte, sólo se ha analizado demodulación con *hard decision*; la implementación de detección con más niveles, denominada *soft decision*, aumenta la ganancia de código con respecto a la detección binaria revisada [11, Cap. 6].

Una alternativa más compleja, pero que puede ofrecer tasas más reducidas, sería la implementación de un esquema combinado de convolucional y Reed-Solomon; para niveles de potencia baja, cuando el enlace es distante, puede usarse sólo el esquema convolucional, y para mayores potencias, cuando el enlace es más próximo a la estación terrestre, usar un esquema Reed-Solomon. Esto requeriría un control de potencia que decida cuándo conmutar el codificador, en base a la calidad del enlace. La complejidad de este sistema debe contraponerse con la necesidad efectiva de implementarlo, pues los recursos energéticos y el tiempo de enlace son limitados para el satélite, y además puede ocurrir que la solución más simple propuesta

funcione adecuadamente.

También está la implementación del modulador y demodulador de BPSK; este último requiere que la señal de referencia del receptor se sincronice en fase y frecuencia con la señal recibida. Variados esquemas existen para construir un transpondedor de BPSK que satisfaga estos requerimientos [64, Cap. 4]. Debe tenerse en cuenta que las señales recibidas están alteradas por efecto Doppler, y sus frecuencias están desplazadas; sin embargo, en un enlace de línea vista esto no debería afectar la diferencia relativa de fases entre dos ráfagas transmitidas. La sensibilidad a la diferencia de frecuencias del receptor, y el ancho de banda aceptable en los filtros del mismo podrían influir en su desempeño efectivo frente al mencionado efecto.

Fue mencionado en 2.2.2 que el desempeño teórico del esquema BPSK es idéntico al de MSK. Este último esquema presenta características propias de FSK además, y puede ser demodulado de forma no coherente, con poca degradación. Dados los resultados favorables para BPSK, es de considerar la implementación de MSK como alternativa. En este trabajo no se ha considerado la evaluación del esquema de modulación más adecuado para el transpondedor, por lo que queda como un problema abierto.

En cuanto a la tasa de código, a pesar de ser la menor entre los tres códigos analizados, las bajas tasas de error que ofrece pueden reducir la necesidad de retransmitir datos, compensando la cantidad limitada de información portada. Puede realizarse una estimación de cuánta información efectiva se realiza por transmisión, tomando como base el protocolo de comunicación empleado por el transpondedor de la misión SUCHAI: cada *frame* de telemetría consta de 128 bytes codificados, sumados con 12 bytes adicionales de cabeceras y bits de control; esto suma 140 bytes. Si se transmite a 1200 [bps], suponiendo un enlace en línea vista de 3 minutos efectivos de comunicación<sup>7</sup>, la cantidad de bytes enviados total es de  $3 \times 60 \times 1200 = 27$  [kB], equivalentes a 192 *frames* de telemetría, con  $192 \times 128 = 24576$  bits codificados, de donde se concluye que la información efectiva es de  $24576 \times (R = 0.5) = 12288$  bits, unos 1.5 [kB]. Este cálculo toma en cuenta que parte del tiempo de enlace es requerido para tomar contacto con el satélite, y otra parte donde se envían telecomandos, dado que el transpondedor actual opera en un único sentido de comunicación a la vez (*half-duplex*). Si este enlace se realiza bajo condiciones tales que  $E_b/N_0 \approx 3$  [dB], el promedio de errores, empleando modulación BPSK codificado con código convolucional, es de  $2 \times 10^{-4}$ , equivalente a 43 bits erróneos del total transmitido, mientras que en BPSK sin codificar dicho promedio es de  $2.5 \times 10^{-2}$ , equivalente a 5400 bits errados. Con 43 bits erróneos la necesidad de retransmisión es bastante más reducida que con 5400<sup>8</sup>; entonces, bajo las condiciones supuestas y conclusiones obtenidas, el esquema convolucional disminuye notoriamente los errores introducidos por el canal.

---

<sup>7</sup>Un enlace en línea vista tiene una duración aproximada de 8-10 minutos, y menor en casos donde el enlace es distante y/o obstaculizado por montañas o edificaciones.

<sup>8</sup>En el peor de los casos, la cantidad de bits erróneos bajo codificación convolucional es similar al promedio de BPSK no codificado.

# Capítulo 4

## Implementación en FPGA

### 4.1. Especificaciones de diseño

#### 4.1.1. Codificador convolucional

El codificador convolucional implementado es esencialmente el mismo utilizado para las simulaciones, proveniente del estándar de la CCSDS [18]. El esquema circuital es el mostrado en la Figura 3.4 de la sección 3.1.8. El inversor es añadido por las razones especificadas en el estándar, en cuanto a asegurar las transiciones de bit que mantengan en cerradura (*in lock*) los sincronizadores de símbolos en el receptor. Las especificaciones entonces son:

- Tasa de código:  $k/n = 1/2$  por cada bit entrante se producen dos bits de salida.
- Número de delays:  $K = 7$
- Vectores de conexiones para sumadores lógicos:  $G_1 = [1111001]$ ,  $G_2 = [1011011]$
- Adición de inversor en salida de bloque secuencial  $G_2$  y multiplexor de dos entradas.
- Distancia mínima:  $d_{min} = 10$ , por lo tanto la capacidad de corrección es de  $t = 4$ .

Una forma útil de describir el código convolucional es mediante los enmallados o *trellis*, como se menciona en 2.5.3. Si un codificador cuenta con  $K$  unidades de memoria, cada uno de ellos tiene un estado **0** si no está almacenando un bit, y un estado **1** en caso contrario. Para las  $K$  unidades, los estados individuales se agrupan en un único vector de estado  $\mathbf{S} = [S_0 S_1 \dots S_K]$ , de donde se desprende que hay  $2^K$  estados posibles, 128 en este caso. Cada estado conlleva asociada la entrada que lo origina y su salida correspondiente, y en el enmallado queda predeterminado qué estado conduce a los siguientes; en virtud de esto se lleva a cabo la decodificación en el receptor. El mensaje de largo  $m$ , al ser codificado será de largo  $n(m + K) = 2(m + 7)$  pues cada *delay* añade una secuencia más de largo  $n$  después de que la entrada recibe el último bit no-nulo.

### 4.1.2. Decodificador convolucional

#### Algoritmo de Viterbi

El decodificador convolucional implementado aplica el algoritmo de Viterbi para la estimación del mensaje. Dicho algoritmo corresponde al método que maximiza la verosimilitud del mensaje en la salida del decodificador [13], y casi siempre toma una decisión de mensaje en lugar de declarar errores de decodificación, razón por la que se denomina *decodificador completo* [9].

Al recibir la secuencia binaria, el decodificador toma la primera secuencia de largo  $n$ , y con el enmallado determinado por la forma del codificador, la compara con la salida asociada a cada uno de los  $2^{128}$  estados posibles. Para esta comparación se requiere definir una métrica, llamada *métrica de rama*, siendo las usuales la distancia de Hamming definida en 2.5.2 o la distancia euclidiana [11]. El resultado de este contraste indicará que hay algunas salidas con menor distancia que otras con respecto a la secuencia comparada; en tal caso los estados asociados a dichas salidas están ligados a estados posteriores más probables. Las trayectorias con distancia mínima se denominan *trayectorias supervivientes* y son almacenadas. Aquellas con distancia mayor no se reservan. Tras esto, la siguiente secuencia de entrada de largo  $n$  es igualmente comparada. En esta etapa, además de calcular la distancia propia, se calcula una *métrica de trayectoria*, correspondiente a la suma de la distancia previa y la actual. Este procedimiento continúa hasta completar las comparaciones, es decir cuando se termina la palabra de código. En todo el procedimiento descrito, se asume que la secuencia recibida comienza a partir del estado  $S_0$ , siendo el punto de arranque del algoritmo. Finalmente, el proceso de *traceback* reconstruye el trazado completo para la trayectoria con menor distancia, a partir de la cual obtiene el mensaje original. El último instante de tiempo, graficado como  $t_b$ , corresponde al instante de largo máximo a decodificar, que como se dijo en 2.5.3, es suficiente para este código. Aumentar ese tiempo máximo significa incrementar el *delay* producido y la capacidad de memoria, con escasa mejora en el desempeño [17]. En la Figura 4.1 se señalan en línea verde las trayectorias supervivientes, y en rojo la trayectoria final, sólo a modo de graficar lo expuesto.

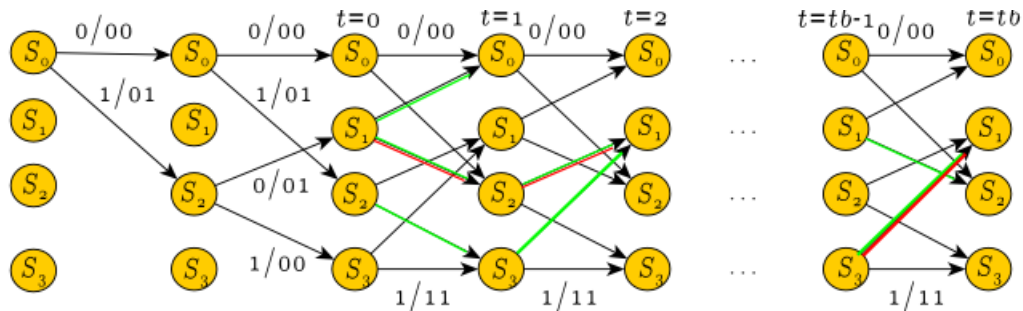


Figura 4.1: Enmallado de decodificación para un codificador con  $K = 2$  unidades de memoria.

## Diseño

El algoritmo de Viterbi se implementa en tres bloques básicos, como se ilustra en la Figura 4.2. El primero de ellos es el bloque de cálculo de métrica de rama o BMU (*Branch Metric Unit*), que realiza esta operación con las métricas mencionadas en el apartado anterior; si aplica distancia de Hamming, se dirá que el decodificador realiza *hard decision*, y en caso de usar métrica euclidiana se denomina *soft decision*. Un segundo bloque denominado PMU (*Path Metric Unit*) suma las métricas de rama para las trayectorias supervivientes, que en principio son las 128 iniciales. Las trayectorias no óptimas son desechadas y las óptimas son almacenadas en la unidad siguiente. El elemento principal del PMU es el ACS (*Add-Compare-Select*), la cual suma un número de métricas de rama, compara los resultados obtenidos, y selecciona el de menor distancia. Para un código con  $K$  *delays*, se requieren  $2^{K-2}$  módulos ACS [8, Sección 2.3]. Debido al crecimiento del valor numérico de la métrica de trayectoria, puede ocurrir un problema de *overflow*. Dado que no interesa el valor absoluto de dicha métrica, sino sólo valor comparable con las demás trayectorias, pueden aplicarse métodos de reducción del valor acumulado mediante resta simple ó detección de proximidad de *overflow*. La tercera unidad es el TBU (*TraceBack Unit*), el cual reconstituye la secuencia completa de estados cuya métrica de trayectoria es la de máxima verosimilitud. Este procedimiento recorre el enmallado desde el final hasta el principio, por lo que una unidad de FILO (*First-In Last-Out*) se requiere para la reconstrucción del mensaje en el orden correcto.

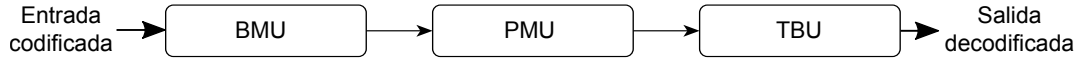


Figura 4.2: Unidades principales del decodificador con algoritmo de Viterbi.

Con respecto al algoritmo de Viterbi, debido a la cuantización en la señal de entrada, BMU, PMU y *traceback* finitos, el desempeño de la implementación en hardware disminuye alrededor de 1 [dB] comparado al algoritmo ideal. En cuanto al BMU, entre realizar *hard decision* y *soft decision* con 3 bits de precisión hay 2 [dB] de diferencia [17]. Debe señalarse que el algoritmo no es capaz de corregir errores en ráfagas (varios errores consecutivos), por lo que su desempeño se ve afectado negativamente al aplicarse en canales que introducen errores de ese tipo. Finalmente, es posible mejorar la cuantización para *soft decision* escogiendo umbrales de decisión  $T$  tales que

$$(4.1) \quad T = \sqrt{\frac{N_0}{2^w}}$$

donde  $w$  es el número de bits de cuantización [17]. Otra solución es la adición posterior de un entrelazador en el codificador, que reduce la probabilidad de errores en ráfaga decodificados, pues permuta los bits codificados mediante un algoritmo conocido, y recupera su formato inicial en el decodificador mediante un de-entrelazador. Esto redistribuye los bits erróneos que pueden haberse producido en ráfaga durante la transmisión [40, Cap. 10].

## 4.2. Implementación

### 4.2.1. Codificador convolucional

El codificador se implementa en software realizando un esquemático en el software ISE de Xilinx. Las unidades de memoria se realizan con bloques *flip-flop*, con entradas de datos, control de ciclos de reloj (**clk**), activación (**enable**) y reinicio (**reset**). La Figura 4.3 muestra esta implementación. Por sus características de entrada, salida y número de bloques de memoria se denotará al código como 'convolucional (2,1,7)'.

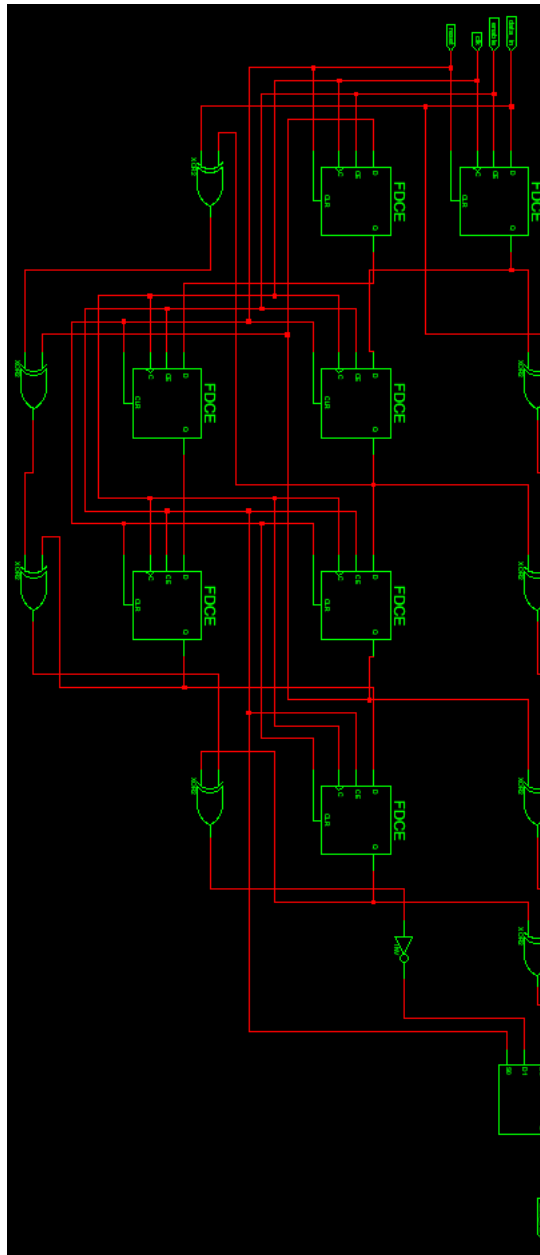


Figura 4.3: Esquemático RTL de codificador convolucional (2,1,7) realizado en ISE.

Tras compilar el archivo en VHDL y configurar los elementos mencionados, se obtiene la Tabla 4.4 donde se muestra la utilización de recursos de la FPGA.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	4	4656	0%
Number of Slice Flip Flops	7	9312	0%
Number of 4 input LUTs	1	9312	0%
Number of bonded IOBs	5	232	2%
Number of GCLKs	1	24	4%

Figura 4.4: Utilización de recursos en FPGA para decodificador Viterbi, generado en ISE.

### 4.2.2. Decodificador convolucional

El decodificador se implementa a partir del modelo en Simulink disponible en el Toolbox de Comunicaciones. Con el módulo HDL Coder es posible exportar el modelo a lenguaje descriptivo de hardware VHDL. En la Figura 4.5, las unidades FDCE corresponden a *flip-flops*, y las otras son el BMU, PMU y TBU de izquierda a derecha. El esquema de decisión en el BMU es de tipo *hard decision*, mientras que el largo total del *traceback* (*traceback depth*) es de  $t_b = 32$ . El método utilizado para prevenir *overflow* es sustraer el mínimo valor de las métricas de trayectoria para cada paso de tiempo [36]. Algunos elementos pueden ser optimizados, en particular el sistema de *traceback*, mediante la interfaz de usuario de HDL Coder.

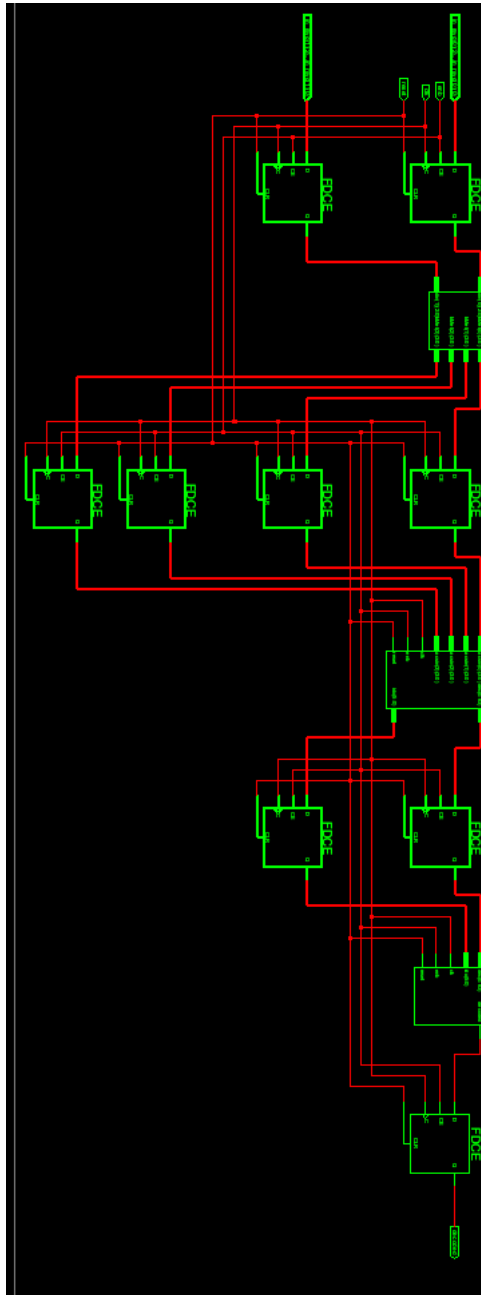


Figura 4.5: Esquemático RTL del decodificador con algoritmo de Viterbi, generado en ISE.

En el software ISE se realiza la compilación de los archivos VHDL, y tras otros procesamientos se traslada el diseño a la FPGA mediante el módulo iMPACT®. El uso de *slices* y otros recursos de la FPGA se detallan en la tabla de la Figura 4.6.



Device Utilization Summary (estimated values)				□
Logic Utilization	Used	Available	Utilization	
Number of Slices	3760	4656	80%	
Number of Slice Flip Flops	4013	9312	43%	
Number of 4 input LUTs	4359	9312	46%	
Number of bonded IOBs	11	232	4%	
Number of GCLKs	1	24	4%	

Figura 4.6: Utilización de recursos en FPGA para decodificador Viterbi, generado en ISE.

### 4.2.3. Revisión

De las Tablas 4.4 y 4.6, se aprecia la gran diferencia en empleo de recursos de la FPGA del codificador y decodificador. Los escasos componentes digitales del codificador redundan en un uso de *slices* menor al 0.09%. En contraste, la implementación de *soft-decoding* con 3 bits de cuantización, y la gran cantidad de *flip-flops* necesarios para cubrir todas las secuencias posibles de decodificar inciden en el alto porcentaje de ocupación de *slices* por el decodificador. El programa Simulink ejecuta por defecto la generación de código en VHDL con esta cuantización, pero es posible implementarlo con *hard decision* interviniendo el esquema en dicho programa, con la consecuente reducción en desempeño.

Al término de esta redacción, se está revisando la asignación de puertos del diseño a la tarjeta, dado que existen errores al respecto según el programa ISE; para ello se recurre a la documentación de usuario de la FPGA [62], donde se reseña su asignación manual. Otro problema pendiente es el reloj asignado al selector del multiplexor, en la salida del codificador, puesto que en principio se conectó directamente al `clk` habilitado para los *flip-flops*. Una vez corregidos estos problemas, la transferencia no debería representar dificultad. Las tareas posteriores consisten en realizar una transferencia de datos conocidos al codificador, cuya salida se comparará a la secuencia codificada mediante simulación. Comprobado el correcto funcionamiento del codificador, se dará por concluido el trabajo de implementación, quedando pendiente la revisión completa del decodificador.

Como recomendación a una posterior implementación y mejora en FPGA, se sugiere instalar y trabajar el programa ISE en el sistema operativo Windows©, debido a la ausencia de *drivers* para sistemas basados en Debian o Ubuntu, al menos en las versiones evaluadas para este trabajo (ISE v10 y v13.1).

# Capítulo 5

## Conclusiones

### 5.1. Conclusiones generales

La revisión bibliográfica ha permitido dar cuenta de la existencia de variados esquemas de codificación, que de distintas maneras permiten reducir los errores que los medios de transmisión añaden a las señales de interés, en diversas situaciones. De dichos esquemas se seleccionaron para su análisis aquellos que mejor parecen adecuarse a las aplicaciones satelitales, ya sea por su probado éxito como es el caso de los códigos Reed-Solomon (RS) y convolucional, o por su reciente incorporación a transpondedores espaciales, situación de los códigos BCH. En particular, se escogieron un esquema RS(255,239) y convolucional(171,133), ambos formando parte del estándar CCSDS, que condensa las recomendaciones de las agencias espaciales NASA y ESA; también se evaluó el código BCH(63,57), integrado al protocolo de comunicaciones del transpondedor de la misión satelital SUCHAI.

El estudio de estos esquemas consistió en su simulación computacional mediante la codificación y decodificación de secuencias binarias aleatorias, al ser transmitidas por un canal binario que asigna varias probabilidades de error de símbolo. Esto, como se demuestra en la literatura, es equivalente a la transmisión modulada en un canal con ruido Gaussiano (AWGN). Se emplearon modulaciones en fase binaria (BPSK) y en frecuencia binaria (BFSK) por su uso habitual en Comunicaciones. Estas simulaciones se contrastaron con transmisiones sin codificar para analizar la mejora en desempeño.

Los resultados indican que los códigos evaluados reducen en efecto la tasa de errores, pero con desempeños relativos diferentes según la probabilidad de error asignada al canal. El esquema BCH exige mayor potencia respecto a los otros para obtener tasas de error razonablemente bajas, para ambos formatos de modulación. Aún así, el peor desempeño esperable con este código es similar al de la transmisión sin codificación, y por lo tanto su inclusión en un sistema de comunicaciones representa una ventaja, considerando que su implementación es sencilla, su tasa efectiva de datos por secuencia es alta ( $R_{BCH} = 0.905$ ), y la longitud de los bloques codificados es lo bastante corta para asegurar la transmisión completa de bloques en un enlace.

Para el esquema Reed-Solomon se obtuvo un resultado idéntico a la transmisión sin codificar, hasta una potencia crítica desde la que experimenta una caída pronunciada, alcanzando tasas de error inferiores a  $10^{-6}$ . En promedio, este código supera en ganancia a BCH, y puede ser una buena opción teniendo en cuenta que su peor desempeño promedio también se iguala al caso sin codificación, además de tener la tasa de código más alta ( $R_{RS} = 0.937$ ). Sin embargo, al constituir un alfabeto no binario, cada símbolo es representado por más de un bit, y cada secuencia consta de 255 símbolos. Luego, para decodificar se requiere un mayor tiempo para transmitir todos los bits de una única secuencia, con el peligro de que el enlace se debilite o se pierda. Puede reducirse este tiempo si se aumenta el número de símbolos del modulador, aunque ello requiere mayor energía (para PSK) o ancho de banda (en FSK) para mantener la misma tasa de errores.

El código convolucional presenta la mayor ganancia relativa a los demás esquemas, aunque difiere su comportamiento según la modulación empleada en el intervalo de potencias simulado. Para BPSK se obtienen tasas de error razonables desde 2 [dB], mientras que en BFSK se logran desde 7.2 [dB]. En este último esquema, para potencias bajo 5 [dB] se produce un desempeño que empeora respecto a la transmisión sin codificar. Esto puede ser debido a la incapacidad del decodificador convolucional para corregir secuencias con errores consecutivos, por las características del algoritmo de Viterbi. No parece entonces recomendado implementar este código con modulación BFSK sin realizar mejoras; éstas pueden ser la adición de un entrelazador en el codificador, y/o un cuantizador en el demodulador, pues el primero reduce los errores en ráfaga en el receptor, y el segundo reduce la tasa de errores. El real desempeño de estas mejoras queda pendiente de evaluar.

Respecto a las modulaciones usadas, por sí misma BPSK tiene tasas de error más bajas que BFSK para una misma potencia, y mejora su desempeño al implementar codificación. Por otra parte, es sabido que el esquema MSK tiene las mismas probabilidades de error que BPSK. Es posible entonces utilizar esta modulación en lugar de BPSK, siendo más flexible en cuanto a la demodulación, que puede ser no coherente con degradación muy moderada. Un análisis comparativo más refinado entre estos dos esquemas no se aborda en este trabajo.

De estos resultados, se concluye que la combinación más promisoría para reducir la potencia y obtener tasas de error bajas está dada por un código convolucional y modulación BPSK o MSK. Las mismas mejoras antes mencionadas para FSK pueden ser implementadas, reduciendo aún más la potencia para lograr un mismo BER. El circuito secuencial del codificador es bastante simple, y la adición de un entrelazador no presenta complicaciones prácticas. En tanto, el receptor puede incluir un desentrelazador y cuantización. De todos modos, el decodificador es bastante más complejo, y requiere más recursos de hardware, del modo que se diseñó en software; esto conlleva que este esquema se recomiende sólo para telemetría. Aún así, según lo expuesto en el Capítulo 2, se han desarrollado métodos para reducir la complejidad del decodificador.

Una ventaja adicional del código convolucional es que el codificador no exige un largo fijo de datos binarios, por lo que puede adaptarse a cualquier protocolo. En cuanto a la baja tasa de código de este esquema ( $R = 0.5$ ), ésta puede incrementarse, mediante *perforación*, a tasas de  $3/4$ ,  $5/6$  y  $7/8$ , aunque ello degrada levemente el desempeño.

Para la transmisión de telecomandos, el actual transpondedor incluye un esquema BCH con decodificador de síndrome, cuya implementación es la más simple y permite corregir o detectar errores. La simplicidad del esquema, su desempeño mejor o igual a una transmisión no codificada, y su buena tasa de código, validan este método como adecuado para el envío de telecomandos de poca longitud.

La implementación de un esquema de codificación y decodificación convolucional sólo fue lograda en software, y queda pendiente la transferencia del diseño a la FPGA y su evaluación mediante envíos de secuencias binarias. Lo realizado demostró sin embargo que el decodificador ocupa bastantes recursos de hardware, con un 80 % de *slices* de la FPGA ocupados, mientras que los ocupados por el codificador no alcanzan a constituir el 0.1 %.

En cuanto a tareas pendientes, en particular el codificador requiere la revisión de los relojes asignados y los puertos asignados, para ser embebido en el chip. Una manera de probar su funcionamiento es codificar secuencias conocidas mediante el hardware y usando *MatLab* y compararlas. Así puede analizarse el desempeño del codificador sin requerir decodificar las secuencias mediante el esquema Viterbi. De este último, deben revisarse los *pines* asignados, y puede evaluarse de forma similar al codificador.

## 5.2. Trabajos Futuros

Este trabajo constituye una novedad por cuanto se orienta al aporte concreto de investigación para el desarrollo del ámbito satelital en el Departamento de Ingeniería Eléctrica. En conjunto con los aportes realizados en paralelo, los cuales abarcan tópicos de codificación modulada entrelazada (BICM) y de estimación de canal, se abren varias posibilidades de investigación futura:

- En implementación, la concreción de codificador y decodificador convolucional en hardware, realización de pruebas, y mejoramiento en software constituyen el paso posterior para la completitud y continuidad del trabajo. En concreto, se menciona en 4.1.2 la diferencia de potencia entre diseñar un BMU con *hard decision* y *soft decision*. En dicho caso, puede aún explotarse la cuantización en *soft decision* diseñando umbrales de decisión también citados. Concatenando un bloque adicional de entrelazamiento de bits se fortalece a este código contra un canal con ráfagas, al cual es más susceptible que otros códigos.
- Las simulaciones realizadas se limitan al análisis del canal AWGN, estudiando su equivalente en el canal BSC. Dado el desconocimiento acerca de la real influencia de los agentes de ruido descritos en 2.4.2, puede extenderse el modelo de canal añadiendo comportamientos de ráfagas, de magnitud variable, y estudiar el desempeño de los esquemas revisados en tales circunstancias, dando espacio para el análisis de otros esquemas de codificación no considerados en este trabajo, como los códigos Turbo o LDPC.

- De manera análoga a lo realizado en este trabajo, es de interés estudiar formas de modulación adecuadas para enlaces satelitales. En general los transpondedores diseñados con este fin emplean esquemas en frecuencia, pero éstos son mejorables en cuanto a tasas de error. Es posible que la implementación de esquemas en fase sea algo más compleja, pero la revisión bibliográfica dio cuenta de varios diseños que pueden simplificar aquella sin degradación significativa, y que probablemente no hayan sido considerados por los fabricantes en favor de diseños simples.

# Glosario

ACS	Add Select Compare.
ARQ	Automatic Repeat Request.
ASK	Amplitude Shift Keying.
AWGN	Additive White Gaussian Noise.
BCH	Códigos Bose-Chaudhuri-Hocquenghem.
BER	Bit Error Rate.
BFSK	Binary Frequency Shift Keying.
BMU	Branch Metric Unit.
BPSK	Binary Phase shift Keying.
BSC	Binary Symmetric Channel.
CCSDS	Consultative Committee for Space Data Systems.
CLB	Configurable Logic Block.
CPFSK	Continuous Phase FSK.
CPM	Continuous Phase Modulation.
CRC	Cyclic Redundancy Check.
DEBPSK	Differentially Encoded BPSK.
DMC	Discrete Memoryless Channel.
DPSK	Differential PSK.
$\frac{E_b}{N_0}$	Razón entre la energía por bit y la densidad espectral del ruido.
ESA	European Space Agency.
FEC	Forward Error Correction.
FILO	First-In Last-Out.
FPGA	Field Programmable Gate Array.
FSSM	Finite State Sequential Machine.
GEO	Geostationary Earth Orbit.
HDL	Hardware Description Language.
LDPC	Low Density Parity Check.
LEO	Low Earth Orbit.
LUT	Look Up Table.
MAP	Maximum A-posteriori Probability.
MLD	Maximum Likelihood Decoding.

MLE	Maximum Likelihood Estimation.
MSK	Minimum Shift Keying.
NASA	National Aeronautics and Space Administration.
NRZ	Non Return to Zero.
OBC	On Board Computer.
OOK	On-Off Keying.
OQPSK	Offset Quadrature Phase Shift Keying.
$P_b$	Probabilidad de error de bit.
PMU	Path Metric Unit.
P-POD	Poly-Picosatellite Orbital Deployer.
QC-LDPC	Quasi-Cyclic Low Density Parity Check.
QPSK	Quadrature Phase Shift Keying.
$R$	Tasa de código.
RS	Códigos Reed-Solomon.
RZ	Return to Zero.
SNR	Signal-to-Noise Ratio.
SUCHAI	Satellite of University of Chile for Aerospace Investigation.
TBU	Traceback Unit.
UHF	Ultra High Frequency.
XOR	Exclusive-OR.

# Bibliografía

- [1] R. J. Adler, R. E. Feldman, and M. S. Taqqu, *A Practical Guide to Heavy Tails*. Birkhaeuser, 1998.
- [2] AllSpace, *AS-COM-01 Documentation*, AllSpace, Julio 2011.
- [3] —, *AS-COM-01 User Manual*, AllSpace, 2011. [Online]. Available: <http://www.cubesat.de/en/index.htm>
- [4] S. Asmussen, *Applied Probability and Queues*, 2nd ed., ser. Applications of Mathematics, B. Rozovskii and M. Yor, Eds. Springer, 2003.
- [5] A. Becerra, “Implementación de codificación de canal para sistemas de comunicaciones digitales,” Mayo 2010.
- [6] E. R. Berlekamp, “On decoding binary Bose-Chaudhuri-Hocquenghem codes,” *IEEE Trans. Inf. Theory*, vol. IT-11, pp. 577–580, 1965.
- [7] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correction coding and decoding,” *IEEE Int. Communications Conf.*, pp. 1064–1070, Mayo 1993.
- [8] S. Bhattacharyya, R. Leupers, E. Deprettere, and J. Takala, Eds., *Handbook of Signal Processing Systems*. Springer, 2010.
- [9] R. Blahut, *Algebraic Codes for Data Transmission*, 2nd ed. Cambridge, United Kingdom: Cambridge University Press, 2003.
- [10] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Inf. Control*, vol. 3, pp. 68–79, Marzo 1960.
- [11] J. Castiñeira and P. Farrell, *Essentials of Error-Control Coding*. John Wiley & Sons, Ltd, 2006, ch. 1st, pp. 41–65.
- [12] S. Chung, D. Forney, T. Richardson, and R. Urbanke, “On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit,” *IEEE Communications Letters*, pp. 58–60, Febrero 2001.
- [13] D. Costello and S. Lin, *Error Control Coding, Fundamentals and Applications*, ser. Computer Applications in Electrical Engineering, A. Simpson, Ed. Prentice-Hall, 1983.



- [14] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, “Applications of error-control coding,” *IEEE Trans. Inf. Theory*, vol. 44, pp. 2531–2560, Octubre 1998.
- [15] H. Deng, H. Li, and Z. Li, “FPGA Implementation of Cascade Correction for RS, Interleaved and Convolutional Codes,” *Information and Control*, vol. 6, 2007.
- [16] S. Derteano, “Evaluación y aplicación de estrategias para control de errores en canales satelitales mediante BICM,” Agosto 2012.
- [17] C. Fleming. (2011) A Tutorial on Convolutional Coding with Viterbi Decoding. Spectrum Applications Technical Consultancy. [Online]. Available: <http://home.netcom.com/~chip.f/viterbi/tutorial.html#specapps>
- [18] C. C. for Space Data Systems, *TM Synchronization and Channel Coding*, CCSDS, Agosto 2011. [Online]. Available: <http://public.ccsds.org/publications/BlueBooks.aspx>
- [19] D. Forney, “Concatenated codes,” MIT, Tech. Rep., 1966.
- [20] —, “Principle of digital communications 2,” 2005. [Online]. Available: [ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-451-principles-of-digital-communication-ii-spring-2005/lecture-notes/chap\\_2.pdf](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-451-principles-of-digital-communication-ii-spring-2005/lecture-notes/chap_2.pdf)
- [21] —, “Principle of digital communications 2,” 2005. [Online]. Available: [ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-451-principles-of-digital-communication-ii-spring-2005/lecture-notes/chap\\_1.pdf](http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-451-principles-of-digital-communication-ii-spring-2005/lecture-notes/chap_1.pdf)
- [22] H. Fujita and K. Sakaniwa, “An Efficient Encoding Method for LDPC Codes Based on Cyclic Shift,” Tokyo Institute of Technology, Tech. Rep., Julio 2004.
- [23] R. G. Gallager, “Low density parity check codes,” *IRE Trans. Inf. Theory*, vol. IT-8, pp. 21–28, 1962.
- [24] *NanoCom U480 Datasheet*, GomSpace, Abril 2011.
- [25] M. Guo, M. Ahmad, M. Swamy, and C. Wang, “FPGA design and implementation of a low-power systolic array-based adaptive Viterbi decoder,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 2, no. 52, pp. 350–365, 2005.
- [26] S. Habinc, “Suitability of reprogrammable FPGAs in space applications,” *Gaisler Research, Feasibility Report*, 2002.
- [27] R. W. Hamming, “Error detecting and error correcting codes,” *Bell Systems Technology Journal*, vol. 29, pp. 147–160, Abril 1950.
- [28] A. Hocquenghem, “Codes correcteurs d’erreurs,” *Chiffres*, vol. 2, pp. 147–156, 1959.
- [29] H. C. Hsieh, “Characteristics of ionospheric thermal noise,” *Journal of Atmospheric and Terrestrial Physics*, vol. 28, pp. 769–781, 1966.

- [30] *Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes*. IEEE Globecom, 2005.
- [31] Y. Jinsha and C. Zhixiong, “Constructing Quasi-Cyclic LDPC Codes For Efficient Encoding Based on Fast Gauss Elimination,” North China Electric Power University, Tech. Rep., 2008.
- [32] T. Langdon. Working your first amateur satellite. AMSAT Radio Amateur Satellite Corporation. [Online]. Available: <http://www.amsat.org/amsat-new/information/faqs/langdon.php>
- [33] D. J. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Trans. Inf. Theory*, pp. 399–431, Marzo 1999.
- [34] D. J. MacKay and R. M. Neal, “Near Shannon limit performance of low-density parity-check codes,” *Electron. Lett.*, pp. 1645–1646, Agosto 1996.
- [35] J. L. Massey, “Step-by-step decoding of the Bose-Chaudhuri-Hocquenghem codes,” *IEEE Trans. Inf. Theory*, vol. IT-11, pp. 580–585, Oct. 1965.
- [36] MathWorks, *HDL Code Generation for Viterbi Decoder*, Mathworks Inc. [Online]. Available: [http://www.mathworks.com/products/communications/examples.html?file=/products/demos/shipping/comm/commviterbihdl\\_m.html](http://www.mathworks.com/products/communications/examples.html?file=/products/demos/shipping/comm/commviterbihdl_m.html)
- [37] J. Meggit, “Error correcting codes and their implementation,” *IRE Trans. Inf. Theory*, vol. IT-7, pp. 232–244, Octubre 1961.
- [38] *Power reduction on QPSK modulation codings for geopositioning application using LEO satellites*. Microwave and Optoelectronics Conference (IMOC), 2009 SBMO/IEEE MTT-S International, Noviembre 2009.
- [39] T. Mizuochi, Y. Konishi, Y. Miyata, T. Inoue, K. Onohara, S. Kametani, T. Sugihara, K. Kubo, H. Yoshida, T. Kobayashi *et al.*, “Experimental demonstration of concatenated LDPC and RS codes by FPGAs emulation,” *Photonics Technology Letters, IEEE*, vol. 21, no. 18, pp. 1302–1304, 2009.
- [40] T. K. Moon, *Error Correction Coding*. John Wiley & Sons Inc., 2005.
- [41] R. Neri, *Comunicaciones por satélite*. Thomson, 2003. [Online]. Available: <http://thomsonlearning.com.mx>
- [42] F. J. Olivier, “An LDPC Error Control Strategy for Low Earth Orbit Satellite Communication Link Applications,” Master’s thesis, Stellenbosch University, Diciembre 2009.
- [43] E. Prange, “Cyclic Error-Correcting Codes in Two Symbols,” *Air Force Cambridge Research Center*, Septiembre 1957.
- [44] J. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2003.

- [45] J. Proakis and M. Salehi, *Communication Systems Engineering*, 2nd ed. Prentice-Hall, Agosto 2002.
- [46] T. L. P. C. Programme, “Electromagnetic compatibility aspects of radio-based mobile telecommunication systems,” ERA Technology Ltd, Tech. Rep., 1999. [Online]. Available: <http://www.emc.york.ac.uk/reports/linkpcp/index.html>
- [47] J. Qingchun, “Encoder and decoder design for convolutional code based on FPGA,” *Electronic Measurement Technology*, vol. 2, 2008.
- [48] I. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Ind. Appl. Math.*, vol. 8, pp. 300–304, 1960.
- [49] T. Richardson and R. Urbanke, “Efficient Encoding of Low-Density Parity-Check Codes,” *IEEE Trans. Inf. Theory*, pp. 638–656, Febrero 2001.
- [50] R. Roosta, “A comparison of radiation-hard and radiation-tolerant FPGAs for space applications,” *NASA Electronic Parts and Packaging (NEPP) Program JPL D-31228*, 2004.
- [51] C. E. Shannon, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Julio, Octubre 1948.
- [52] M. Shaqfeh and N. Goertz, “Systematic Modification of Parity-Check Matrices for Efficient Encoding of LDPC Codes,” University of Edinburgh, Tech. Rep., 2008.
- [53] C. Spagnol, W. Marnane, and E. Popovici, “FPGA implementations of LDPC over GF(2<sup>m</sup>) decoders,” in *Signal Processing Systems, 2007 IEEE Workshop on*. IEEE, 2007, pp. 273–278.
- [54] L. Sun, H. Song, Z. Keirn, and B. Kumar, “Field programmable gate array (FPGA) for iterative code evaluation,” *Magnetics, IEEE Transactions on*, vol. 42, no. 2, pp. 226–231, 2006.
- [55] “Satellite classification,” Surrey Satellite Technology Ltd. [Online]. Available: [http://centaur.sstl.co.uk/SSHP/sshp\\_classify.html](http://centaur.sstl.co.uk/SSHP/sshp_classify.html)
- [56] A. J. Viterbi, “Error bounds for convolutional codes and asymptotically optimum decoding algorithm,” *IEEE Trans. Inf. Theory*, vol. IT-13, pp. 260–269, Abril 1967.
- [57] A. J. Viterbi and J. K. Omura, *Principle of Digital Communication and Coding*, 1st ed., ser. McGraw-Hill Series In Electrical Engineering, S. W. Director, Ed., 1979.
- [58] Z. Wang and Z. Cui, “Low-complexity high-speed decoder design for quasi-cyclic LDPC codes,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 1, pp. 104–114, 2007.
- [59] R. Wiid, “Implementation of a Protocol and Channel Coding Strategy for use in Ground-Satellite Applications,” Master’s thesis, Stellenbosch University, Marzo 2012. [Online]. Available: <http://scholar.sun.ac.za/handle/10019.1/20346?show=full>

- [60] J. Wozencraft and B. Reiffen, *Sequential Decoding*. MIT Press, Marzo 1961.
- [61] P. Wu and S. Pisuk, "Implementation of a low complexity, low power, integer-based Turbo decoder," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, vol. 2. IEEE, 2001, pp. 946–951.
- [62] Xilinx, *Spartan-3 Generation Configuration User Guide*, Xilinx Inc. [Online]. Available: [http://www.xilinx.com/support/documentation/user\\_guides/ug331.pdf](http://www.xilinx.com/support/documentation/user_guides/ug331.pdf)
- [63] —, *Spartan-3 Generation FPGA User Guide*, Xilinx Inc. [Online]. Available: [http://www.xilinx.com/support/documentation/user\\_guides/ug332.pdf](http://www.xilinx.com/support/documentation/user_guides/ug332.pdf)
- [64] F. Xiong, *Digital Modulation Techniques*, 1st ed. Artech House, 2000.
- [65] Z. Yang, Q. Xie, K. Peng, and J. Fu, "A Fast and Efficient Encoding Structure for QC-LDPC Codes," Tsinghua University, Tech. Rep., 2008.
- [66] D. Zhao and H. Liu, "FPGA Implementation of Viterbi Decoding Algorithm for Convolutional Codes," *Modern Electronic Technique*, vol. 1, 2004.

# Apéndice A

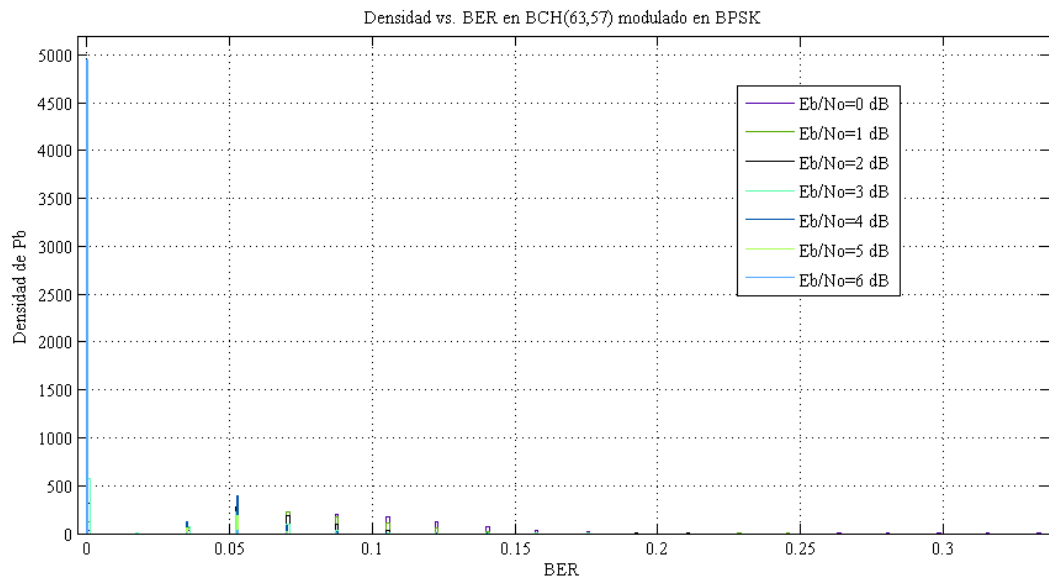
## Distribuciones de tasas de error en codificación

En este apéndice se exponen los histogramas obtenidos para las tasas de error producidas por los esquemas de codificación simulados en el Capítulo 3. Para todos los casos, aunque en diversos grados, las curvas tienden a acumular los valores de BER en 0 a medida que se incrementa el nivel de  $E_b/N_0$ . Este fenómeno se caracteriza como una familia de distribuciones, que reciben el nombre de *distribuciones de cola pesada*, por la mencionada acumulación de realizaciones en torno a algún valor. La definición y propiedades de ellas pueden ser revisadas en [4], y sus aplicaciones se hallan en [1].

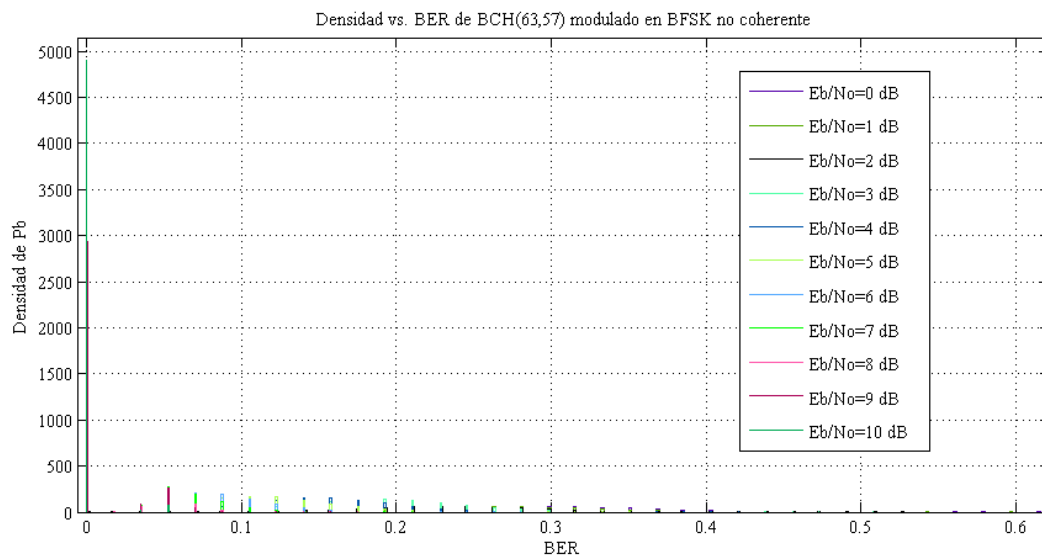
Es importante señalar que en estos histogramas la cantidad de tasas de error concentradas en 0 puede ser órdenes de magnitud superior a las distribuidas en BER mayores, oscilando entre  $10^3$  y  $10^5$  veces dependiendo del esquema de codificación y modulación. En general esto no se aprecia visiblemente, dado que en los gráficos además se intenta mostrar las curvas iniciales, de baja frecuencia de ocurrencia.

Los histogramas se obtienen mediante la herramienta `dfittool` de Matlab.

## A.1. Distribución de BER código BCH



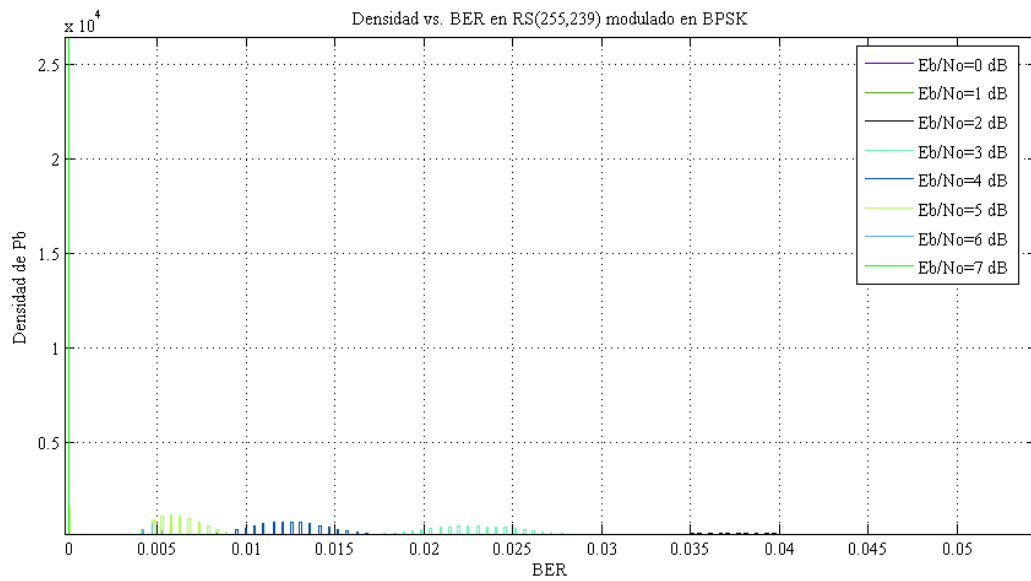
(a)



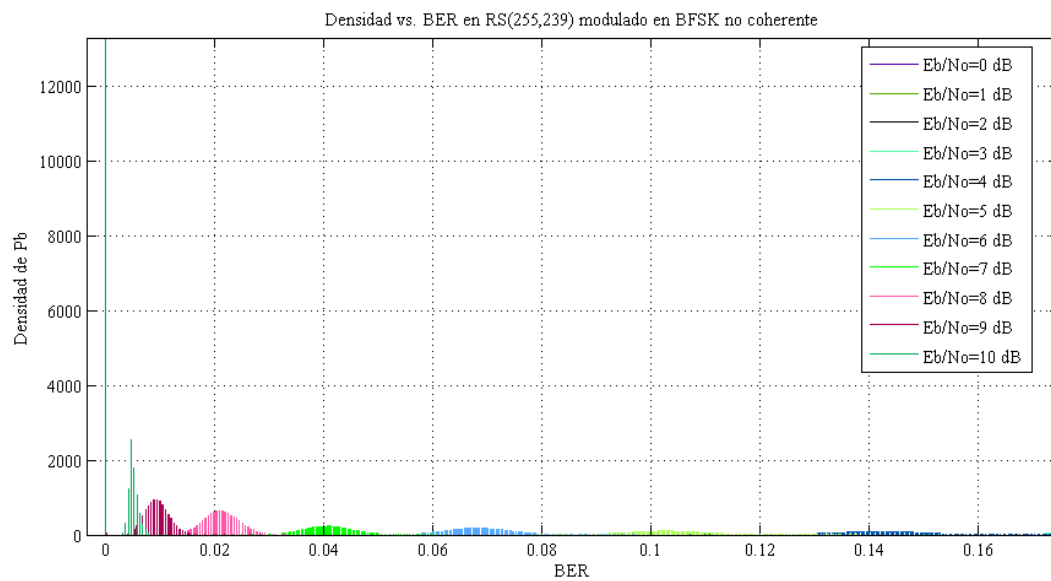
(b)

Figura A.1: Distribución de BER en código BCH(63,57) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK.

## A.2. Distribución de BER código Reed-Solomon



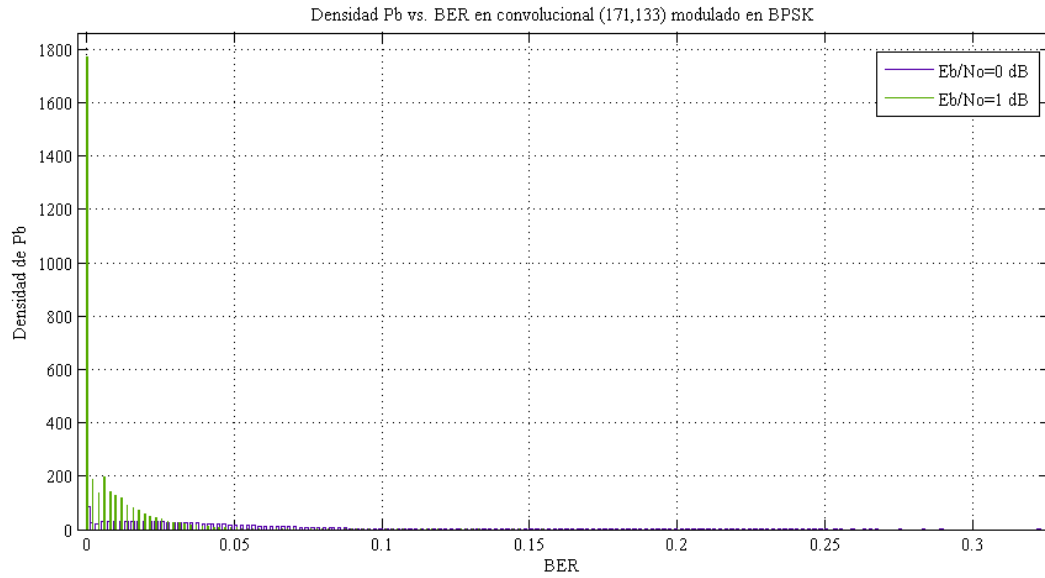
(a)



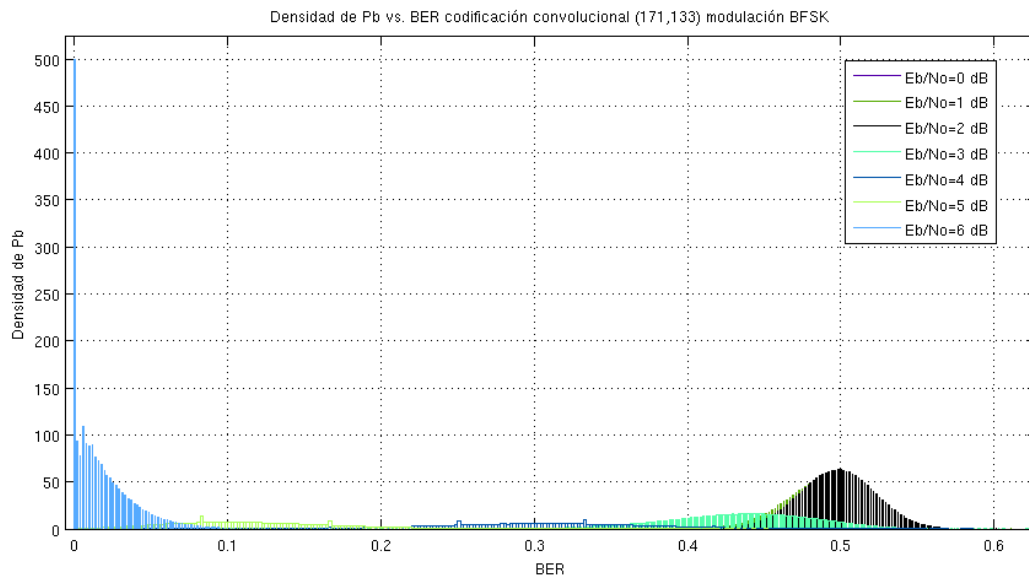
(b)

Figura A.2: Distribución de BER en código RS(255,239) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK.

### A.3. Distribución de BER código convolucional



(a)



(b)

Figura A.3: Distribución de BER en código convolucional(171,133) en canal binario simétrico. (a) Modulación BPSK, (b) Modulación BFSK.



## A.4. Aspectos de distribuciones observadas

Se aprecia que los esquemas simulados tienen el comportamiento descrito de acumulación situada en  $BER=0$ . Esto ocurre a medida que se incrementa la potencia asignada, que es equivalente a disminuir la probabilidad de error del canal. A mayor energía, más secuencias son correctamente decodificadas; el número de tales secuencias crece más que proporcionalmente con el incremento lineal de potencia, siendo muy notorio el caso de Reed-Solomon con BPSK, Figura A.2a, donde con 7 [dB] se acumulan 25000 secuencias decodificadas con éxito. Más aún, no se extendió el rango de potencia pues con 8 [dB] es imposible distinguir visualmente las curvas iniciales.

Aún cuando no se ve claramente en todos los casos, a bajas potencias la distribución del BER es una Normal centrada en la tasa promedio, y el incremento de energía las traslada hacia la izquierda, produciendo el efecto consabido. La Figura A.2b ilustra bien esta observación.

Por último, es interesante notar las diferencias en la dispersión de las distribuciones; por ejemplo, el código BCH tiene tasas muy dispersas, lo que se refleja en una mayor variación en torno al promedio, como fue expuesto en los respectivos gráficos del Capítulo 3. El esquema Reed-Solomon tiene distribuciones más concentradas para ambas modulaciones, y el convolucional es muy disperso. En este último caso, con BPSK se presenta una acumulación muy pronunciada en 0 con 1 [dB], y en 2 [dB] sólo se aprecia gráficamente la línea centrada en 0; esto comprueba que las secuencias correctamente decodificadas son muy superiores en cantidad respecto de aquellas erróneas. En el caso de BFSK se observa una distribución centrada en  $BER=0.5$ , con una alta concentración de secuencias erróneas, para potencias bajas, ilustrando la ineficacia de este esquema bajo modulación en frecuencia.