

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

COMPARACIÓN DE LAS TÉCNICAS DE CODIFICACIÓN DE CANAL DE QUINTA GENERACIÓN NR-LDPC Y NR-POLAR EN UN CANAL AWGN CON MODULACIÓN BPSK Y CUANTIZACIÓN DE PUNTO FIJO MEDIANTE UNA SIMULACIÓN EN MATLAB

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

JUAN PABLO CORTEZ MOSQUERA

DIRECTOR: PABLO ANÍBAL LUPERA MORILLO

Quito, julio 2021

AVAL

Certifico que el presente trabajo fue desarrollado por Juan Pablo Cortez Mosquera, bajo mi supervisión.

PABLO ANÍBAL LUPERA MORILLO
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Juan Pablo Cortez Mosquera, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Juan Pablo Cortez Mosquera

DEDICATORIA

En la querida memoria de quienes en vida fueron Jaime Alfonso Mosquera Viteri, René Cortez Villavicencio, Mercedes Yolanda Arias Paz y Zoila Genoveva Alvear Camacho. Grandes personas, devotos padres y los mejores abuelos.

A mis padres, Pablo y Ximena, el mejor soporte e inspiración de mi vida.

AGRADECIMIENTO

A toda mi familia, en especial a mis tíos, Carlos y Patricia, César y Sonia, Byron y Ñachita, Esperancita y Jaime, Boris y María Isabel y Anita y Santiago por su apoyo incondicional e inspiración en este tiempo para poder graduarme y poderme superar cada día.

A mi hermano, Felipe, como a mis primos, Jaime David, Julito, Byriton, César, Dani, Nati, Mile, Andrés, Emilio, Alejandrita, Mati, Mikita, Lucianita y Johanita por ser un gran motor para inspirarme a ser el mejor.

A todos mis amigos dentro y fuera de la Universidad que he hecho a lo largo de este tiempo, de los cuales tengo los mejores recuerdos y siempre esperaré por escribir nuevos. Un agradecimiento especial a Rafa, Sarita, Ángel, Sebas y Alex, quienes son el mejor ejemplo que un amigo puede tener.

A mi familia hardstyle, del cual poco a poco se ha convertido en la pieza del rompecabezas que une cada aspecto de mi vida. Aquí lloverían los agradecimientos a cientos de personas que han hecho de mí la persona que soy en hoy en día. Quiero agradecer de manera especial a Marco Hammer, Jeroen y Martjin Boeren, grandes amigos que siempre estaré agradecido por darme una oportunidad de contar historias y poder colaborar con mi granito de arena con lo que llamo hogar.

A la planta docente y administrativa de la Escuela Politécnica Nacional y su comunidad por darme una oportunidad de crecer como ser humano y profesional y poder ayudar de mejor forma a la sociedad. Quiero agradecer en especial al PhD. Pablo Lupera, quien ha guiado mis pasos en la realización de este proyecto técnico con mucha paciencia y me ha ayudado de una manera extraordinaria. Además, quiero agradecer al PhD. Diego Reinoso, al Ing. Jorge Grijalva y a la Secre Mayrita, con ustedes mi agradecimiento y deuda eterna.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1. INTRODUCCIÓN.....	1
1.1.1 OBJETIVO PRINCIPAL.....	1
1.1.2 OBJETIVOS ESPECÍFICOS.....	1
1.3.1 INTRODUCCIÓN A LOS SISTEMAS DE COMUNICACIONES INALÁMBRICAS.....	5
1.3.2 MODULACIÓN.....	7
1.3.3 RENDIMIENTO DE UN SISTEMA DE COMUNICACIÓN INALÁMBRICO...	9
1.3.4 CODIFICACIÓN DE CANAL.....	13
1.3.5 DECODIFICACIÓN DE CANAL.....	18
1.3.6 CODIFICACIÓN LINEAL DE BLOQUES	25
1.3.7 INTRODUCCIÓN A LOS ESQUEMAS DE CODIFICACIÓN LDPC Y POLAR PARA SISTEMAS <i>NEW RADIO</i> DE QUINTA GENERACIÓN.....	29
1.3.8 CODIFICACIÓN <i>LDPC (LOW DENSITY PARITY CHECK CODES)</i> PARA SISTEMAS DE RADIO <i>NEW RADIO 5G</i>	31
1.3.9 CODIFICACIÓN NR-POLAR	54
2. METODOLOGÍA.....	72
2.1. GENERALIDADES.....	72
2.2. REQUISITOS TÉCNICOS PARA LA SIMULACIÓN DE LOS SISTEMAS DE COMUNICACIÓN PROPUESTOS.....	74
2.2.1. GENERACIÓN DEL MENSAJE.....	75
2.2.2. MODULACIÓN BPSK Y CANAL AWGN.....	75
2.2.3. CODIFICACIÓN.....	76
2.2.4. CUANTIZACIÓN DE PUNTO FIJO (<i>FIXED-POINT QUANTIZATION</i>)	78
2.2.5. DECODIFICACIÓN.....	78
2.2.6. CONTEO DE ERRORES Y TIEMPO DE PROCESAMIENTO.....	80
2.2.7. VISUALIZACIÓN DE RESULTADOS Y ALMACENAMIENTO.....	81

2.3.	IMPLEMENTACIÓN DE LOS SISTEMAS DE COMUNICACIÓN	81
2.3.1.	VARIABLES PREDEFINIDAS.....	81
2.3.2.	IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN CON MODULACIÓN BPSK SOBRE CANAL AWGN Y CUANTIFICACIÓN CON REPRESENTACIÓN DE PUNTO FIJO.....	83
2.3.3.	IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN CON EL ESQUEMA DE CODIFICACIÓN NR-LDPC	85
2.3.4.	IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN CON EL ESQUEMA DE CODIFICACIÓN NR-POLAR.....	91
2.3.5.	IMPLEMENTACIÓN PARA VISUALIZACIÓN DE RESULTADOS.....	100
3.	RESULTADOS Y DISCUSIÓN	103
3.1	ESCENARIO DE SIMULACIÓN	103
3.2	PRUEBAS DE FUNCIONAMIENTO.....	104
3.3	RENDIMIENTO DE LOS ESQUEMAS DE CODIFICACIÓN	107
3.3.1	RENDIMIENTO DEL ESQUEMA DE CODIFICACIÓN NR-LDPC	108
3.3.2	RENDIMIENTO DEL ESQUEMA DE CODIFICACIÓN NR-POLAR.....	115
3.3.3	COMPARACIÓN DEL RENDIMIENTO DE LOS ESQUEMAS NR-LDPC Y NR-POLAR Y ASIGNACIÓN DEL PLANO DE USUARIO Y CONTROL.....	119
4.	CONCLUSIONES Y RECOMENDACIONES.....	125
4.1	CONCLUSIONES.....	125
4.2	RECOMENDACIONES	129
4.3	TRABAJOS FUTUROS	130
	ANEXOS	138

RESUMEN

Una de las características de las comunicaciones inalámbricas consiste en que los datos recibidos difieren de los transmitidos debido a fenómenos/anomalías presentes en el canal inalámbrico, tales como el desvanecimiento, la interferencia y el ruido, un inconveniente que puede ser solucionado con la implementación de códigos de canal. En el caso de las comunicaciones de radio celular en las especificaciones del Proyecto de Tercera Generación (*Third Generation Partnership Project* -3GPP por sus siglas en inglés-) se encuentran las especificaciones de la quinta generación de redes de radio celular denominada *New Radio* (NR). La versión 15 del mencionado estándar a través del Reporte Técnico TS 38.212 define en base a estudios teóricos y prácticos dos nuevos esquemas de codificación de canal: NR-LDPC (*New Radio – Low Density Parity Check Codes*) y NR-Polar (*New Radio-Polar*). El objetivo de introducir dichos esquemas es brindar soporte a los tres nuevos campos de aplicación definidos en el reporte técnico TR 138.912 para *New Radio* (*enhanced Mobile Broadband -eMBB-, Ultra Reliable Low Latency Communications -URLLC-, massive Machine Type Communications -mMTC-*). El presente documento hace una recopilación teórica y una evaluación del rendimiento de ambos esquemas de codificación con el objetivo de analizar sus características. Los resultados han concluido que ambos esquemas manejan características definidas tales como el número de iteraciones para NR-LDPC o de número de listas para NR-Polar para obtener rendimientos similares en cuanto a tasa de bits errados a una tasa de codificación similar.

PALABRAS CLAVE: 5G, New Radio, Codificación, Decodificación, LDPC, Polar

ABSTRACT

One of the characteristics of wireless communications is that the received data differs from the transmitted data due to phenomena / anomalies present in the wireless channel, such as fading, interference and noise, a drawback that can be solved with the implementation of channel codes. In the case of cellular radio communications, the specifications of the fifth generation of cellular radio networks called New Radio (NR) can be found in the specifications of the Third Generation Partnership Project (3GPP). Version 15 of the standard through Technical Report TS 38.212 defines, based on theoretical and practical studies, two new channel coding schemes: New Radio - Low Density Parity Check Codes (NR-LDPC) and New Radio-Polar (NR- Polar). The objective of introducing these schemes is to provide support to the three new fields of application defined in technical report TR 138.912 for New Radio (enhanced Mobile Broadband -eMBB-, Ultra Reliable Low Latency Communications -URLLC-, massive Machine Type Communications -mMTC-). This document makes a theoretical compilation and an evaluation of the performance of both coding schemes to analyze their characteristics. The results have concluded that both schemes handle defined characteristics such as the number of iterations for NR-LDPC or the number of lists for NR-Polar to obtain similar performance in terms of erroneous bit rates at a similar coding rate.

KEYWORDS: 5G, New Radio, Encoding, Decoding, LDPC, Polar

1. INTRODUCCIÓN

1.1 OBJETIVO PRINCIPAL

Comparar mediante la simulación en MATLAB, los esquemas de codificación y decodificación de la capa física de las especificaciones *New Radio - Low Density Parity Check Codes* (NR-LDPC) y *New Radio - Polar* en un canal AWGN con modulación BPSK y cuantización de punto fijo (*Fixed-Point Quantization*).

1.1.2 OBJETIVOS ESPECÍFICOS

- Estudiar los conceptos de codificación LDPC y Polar que son parte de la capa física de las especificaciones *New Radio* de la generación 5G.
- Proponer el diagrama de bloques de los codificadores y decodificadores respectivos.
- Implementar por medio de un script en MATLAB los codificadores y decodificadores en un sistema de comunicación con canal de ruido gaussiano blanco aditivo o AWGN por sus siglas en inglés (*Additive White Gaussian Noise*), modulación BPSK (*Binary Phase Shift Keying*) y cuantificación de punto fijo.
- Realizar pruebas de la implementación con la misma tasa de codificación R para ambos esquemas de codificación con diferentes Eb/No en dB y número de bloques enviados para obtener la tasa de bits errados o BER por sus siglas en inglés (*Bit Error Rate*), la tasa de tramas erradas o FER por sus siglas en inglés (*Frame Error Rate*), tiempo de procesamiento de codificación, decodificación y total.
- Comparar los resultados obtenidos entre esquemas de codificación y generar un análisis en base a los tres escenarios generales de aplicación de New Radio 5G.

El presente documento propone explicar, simular y comparar los esquemas de codificación de canal NR-LDPC y NR-Polar dentro de las especificaciones New Radio de la generación 5G definidos en el Reporte Técnico TS 38.212 [1]. Para esto, primero se desarrollará todo el marco teórico necesario para realizar las simulaciones requeridas que permitan presentar los resultados deseados. Posterior a eso, dichos esquemas se los implementará en MATLAB por medio de scripts en lenguaje .m y tendrán el esquema propuesto en la Figura 1.1.

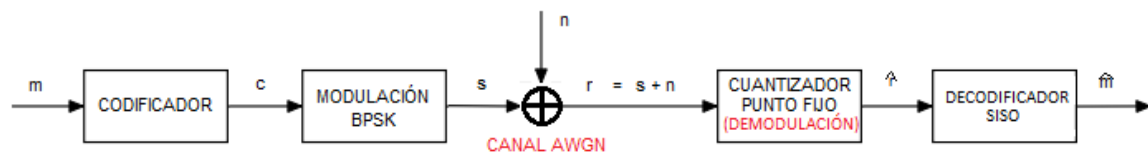


Figura 1.1. Sistema de comunicación a implementar en MATLAB con un bloque de codificación y modulación BPSK y enviada por medio de un canal AWGN.

El modelo planteado en la Figura 1.1 en su transmisor hace las operaciones de codificación y modulación BPSK de un mensaje de k bits, luego los símbolos son enviados por el canal AWGN y para que en la recepción sea cuantizado y decodificado con el objetivo de estimar el mensaje enviado. Recuérdese que el lado del transmisor generará un mensaje aleatorio de k bits, mismo que será codificado en NR-LDPC o NR-POLAR. Algunos detalles sobre estos esquemas se explican a continuación.

- La codificación LDPC se realiza mediante el manejo de matrices de verificación de paridad H . Específicamente para el estándar *New Radio 5G*, Reporte Técnico TS 38.212 define un proceso para obtener la matriz H por medio de la expansión en un factor Z_c de dos matrices/grafos base BG1 (46x68) o BG2 (42x52) [2]. Estos parámetros están en función de la longitud de la palabra código y la tasa de codificación R que se desea obtener. Para el presente proyecto se ha establecido una tasa de codificación de $R \approx 1/2$ con longitud de palabra código aproximada a 1024 bits. En estas circunstancias usando una matriz BG1 se tendrá un $Z_c = 24$ y una palabra código de longitud 1056 bits y usando una matriz BG2 se tendrá un $Z_c = 52$ y una palabra código de longitud 1040 bits.
- La codificación NR-Polar se realiza mediante el uso de una secuencia de confiabilidad Q definida por el Reporte Técnico TS 38.212 [2]. Este tipo de codificación se denomina código secuencial, ya que utiliza matrices de transformación polar que se consideran como núcleos o *Kernel*, Para el presente proyecto, en donde se considera el enlace de uplink, la longitud de la palabra código en NR-Polar es de 1024 con una tasa de codificación $R \approx 1/2$, ya que 512 bits son denominados de “control”, 500 bits de información y 12 bits para el polinomio generador de código de redundancia cíclica (*Cyclic Redundancy Codes* -CRC por sus siglas en inglés-) [3].

Para ambos tipos de codificación, la palabra código obtenida será aproximadamente de 1024 bits y previo a ser transmitida será modulada con BPSK. El motivo por el cual se escogió este esquema de modulación se debe a que es uno de los más robustos, por lo

tanto, la transmisión no será significativamente vulnerable respecto a las anomalías presentes en el canal inalámbrico.

Por su parte el receptor deberá realizar las operaciones de cuantificación de punto fijo (*fixed-point quantization*), de decodificación y decisión.

- El proceso de cuantización de punto fijo (*fixed-point quantization*) tiene como objetivo transformar un vector de valor real, proveniente del canal AWGN, a una variable integral (*integer*). Esto se lo hace porque el decodificador no puede aceptar valores reales sino solamente un valor definido en n bits, por lo que este bloque prepara la información proveniente del canal para posteriormente entregarlo al decodificador. Nótese que este proceso reemplaza al que, tradicionalmente, sería el bloque de demodulación BPSK. Esto es debido a que el decodificador que se usará es *soft* o de distancia euclidiana, mismo que a diferencia de los decodificadores tradicionales que son del tipo *hard* o de distancia de Hamming, se caracterizan por trabajar con todos los niveles de energía tanto en su entrada como salida y no solamente con dos valores binarios (0 o 1) [4], [5]. Por este motivo, el trabajar con un demodulador tradicional queda obsoleto para el sistema que se intenta implementar y, en consecuencia, se deba acudir a este concepto [6]. Para el presente trabajo se usará un valor de $n = 5$ bits.
- La decodificación NR-LDPC manejará un proceso de decodificación SISO MINSUM, mismo que es un proceso de decodificación tipo iterativo que puede representarse de mejor forma a través de un diagrama de la Figura 1.2.

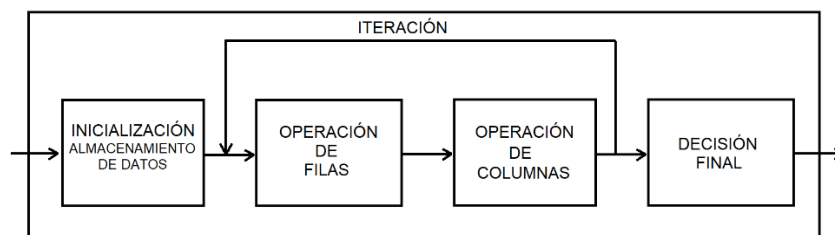


Figura 1.2. Canal Genérico de Telecomunicaciones

La operación de filas involucra un proceso de decodificación *SISO MINSUM Single Parity Check (SISO SPC MINSUM Decoder)*, mientras que la operación de columnas es un proceso de decodificación *SISO Repetition Code* [7]. El número de iteraciones que será usado será de 8. Además, el proceso de decodificación incorporará características de *Rate Matching, Layered* y *Offset* [8] que permiten mejorar el desempeño de los decodificadores en términos de BER.

- La decodificación NR-POLAR por su lado utilizará un decodificador tipo SISO de cancelación sucesiva simplificada con listas (*Sucesive Simplified Canceled Decodification with List* - SSCL por sus siglas en inglés) [9]. Se denomina de “cancelación sucesiva” porque la decodificación es secuencial y “simplificada” porque utiliza el concepto de un árbol binario y con listas porque se utilizarán códigos de redundancia cíclica que permiten generar una M cantidad de palabras código candidatas, las cuales posteriormente serán evaluadas en un bloque de verificación CRC. La palabra código candidata que tenga el CRC original será la seleccionada. Para este caso, que se trabajará en uplink, el estándar especifica que el código debe incorporar en la parte del codificador un bloque que permita generar un CRC con un polinomio de 12 bits y reducir el mensaje solamente a 500 bits [3].

Al final de los procesos de codificación, modulación, transmisión a través del canal, decodificación demodulación se contará el número de errores a nivel de bits (BER) como de bloques (FER) que se envíen y los tiempos de procesamiento tanto de codificación por bloque, decodificación por bloque y total. Estos resultados serán almacenados para ser verificados con estudios similares para asegurar el correcto funcionamiento de los sistemas implementados y para luego presentarlos por medio de gráficas y tablas para su análisis.

Los resultados de la simulación se podrán visualizar en gráficas BER Vs. Eb/No, FER Vs. Eb/No, BER Vs. Número de Bloques Enviados, tiempo de procesamiento Vs. Número de Bloques Enviados, tiempo promedio de codificación por bloque Vs. número de bloques enviados y tiempo promedio de decodificación por bloque Vs. número de bloques enviados como así también tablas. Finalmente, se hará un análisis de los resultados obtenidos de ambos esquemas de codificación, tomando en cuenta los tres escenarios de aplicación definidos en el reporte técnico TR 38.913 – V15.0.0 que son: Ancho de Banda Móvil Avanzado (eMBB), Comunicaciones Confiables de Ultra Baja Latencia (URLLC) y Comunicaciones Masivas tipo Máquina (mMTC) [10].

Los resultados consistirán en gráficas y tablas obtenidas de los datos de las simulaciones que serán analizadas, por esto, este trabajo de titulación no contiene un producto final demostrable.

1.3.1 INTRODUCCIÓN A LOS SISTEMAS DE COMUNICACIONES INALÁMBRICAS

Un sistema de comunicación es el conjunto de recursos en espectro, espacio, tiempo y equipos, necesarios para realizar una comunicación a corta o larga distancia. En este sentido, el canal de radio comprende el medio de transporte, es decir, el vacío o el aire, que permite que la energía electromagnética generada en el transmisor sea radiada por medio de una antena al medio de transmisión y transportada hasta el receptor sin conexión física entre éste y el transmisor [11]. La porción del canal de radio que comprende sólo el medio de transporte, es decir, el vacío, el aire u otro medio material en el que se propaga la energía electromagnética, suele designarse como canal de propagación en cuyo caso no se incluyen ni las antenas ni las líneas de transmisión [11].

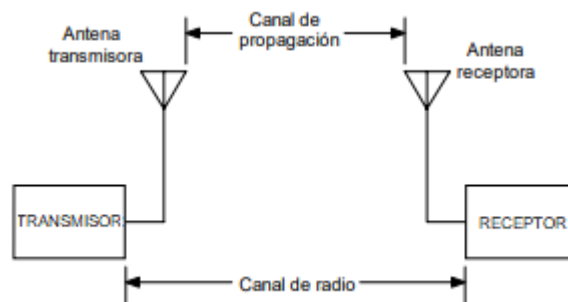


Figura 1.3. Canal de radio y canal de propagación

Canal AWGN (*Additive White Gaussian Noise*)

Es el canal de propagación que, como su nombre lo dice, contiene ruido blanco aditivo de distribución gaussiana. Al ser de naturaleza aleatoria, el ruido AWGN es probabilístico y puede ser descrito como una variable n de naturaleza aleatoria, con promedio $u = 0$ y varianza σ^2 tal que $n \sim N(0, \sigma^2)$ [12]. Por lo tanto, la Función de Densidad de Potencia o PDF por sus siglas en inglés (*Power Density Function*) de un canal tipo AWGN puede ser descrito matemáticamente de la siguiente manera.

$$PDF = F_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x-u}{\sigma^2}} \quad (1.1)$$

$$PDF_{AWGN} = F_N(n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x}{\sigma^2}} \quad (1.2)$$

La ecuación 1.3. representa la Función de Densidad de Potencia de una distribución gaussiana con promedio u y con varianza σ^2 . Mientras tanto, la ecuación 1.2 representa la

Función de Densidad de Potencia de un canal AWGN. La Figura 1.4. muestra el espectro de densidad de potencia característico del ruido AWGN.

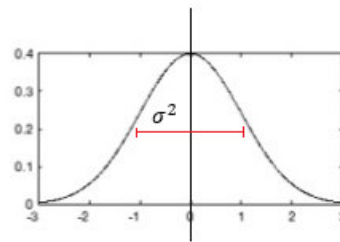


Figura 1.4. Distribución de la función de probabilidad de densidad de potencia del ruido AWGN

Desde el punto de vista de un sistema de comunicaciones, tal como se muestra en la Figura 1.5, el canal AWGN puede ser caracterizado con señales discretas como una sumatoria entre la señal transmitida s y la señal aleatoria de ruido AWGN n .

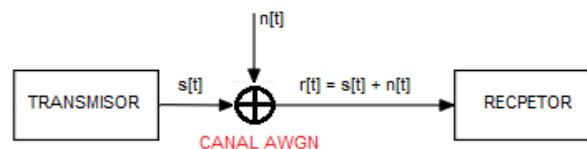


Figura 1.5. Canal AWGN con señales discretas

Tal como puede apreciarse en la figura, el ruido al ser de naturaleza aleatoria y aditiva afectará las señales que serán transmitidas, haciendo que difieran de las que serán recibidas. La medida de cómo el ruido AWGN afecta a la señal s está en función de la varianza σ^2 porque este parámetro es también conocido como la desviación de potencia de ruido (en textos técnicos en inglés este concepto es conocido como *Noise Power*). Así, si la varianza aumenta, la variación de la potencia del ruido n será mayor y la señal recibida r tiene mayores probabilidades que difiera del símbolo transmitido. Este es el motivo por lo cual se han ideado métodos para medir el rendimiento de un sistema de comunicación, tales como el BER, SNR y el E_b/N_0 , y también se han ideado métodos para la corrección de los errores como la codificación de canal. Estos conceptos se analizarán en detalle más adelante.

1.3.2 MODULACIÓN

Este proceso comprende, tal como se muestra en la Figura 1.6, desde la salida del modulador en el transmisor hasta la entrada del demodulador en el receptor [13].



Figura 1.6. Canal de Radiofrecuencia

En la Figura 1.6 el bloque de modulación transforma una señal de información a en una señal modulada s . Debido al ruido introducido por el canal AWGN, la señal en la recepción debe considerarse una estimación y previo a ser demodulada pasa primero por un Dispositivo de Decisión que por medio de reglas ajusta la señal a los niveles de potencia necesarios para que el demodulador pueda transformar la señal modulada con ruido estimada \hat{r} en información estimada \hat{a} sin que el dispositivo sufra daños.

Modulación digital

Permite transmitir una señal a digitalizada que está en banda base en una señal s en radiofrecuencia. Es decir, transmite una señal digital de datos a con el uso de una portadora analógica que varía uno de sus parámetros (amplitud, frecuencia o fase). La ecuación 1.3 define matemáticamente a la portadora modulada analógica.

$$X_C = A_C \cos(\omega_C t + \theta) \quad (1.3)$$

Donde A_C es la amplitud, ω_C es la frecuencia angular que está definida en $[Hz]$ y θ es la fase en $[rad]$

Modulación digital PSK (*Phase Shift Keying*)

Es una técnica de modulación digital que permite transmitir una señal digital con el uso de una portadora analógica que varía su fase. La ecuación 1.4. describe a este esquema de la siguiente manera.

$$s(t) = \cos\left(\omega_C t + \frac{a_n(t) \cdot \Delta\theta}{2}\right) \quad (1.4)$$

Donde $s(t)$ es la señal modulada en fase que está en función de ω_C que es la frecuencia angular de la portadora en $[Hz]$. La señal portadora se modula de tal forma que se separa

entre fases de señales adyacentes $\Delta\theta = \frac{2\pi}{M}$ [rad]. El número de fases es determinado por $M = 2^n$, donde n es el número de bits transmitidos por símbolo modulado.

Alfabeto de modulación de fase

Es el conjunto de símbolos de cada tipo de modulación M-PSK y está en función del número de fases M y número de bits transmitidos por símbolo modulado n , tal como se muestra en la Tabla 1.1 [14].

Tabla 1.1. Abecedario para esquema de modulación PSK

Esquema de Modulación PSK	n	M
BPSK	1	2
QPSK	2	4
8-PSK	3	8

Así, en el caso que el esquema de la modulación sea binario o BPSK, el abecedario está compuesto de dos símbolos, de donde se transmite hasta $n = 1$ bits por símbolo.

Modulación BPSK (*Binary Phase Shift Keying*)

Si la señal que se transmite solamente permite transmitir un bit por símbolo, existirán $M = 2$ fases con una separación de $\Delta\theta = \pi$ [rad] entre cada una de ellas. Este tipo de modulación se denomina BPSK y se parte en base a la ecuación 1.5. para obtener la siguiente expresión:

$$s(t) = \cos\left(\omega_c t + \frac{a_1(t)\pi}{2}\right) \quad (1.5)$$

De la ecuación 1.5. nótese que la señal $a_1(t)$ a modular se limita a ser binaria (0 o 1). Esto es un dato importante porque permite caracterizar al esquema de modulación desde el enfoque de la potencia [15]. Considerando que la potencia promedio de transmisión es de P [w], entonces un símbolo s que será transmitido a través del canal AWGN puede tener la siguiente potencia.

$$s = \begin{cases} 0 \rightarrow P \\ 1 \rightarrow -P \end{cases} \quad (1.6)$$

Cuando la señal transmitida pasa por el medio de propagación (en este caso el canal AWGN) y llega al receptor, el dispositivo de decisión detecta la señal y estima el símbolo

transmitido en función del valor de potencia con la que llega la señal. Este es uno de los métodos de detección que se denomina detector basado en un umbral (*Threshold Based Detector*) y se describe a través de la ecuación 1.7 en términos de la señal recibida [16].

$$\hat{s} = \begin{cases} 0 & \text{sí } r \geq 0 \\ 1 & \text{sí } r < 0 \end{cases} \quad (1.7)$$

1.3.3 RENDIMIENTO DE UN SISTEMA DE COMUNICACIÓN INALÁMBRICO

Ahora, considere el sistema de comunicación inalámbrico con modulación BPSK sobre un canal AWGN en el dominio discreto del tiempo, tal como puede verse en la Figura 1.7.

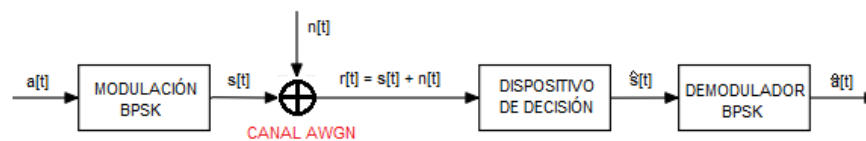


Figura 1.7. Modulación BPSK sobre canal AWGN

Considere un ejemplo donde se envían dos bits de información (0 y 1) para un instante de tiempo uno y dos, respectivamente, y que la potencia promedio de transmisión es $P = 1[w]$. Basados en la ecuación 1.6, los resultados de esta transmisión se muestran en la Tabla 1.2.

Tabla 1.2 Ejemplo de transmisión de datos en un sistema de comunicación con modulación BPSK sobre canal AWGN de dos bits de información en dos instantes de tiempo discreto

t	$a[t]$	$s[t]$	$r[t]$	$\hat{s}[t]$	$\hat{a}[t]$
1	0	+1	0.8	+1	0
2	1	-1	0.2	+1	0

Tal como puede apreciarse en la Tabla 1.2, para el instante de tiempo número dos, el bit de información transmitido a difiere del estimado \hat{a} , debido a que el ruido aleatorio introducido por el canal AWGN afectó al símbolo transmitido, ocasionando que el símbolo recibido sea erróneo. De hecho, tal como se ha venido mencionando, es normal que en canales inalámbricos los bits transmitidos difieran de los recibidos por fenómenos propios del canal como son la interferencia, distorsión o en este caso el ruido AWGN, motivo por el cual el error puede ocurrir y existen métodos para caracterizarlo matemáticamente.

Tasa de error de bit (BER - *Bit Error Rate*)

La tasa de error de bit es una métrica usada para caracterizar el error dentro de un sistema de comunicación [17]. Matemáticamente es la tasa promedio de error de bit y puede describirse de la siguiente manera.

$$BER = \frac{\# \text{ de bits errados}}{\# \text{ de bits transmitidos}} = \frac{n_e}{N} \quad (1.8)$$

Pese a eso el BER también puede ser expresado como una probabilidad de error que un bit de información a no sea igual a \hat{a} . El BER está en función del esquema de modulación y de las características del canal de propagación [18].

$$BER = \Pr (a \neq \hat{a}) \quad (1.9)$$

Considerando que se recibe del canal modulado la señal r tal como se puede observar en la Figura 1.7, entonces la probabilidad descrita en la ecuación 1.10 puede ser descrita de la siguiente manera.

$$BER = \Pr (r \geq 0) \quad (1.10)$$

Relación Señal-Ruido (SNR – *Signal Noise Relationship*)

Es una métrica usada para caracterizar la intensidad de la señal recibida con respecto al ruido del sistema [19]. Tal como su nombre lo indica y se lo describe en la ecuación 1.11, relaciona la potencia de la señal deseada transmitida con respecto a la potencia del ruido del canal.

$$SNR = \frac{\text{Potencia de transmisión}}{\text{Potencia del ruido}} \quad (1.11)$$

Para un análisis matemático más detallado de este concepto se debe considerar a un sistema de comunicación inalámbrico desde los enfoques del dominio continuo y discreto del tiempo, descritos por la Figura 1.5. En primer lugar, desde el enfoque del dominio continuo del tiempo considere que el sistema de comunicación tiene una potencia promedio de transmisión de la señal de P [w] y que el ruido AWGN tiene una densidad espectral de potencia de ruido de $PDS = N_o/2$ ¹ y que la señal tiene un ancho de banda de $2W$ [20].

$$P_N = \frac{N_o}{2} \cdot 2W = N_o W \quad (1.12)$$

¹ N_o para este trabajo de titulación se lo define como la densidad espectral de potencia de ruido. Densidad Espectral (*Spectral Density*) de una señal es una función matemática que nos informa de cómo está distribuida la potencia o la energía

Bajo estas consideraciones, la ecuación 1.13. describe matemáticamente el SNR de una señal continua basado en los parámetros descritos anteriormente.

$$SNR = \frac{P}{N_o W} \quad (1.13)$$

En segundo lugar, para el análisis del SNR en tiempo discreto, nótese que basados en las consideraciones hechas anteriormente el tiempo por símbolo en la señal es $T = 1/2W$. Además, recuérdese que cuando se trabaja con señales en el dominio discreto del tiempo, se habla de señales de energía. Bajo estas consideraciones, se tiene que la energía por símbolo será:

$$E_s = P \cdot T = \frac{P}{2W} \quad (1.14)$$

La energía del ruido AWGN en este caso viene dada por la varianza del ruido $\sigma^2 = N_o/2$ y, por lo tanto, se tiene que $P_N = \sigma^2$ [21]. La ecuación 1.15. describe y hace caer en cuenta que, para ambos escenarios, tanto continuo como discreto, la relación señal a ruido en un sistema de comunicaciones es la misma. Observe que, en función de las distintas ecuaciones que se han podido describir, la ecuación que describe el SNR de un sistema con un canal con ruido AWGN puede escribirse de varias formas.

$$SNR = \frac{E_s}{P_n} = \frac{E_s}{\sigma^2} = \frac{P}{N_o W} \quad (1.15)$$

TASA DE ERROR DE BIT EN BPSK

Sea el sistema de comunicación de modulación BPSK sobre un canal AWGN como el propuesto en la Figura 1.7, se tiene que el símbolo a transmitir s por el canal AWGN tiene un valor binario de $s = 1$ y una potencia de $-P$. Recuérdese que por lo mencionado en la ecuación 1.9 y 1.10, el BER puede ser expresado como una probabilidad de error de que un bit de información a no sea igual a \hat{a} , es decir que se debe cumplir que $r \geq 0$ [18]. Bajo lo expuesto anteriormente, si se recibe un símbolo r , mismo que es la suma del símbolo transmitido s con el del ruido AWGN, la probabilidad mencionada puede ser descrita de la siguiente manera:

$$r = s + n \quad (1.16)$$

$$r = -P + n \quad (1.17)$$

$$\exists! \text{ error ssí } r \geq 0 \rightarrow -P + n = 0 \quad (1.18)$$

$$\therefore \text{Error de probabilidad} = \Pr(n \geq P) \quad (1.19)$$

Al trabajar con probabilidades, la ecuación 1.18 puede ser resuelta de la siguiente forma:

$$\Pr(n \geq P) = \int_P^{\infty} F_N(n) dn \quad (1.20)$$

Donde F_N es la función de distribución de potencia del ruido, misma que viene dada en la ecuación 1.2. Reemplazando la ecuación mencionada en la ecuación 1.20 se tiene:

$$\Pr(n \geq P) = \int_P^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n^2}{2\sigma^2}} dn \quad (1.21)$$

La solución a la integral puede realizarse mediante el cambio de variable, considere para dicho cambio que $t = \frac{n}{\sigma}$, por lo tanto, $dn = \sigma dt$.

$$\Pr(n \geq P) = \int_{P/\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \quad (1.22)$$

Caiga en cuenta que la expresión $\frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$ tiene la forma de una variable aleatoria gaussiana de promedio cero y varianza de valor uno, tal que puede ser expresado como $N(0,1)$. Si se expresa a la integral propuesta en la ecuación 1.22 como una probabilidad, se tendrá la siguiente expresión:

$$\int_{\sqrt{P}/\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \Pr\left(N(0,1) \geq \sqrt{\frac{P}{\sigma^2}}\right) \quad (1.23)$$

Observe que, la expresión obtenida en la ecuación 1.23 es también conocida en el campo de la probabilidad como función Q [22]. Recuérdese que en la ecuación 1.15 se menciona que la relación P/σ^2 es la relación señal a ruido del sistema de comunicación (SNR). Bajo estas dos consideraciones, el BER para un sistema de comunicación con modulación digital BPSK sobre un canal AWGN será:

$$\Pr\left(N(0,1) \geq \sqrt{\frac{P}{\sigma^2}}\right) = Q\left(\sqrt{\frac{P}{\sigma^2}}\right) = Q(\sqrt{SNR}) \quad (1.24)$$

FUNCIÓN DE ERROR COMPLEMENTARIA

El problema de la Función Q es su complejidad de implementación a través de codificación en MATLAB. Por este motivo, en el presente trabajo de titulación se trabajará con una aproximación de dicha función que es más fácil de implementarla y que es conocida como Función de Error Complementaria [23]. La ecuación 1.25 describe matemáticamente a dicha función para una variable X_0 .

$$\Pr\left(N(0,1) \geq \sqrt{\frac{P}{\sigma^2}}\right) = Q\left(\sqrt{\frac{P}{\sigma^2}}\right) = Q(\sqrt{SNR}) \quad (1.25)$$

Nótese la similitud que existe entre la ecuación 1.25 y 1.24 si $X_o = \frac{x}{\sqrt{2}}$. Por lo tanto, si se reemplaza este término en la primera Ecuación mencionada, la ecuación 1.26 es una aproximación de la Función Q en términos de la Función de Error Complementaria de Gauss.

$$Q(X_o) = 0.5 \text{ERFC}\left(\frac{X_o}{\sqrt{2}}\right) \quad (1.26)$$

En este caso, ERFC hace mención del uso de las iniciales de la función escrita en inglés (*Complementary Error Function*). Como ejemplo, considere que el SNR de un sistema de comunicaciones que tiene modulación BPSK sobre un canal AWGN es de 10[dB]. En este caso si se realiza el cálculo con la función Q , la probabilidad de error en este sistema de comunicación será de 7.82×10^{-4} , mientras que en términos de la aproximación hecha con la función de error complementaria será 3.91×10^{-4} . El valor difiere, sin embargo, la implementación se facilita con la aproximación mencionada.

1.3.4 CODIFICACIÓN DE CANAL

Definición

Tal como se mencionó anteriormente, la distorsión, interferencia o el ruido de un canal inalámbrico hace que los símbolos transmitidos difieran de los recibidos en un sistema de comunicación y de la misma manera como hay formas para caracterizar la ocurrencia del error en el mencionado sistema, también hay métodos para poder detectarlos y corregirlos a través del concepto de codificación de canal [24].

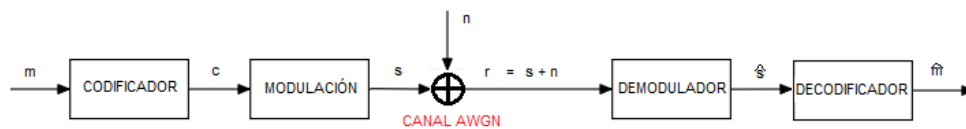


Figura 1.8. Sistema de comunicación genérico con modulación BPSK y codificación sobre un canal AWGN

Ahora, tal como puede apreciarse en la Figura 1.8, un bloque de codificación de canal (k, n) es aquel que toma un mensaje m de k bits y forma una palabra código c de n bits con la forma $c = [m p]$ [25], [26]. Los $n - k$ bits que complementan a la palabra código y acompañan al mensaje son conocidos como bits de paridad p , mismos que permiten detectar y corregir los errores introducidos, en este caso por el canal AWGN.

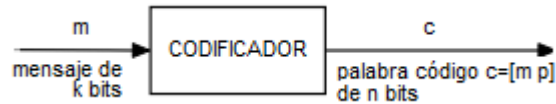


Figura 1.9. Sistema de codificación

Pese a eso, en la práctica, el uso de codificadores de canal no asegura que siempre haya una corrección total de los errores introducidos por el canal, motivo por el cual, la tasa de error de bits disminuirá considerablemente pero jamás será cero. Otro punto para tomar en cuenta es sobre el rendimiento del sistema de comunicación, porque si bien la señal transmitida se robustece y tiene menos probabilidades de error en la recepción debido a la introducción de los bits de paridad, estos mismos bits hacen que el sistema de comunicación necesite mayor tiempo de procesamiento para detectar y corregir un error, en caso de que existiese, por lo que, el mencionado rendimiento del sistema disminuye [26]. Estas son consideraciones que deben tomarse en cuenta a la hora del diseño de un sistema de comunicación e implementación de un esquema de codificación óptimo.

Tasa de codificación

Para un bloque de codificación con notación (n, k) , se define la tasa de codificación como la relación adimensional de valor menor o igual a uno entre la longitud del mensaje de k bits y la longitud de la palabra código de n bits [26], [27]. Matemáticamente es notada con la letra R y se describe por medio de la ecuación 1.27.

$$R = \frac{k}{n} \quad (1.27)$$

Alfabeto de palabras código o codificación

El alfabeto de codificación A es el conjunto de combinaciones posibles entre los k bits del mensaje con los $n - k$ bits de paridad. Cada palabra del alfabeto de codificación es nombrada con la letra B , tal que $B \in A$ [28] y está en función del esquema de codificación seleccionado, y es pertinente mencionar que matemáticamente se calcula usando el número k de bits del que se compone el mensaje.

$$\# \text{ Palabras código} = 2^k \quad (1.28)$$

Relación E_B/N_o

Es la relación entre la energía contenida en un bit E_b con respecto a la densidad espectral del ruido N_o [29]. La energía de bit se define como la relación entre la energía de símbolo y la tasa de codificación R .

$$E_b = \frac{E_s}{R} \quad (1.29)$$

Por su lado, tal como se menciona, la densidad de potencia de ruido es la relación de la potencia del ruido N con respecto al ancho de banda W .

$$N_o = \frac{N}{W} \quad (1.30)$$

Es necesario mencionar que el E_b/N_o está en función de las características de modulación y del canal de un sistema de comunicación [30]. En el caso del sistema propuesto en la Figura 1.8, considere la ecuación 1.15 como punto de partida para determinar la relación E_b/N_o de dicho sistema. Recuérdese que la potencia del ruido dada por la varianza del ruido AWGN σ^2 , también puede ser $\frac{N_o}{2}$. La ecuación 1.32 se obtiene con los cambios realizados.

$$SNR = \frac{E_s}{\sigma^2} = \frac{R \cdot E_b}{\frac{N_o}{2}} \quad (1.31)$$

$$SNR = 2 \cdot R \left(\frac{E_b}{N_o} \right) \quad (1.32)$$

Ahora, recuérdese por conceptos básicos de Comunicación Digital que la energía de los símbolos puede ser calculada como el promedio de los cuadrados de los símbolos [30]. El resultado se muestra en función de la modulación BPSK que está descrita en la ecuación 1.6.

$$E_s = \frac{(-1)^2 + (+1)^2}{2} = 1[W] \quad (1.33)$$

Si se reemplaza este resultado en la ecuación 1.31 se podrá obtener la relación E_b/N_o para un sistema de comunicación con modulación BPSK sobre un canal AWGN y con codificación:

$$\frac{E_b}{N_o} = \frac{1}{2R\sigma^2} \quad (1.34)$$

Note en un primer vistazo que la implementación de un esquema de codificación en un sistema de comunicaciones permite que la potencia de transmisión de un símbolo modulado deba ser mayor a un sistema que no tiene codificación.

Codificación repetitiva (1,3)

Considere como ejemplo de aplicación el esquema de codificación (1,3) o n=3 repetitiva. Basado en la ecuación 1.27, el esquema tiene una tasa de codificación $R = \frac{1}{3}$. La tabla 1.3 muestra como cada bloque del transmisor del sistema de comunicación de la Figura 1.8 actúa sobre las dos posibilidades de mensajes que pueden ser transmitidos sobre el canal AWGN hacia el receptor.

Tabla 1.3. Comportamiento de un mensaje que es transmitido por un transmisor compuesto con un codificador (1,3) y un modulador BPSK

Mensaje (m)	Palabra código (c)	Modulación (s)
0	[0 0 0]	[+1 + 1 + 1]
1	[1 1 1]	[-1 - 1 - 1]

En este caso, los símbolos +1 y -1 representan la señal modulada mediante el uso del diagrama de constelación para el esquema de modulación respectivo [31]. Note que el alfabeto de codificación A está compuesto, en este caso, de 2 palabras código; esto será muy importante más adelante para introducir el concepto de decodificador. Note que si se aplica la ecuación 1.34, el valor de $\frac{Eb}{No}$ para un sistema con este esquema de codificación será igual a $\frac{3}{2\sigma^2}$ y en condiciones sin codificación será de $\frac{1}{2\sigma^2}$. Tal como se había dicho anteriormente, se ha comprobado que cuando se transmite un símbolo codificado, la potencia que se usará para transmitirlo ser mayor.

Tasa de error de bit para un sistema de comunicación con codificación y modulación BPSK sobre un canal AWGN

Recuérdese que se había mencionado que para un sistema con modulación BPSK, la energía de los símbolos es de $1[w]$. Si se reemplaza esto en la ecuación 1.24, se tendrá la siguiente fórmula del BER.

$$BER = Q\left(\sqrt{\frac{1}{\sigma^2}}\right) = Q\left(\frac{1}{\sigma}\right) \quad (1.35)$$

Donde σ es la desviación estándar del ruido AWGN. Ahora, si se reemplaza esto en términos de la función de error complementaria se tendrá la ecuación 1.36.

$$BER = Q\left(\frac{1}{\sigma}\right) = 0.5 \operatorname{ERFC}\left(\frac{1}{\sigma\sqrt{2}}\right) \quad (1.36)$$

En general el cálculo de la varianza y del BER para cualquier sistema de comunicaciones con un esquema de codificación y modulación BPSK viene dado por la ecuación 1.37 y 1.38 [18].

$$\sigma = \sqrt{\frac{1}{2R\frac{Eb}{No}}} \quad (1.37)$$

$$BER = Q\left(\frac{1}{\sigma}\right) = 0.5 \operatorname{ERFC}\left(\sqrt{R\frac{Eb}{No}}\right) \quad (1.38)$$

Si se despeja la variable de la desviación estándar σ de la ecuación 1.38 se concluirá por lo analizado hasta ahora que el BER de un sistema de comunicación con un esquema de codificación usando demodulación BPSK es mayor a uno que no posee codificación y eso se explica por la introducción de los bits de paridad p que hacen que la tasa de codificación R disminuya y con ello, la desviación estándar σ y la potencia del ruido dado por la varianza σ^2 aumenten. Todo lo expuesto anteriormente puede ser visto de mejor manera a través de la gráfica Eb/No vs. BER en la Figura 1.10.

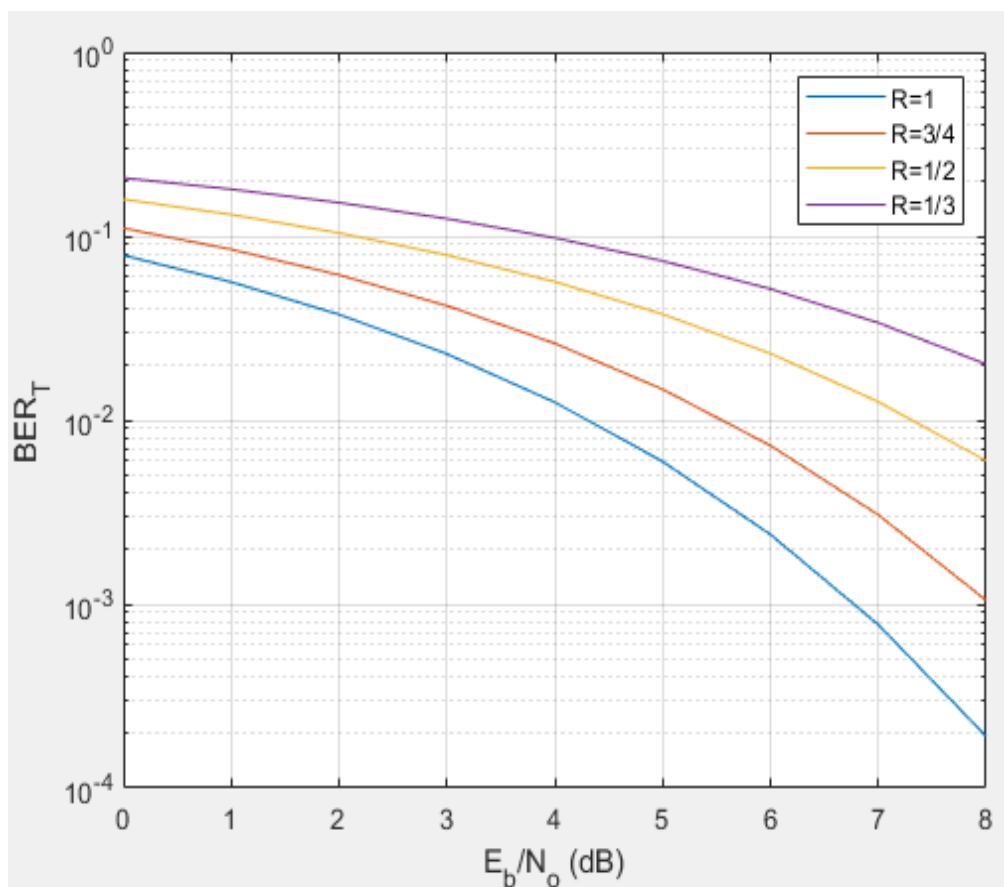


Figura 1.10. BER Teórico para un sistema de comunicación con modulación BPSK y con diferentes valores de tasa de codificación R y de E_b/N_o en dB

Se debe puntualizar que hasta ahora no se toma en cuenta el proceso de decodificación. La pregunta que posiblemente el lector en este momento se plantea es: ¿cómo se puede mejorar el rendimiento del sistema de comunicación manteniendo el bloque de codificación? La solución a la pregunta está en la técnica que usará el bloque decodificador para estimar la palabra código y recuperar el mensaje.

1.3.5 DECODIFICACIÓN DE CANAL

Basado en la Figura 1.8, se puede definir a la decodificación como el proceso inverso a la codificación que tiene como objetivo estimar la palabra código transmitida con el menor número de errores [32]. Para ello, el bloque decodificador usa algoritmos matemáticos que se encargan de detectar y/o corregir los errores introducidos por el canal AWGN en el símbolo transmitido con la ayuda de los bits de paridad p . De hecho, los modelos de decodificación se pueden dividir en dos grupos en función del número de estados que pueden evaluar con la ayuda de la matemática: *hard* o *soft* [33].

Decodificación tipo *Hard Decision*

Tal como se muestra en la Figura 1.11, la decodificación tipo *Hard*, consiste primero en comparar cada muestra de la señal recibida con un umbral o regla de decisión [4], tal como se describe en la ecuación 1.7 para BPSK. Es decir, si el símbolo recibido tiene una potencia mayor al umbral entonces se asigna 0, caso contrario se asigna 1.



Figura 1. 11 Sistema de decodificación Hard Decision

Para el caso de la figura, observe que el umbral o regla de decisión trabaja como demodulador BPSK y convierte el símbolo recibido r , proveniente del canal AWGN, en una palabra código estimada \hat{b} de n bits, donde posteriormente entra al bloque que realiza las operaciones de decodificación. En este caso, cuando existe una coincidencia entre la palabra código estimada y una palabra código contenida en el alfabeto de codificación A se obtiene la palabra código estimada \hat{c} de forma directa, pero, en el caso que no exista coincidencia, se procede a calcular el peso Hamming entre la palabra código \hat{b} con cada uno de los vectores \hat{c} del alfabeto [4]. Matemáticamente, tal como lo describe la ecuación 1.39, el peso o distancia de Hamming puede ser descrita como el número de veces en donde dos vectores de igual dimensión difieren por posición, también conocido como una operación lógica XOR o una suma módulo 2 entre el símbolo \hat{b} y \hat{c} [4], [34].

$$d_H = w(\hat{c} \oplus \hat{b}) \quad (1.39)$$

Donde $w(\hat{b} \oplus \hat{c})$ es conocido como el peso de Hamming. Al finalizar el cálculo, se sustituye la palabra código estimada \hat{b} por aquella del alfabeto que tiene la menor distancia. Si existen dos o más palabras con la misma distancia, el decodificador elegirá una al azar. Matemáticamente, esta relación puede ser descrita por la Ecuación 1.34.

$$\hat{c} = \operatorname{argmin}_{B \in A} [w(\hat{c} \oplus \hat{b})] \quad (1.40)$$

Donde \hat{b} es una palabra código que pertenece al alfabeto de codificación A . Nótese que este método de decodificación si bien es simple en su implementación, aumenta en complejidad y disminuye en rendimiento en función de la longitud del mensaje k que se quiere transmitir, ya que de acuerdo con la ecuación 1.24, si el valor de la longitud del mensaje aumenta, el número de palabras códigos en el diccionario también aumenta y con ello evidentemente el proceso para estimar una palabra código \hat{c} se hace más largo. Para ejemplificar esta situación, considere la implementación de un sistema de comunicación con codificación repetitiva $n=3$ y que utiliza un bloque decodificador *Hard Decision*. La Tabla 1.4 representa los posibles vectores b que pueden ser demodulados en el bloque de demodulación BPSK con su respectiva palabra código y mensaje estimado.

Tabla 1.4. Posibles vectores \hat{b} que pueden ser demodulados en BPSK con sus respectivas palabras código y mensaje estimado para un bloque de decodificación *hard decision* con un esquema de codificación $n=3$ repetitivo

\hat{b}	\hat{c}	$\operatorname{argmin}_{B \in A} [w(\hat{c} \oplus \hat{b})]$	\hat{m}
[0 0 0]	[0 0 0]	<i>No se realiza la operación</i>	0
[0 0 1]	[0 0 0]	<i>Sí se realiza la operación</i>	0
[0 1 0]	[0 0 0]	<i>Sí se realiza la operación</i>	0
[1 0 0]	[0 0 0]	<i>Sí se realiza la operación</i>	0
[0 1 1]	[1 1 1]	<i>Sí se realiza la operación</i>	1
[1 0 1]	[1 1 1]	<i>Sí se realiza la operación</i>	1
[1 1 0]	[1 1 1]	<i>Sí se realiza la operación</i>	1
[1 1 1]	[1 1 1]	<i>No se realiza la operación</i>	1

Así mismo, tome en cuenta que otra desventaja de este tipo de decodificación es que no es óptima porque renuncia a una gran cantidad de información que fue recibida y que es transformada a un solo bit de información cuando pasa previamente por el demodulador BPSK, algo que afectará al rendimiento del sistema en términos de BER. Como ejemplo, considérese la implementación del sistema de comunicación propuesto en la Figura 1.8 con codificación (1,3) en MATLAB. La Figura 1.12 evalúa el rendimiento en términos de BER con diferentes niveles de E_b/N_o .

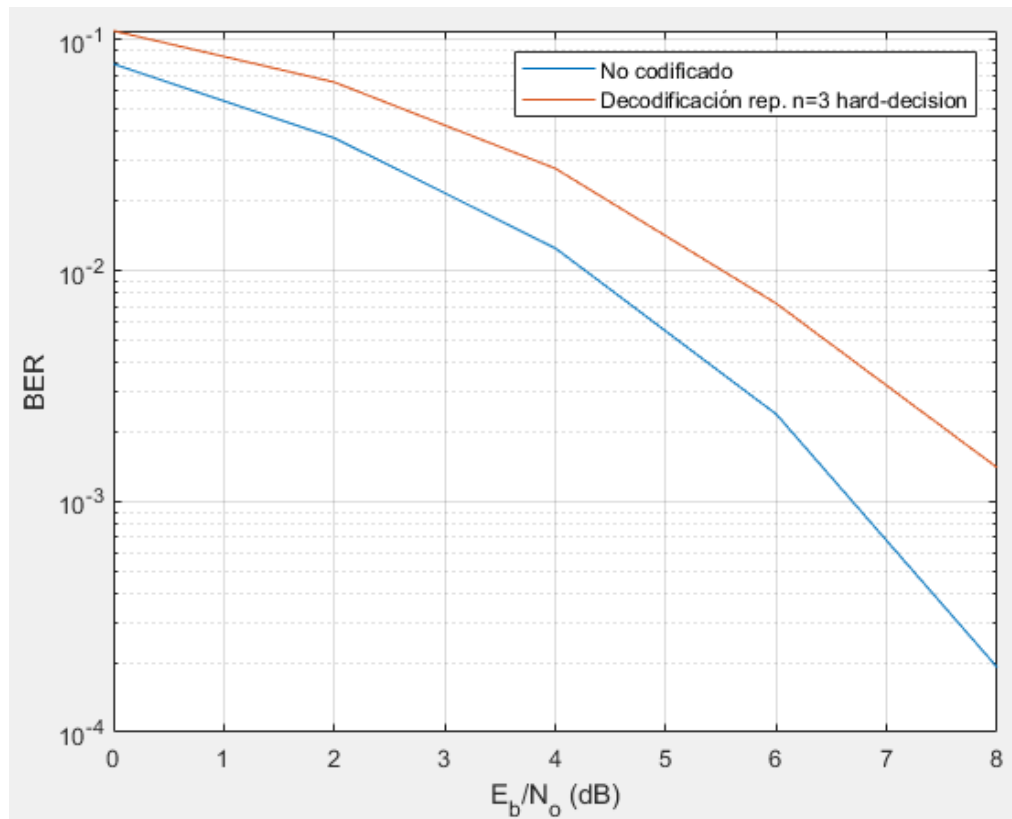


Figura 1.12. Rendimiento de un sistema de comunicación con modulación BPSK, codificación repetitiva (1,3), decodificación tipo *hard decision* sobre un canal AWGN

La Figura 1.12 permite observar que cuando se utiliza decodificación tipo *Hard Decision*, se necesita una mayor potencia de transmisión. Esto es porque, si se nota la similitud que existen entre la Figura 1.10 y 1.12, el esquema de decodificación no introduce ninguna mejora al rendimiento del sistema de comunicación que, en este caso, viene dado por el bloque de codificación a través de la ecuación 1.34.

Decodificadores tipo *Soft Decision*

Recuérdese que una de las grandes desventajas de implementar un decodificador *Hard Decision* es la pérdida de información que experimentan este tipo de decodificadores debido a que, en este caso, el demodulador BPSK transforma la información proveniente del canal AWGN en solamente dos estados binarios. Esto es algo que como se explicó, se ve reflejado en el pobre rendimiento del sistema de comunicación en términos de tasa de bits errados (BER). En consecuencia, para este caso, tal como se describe en la Figura 1.13, la solución más fácil que viene a la mente es crear un tipo de decodificación que trabaje teóricamente con un infinito número de niveles de voltaje² y que al final realice el proceso de decisión o transformación a estados binarios [33]. Este es el concepto detrás de la famosa decodificación tipo *Soft Decision*.

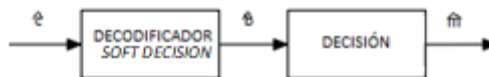


Figura 1.13. Decodificación tipo Soft

Tome en cuenta que, en el caso de la Figura 1.13 la palabra estimada \hat{c} entra en un decodificador tipo *Soft Decision*, es decir que el bloque decodificador recibe un vector con distintos valores o niveles de voltaje y en su salida entregará un vector \hat{b} que es el mensaje estimado, pero con diferentes valores o niveles de voltaje. He de aquí que se necesita y justifica el uso de un bloque de decisión para transformar este mensaje a solamente dos estados binarios (1 o 0). Este último proceso no se lo puede comparar la demodulación BPSK como se lo hacía en los procesos de decodificación *Hard Decision* debido a que este bloque transforma a valores binarios en función de los parámetros de decodificación y esto varía en función del esquema que se vaya a implementar [35].

Ahora, para poner de manifiesto este tipo de decodificación, un decodificador tipo *Soft Decision* es aquel que trabaja con distancia euclidiana. En este caso el decodificador recibe la palabra estimada $\hat{c} = [\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n]$, y compara con la imagen euclidiana de cada una de las palabras código que forman el alfabeto de codificación A . El mensaje estimado \hat{b} será aquel elemento B del alfabeto A que se encuentre a la menor distancia de \hat{c} [5], tal que se cumpla la condición:

$$\hat{b} = \operatorname{argmin}_{B \in A} \{|\hat{c} - (1 - 2B)|^2\} \quad (1.41)$$

² Recuérdese que todos los vectores mencionados en la vida real tienen una representación como ondas de voltaje.

Donde el operador $|\cdot|$ es la norma Euclidiana y $1 - 2B$ representa la imagen Euclidiana de la palabra código B , que es parte del alfabeto de codificación A , esto asumiendo el esquema de codificación BPSK con los niveles descritos en la ecuación 1.6, que son -1 y $+1$. Como ejemplo, considere el esquema de codificación repetitiva $n=3$, mismo que en la Tabla 1.5 tiene las siguientes imágenes euclidianas.

Tabla 1.5. Imágenes Euclidianas de un decodificador *soft decision* para un sistema de codificación $n=3$ repetitivo

B - Palabra del alfabeto	$(1 - 2B)$ o Imagen Euclidiana
[0 0 0]	[+1 + 1 + 1]
[1 1 1]	[-1 - 1 - 1]

Como ejemplo, dada la Figura 1.14, si el decodificador recibe un vector que tiene la forma $\hat{c} = [1.9 - 0.6 - 0.2]$, las operaciones correspondientes serían:

$$|\hat{c} - [1 \ 1 \ 1]|^2 = (0.9^2 + 1.6^2 + 1.2^2) = 4.81 \quad (1.42)$$

$$|\hat{c} - [-1 - 1 - 1]|^2 = (1.9^2 + 0.4^2 + 0.8^2) = 4.41 \quad (1.43)$$

Por lo expuesto anteriormente, el mensaje estimado \hat{b} es aquel elemento de A que se encuentra a la menor distancia de \hat{c} que es $[-1 - 1 - 1]$ y que, a su vez, una vez realizada la decisión, el mensaje estimado \hat{m} será $[0]$. Nótese por el ejemplo propuesto lo mencionado anteriormente que la regla de decisión en este caso depende del parámetro con el cuál se vaya a trabajar los decodificadores, en este caso que es la menor distancia y que el bloque de decisión muchas veces puede estar oculto.

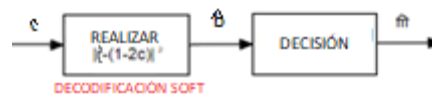


Figura 1.14. Sistema de decodificación por distancia euclidiana

Pero en comparación a un esquema de decodificación tipo *Hard* ¿Cómo cambia el rendimiento del sistema en términos de BER cuando se usa un bloque de decodificación tipo *Soft Decision* [33]. Aquí, considere el ejemplo de la implementación del sistema de comunicación propuesto en la Figura 1.14 con codificación basado en repetición (1,3) en MATLAB y decodificación *soft decision*. La Figura 1.15 evalúa el rendimiento del mencionado sistema en términos de BER con diferentes niveles de E_b/N_0 y su comparación con un sistema de comunicación con decodificación *hard decision* para el mismo esquema de codificación propuesto y sin codificación [36].

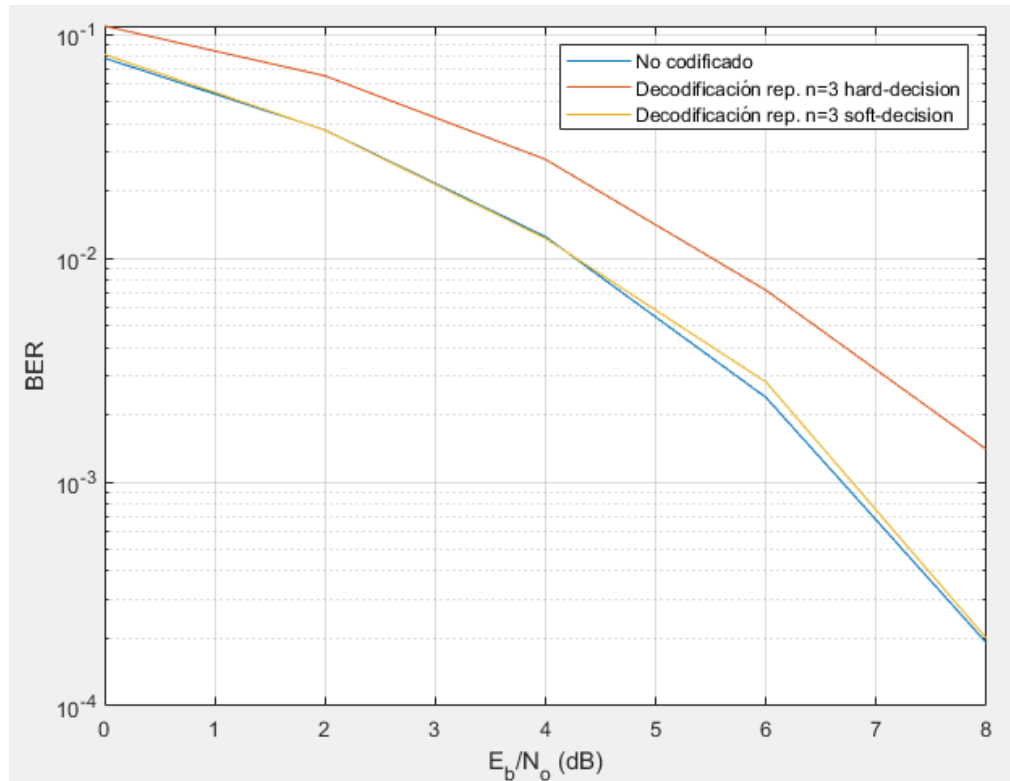


Figura 1.15. Comparativa de rendimientos entre sistemas de comunicación sin codificación, con decodificación *hard decision* y *soft decision*

Observe que, para el sistema de codificación propuesto, el rendimiento con el uso de un esquema de decodificación tipo *soft decision* mejora ampliamente con respecto al tipo *hard decision*. Lamentablemente, la desventaja de la implementación de este tipo de decodificadores es el aumento de la complejidad ya que, para este trabajo de titulación, será necesario el uso de teoría de probabilidad y álgebra lineal, el añadido de iteraciones o el uso de listas dentro de la decodificación para tener mejores estimaciones de la palabra código [33]. Otra desventaja que nace al usar este tipo de decodificación es encontrar reemplazante al proceso de modulación tradicional BPSK que realice la transformación de símbolo a palabra código estimada \hat{c} , pero con la característica que esta esté a distintos niveles de voltaje y no solo en dos estados como se lo ha venido haciendo tradicionalmente. Este tipo de bloques se los denomina cuantizador y será abordado a continuación.

Cuantización de punto fijo (*Fixed-Point Quantization*)

Ahora, con la introducción de los decodificadores tipo *Soft Decision* hay un problema que debe ser solucionado. Tome en cuenta en este caso que el demodulador tradicional BPSK no puede ser usado porque este bloque transforma el vector recibido del canal AWGN en

un vector binario. Una solución que quizás puede pasar rápidamente por la mente es colocar el decodificador de forma directa al canal AWGN. Esto es algo que no puede suceder por tres motivos, el primero es que los vectores que se reciben del canal AWGN son símbolos y no bits y el proceso de decodificación trabaja a nivel de bits. Quizás con modulación BPSK que trabaja con dos estados esto puede sonar irrelevante, pero con otros tipos de codificación tales como QPSK, 16QAM o 64QAM es algo que ya cobra importancia y puede crear fallas en el sistema de comunicaciones que se desee implementar. El segundo motivo por el cual no puede realizarse esta operación es debido a que los vectores que son recibidos del canal AWGN son valores reales, cuando los decodificadores trabajan con valores digitalizados. El tercer motivo se deriva en parte del segundo y es que, en la vida práctica, al trabajar en un mundo digital y consecuentemente con electricidad, todo decodificador, ya sea *Hard* o *Soft Decision*, debe trabajar con rangos de voltaje establecidos, caso contrario el bloque puede sufrir daños. En este sentido, la solución a los problemas expuestos viene dado por un bloque que transforme la información proveniente del canal AWGN a un vector que se encuentre en un ancho de n bits, pero que también, en la mencionada transformación, se asegure proporcionar el valor más exacto posible para una correcta estimación del mensaje en el decodificador. Es por este motivo que en el presente trabajo de titulación se usará una cuantificación de punto fijo (*fixed-point quantization*) [37], que matemáticamente se define por medio de la ecuación 1.44.

$$S_{\text{cuantizado}} = \frac{S_{\text{recibido}}}{\text{Valor pico}} * (2^n - 1) \quad (1.44)$$

Donde n se la conoce como mantisa o ancho de bits y corresponde al número de bits con el cual se cuantificará el valor recibido. Tal como puede apreciarse en la Figura 1.18, el bit más significativo de los n bits es usado para representar el signo del número y el resto se dividen para la parte entera y fraccionaria.



Figura 1.16. Representación de n bits para la cuantización con representación de arreglo de punto

La relación $\frac{S_{\text{recibido}}}{\text{Valor Pico}}$ es conocida como tamaño de paso o resolución, donde *Valor Pico* es el valor en amplitud máxima de la operación de cuantificación. Tal como se puede observar en la Figura 1.19, debido a que la relación debe ser siempre menor o igual a 1, si la información recibida tiene una amplitud mayor al rango de operación del cuantizador, el símbolo recibido por defecto corresponde al valor pico.



Figura 1. 17. Rangos permitidos de operación para cuantificación con representación de punto de arreglo

En consecuencia, tal como se puede ver en la Figura 1.18, el añadimiento de este bloque completa el sistema de comunicación propuesto para el presente trabajo de titulación. Tome en cuenta que el bloque de cuantización reemplaza al de demodulación tradicional y juega un papel importante porque sin él la decodificación tipo *Soft*, que es la que será usada en el presente trabajo, no puede funcionar. Esto es porque, tal como se mencionó anteriormente, este bloque es el que permite transformar los símbolos que provienen del canal AWGN en bits y protege al decodificador de posibles daños producidos por señales que por el ruido propio del canal que pueden producir picos en la señal recibida r . Pero la desventaja de este tipo de sistemas viene dada porque a un mayor número de bits usados para implementar el cuantizador, mayor será la complejidad de este para implementarlo en la vida real. Por este motivo el ancho de bits se recomienda máximo ser entre 4 a 5 [6].



Figura 1.18. Sistema de comunicación a implementar con codificación, modulación BPSK, decodificación *soft-decision* sobre un canal AWGN

1.3.6 CODIFICACIÓN LINEAL DE BLOQUES

Definición

La codificación de bloques se basa en la segmentación sistemática de un mensaje m en bloques de n bits para que se conviertan en palabras código de longitud k por medio de un codificador que añade bits de paridad p y que forman en su conjunto un alfabeto de codificación A [26]. En consecuencia, la codificación lineal de bloques (n, k) es aquella que se basa en un conjunto de n bits independientes, o subespacio ortogonal de dimensión k , que al combinarse de forma lineal (suma módulo 2), generan los bits de paridad p [38]. Para entender de mejor manera este concepto y el desarrollo de algunos otros relacionados

a este tema, considérese como ejemplo un esquema de codificación (6,3) que está representado por la Figura 1.19.

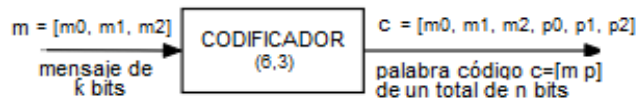


Figura 1.19. Codificación lineal (6,3)

Tal como se menciona en la teoría, cada bit de paridad p se origina de la combinación lineal de los n bits independientes. Dicho esto, se describen las ecuaciones generales para hallar los bits de paridad.

$$p_0 = m_0 + m_1 \quad (1.46)$$

$$p_1 = m_1 + m_2 \quad (1.47)$$

$$p_2 = m_0 + m_2 \quad (1.48)$$

Donde p_1, p_2 y p_3 son los elementos del vector de bits de paridad p y m_1, m_2 y m_3 son los elementos del vector de bits del mensaje m . En este caso, cuando se menciona el concepto de combinación lineal se hace referencia a suma módulo 2; por lo tanto, las ecuaciones antes mostradas también pueden ser reescritas de la siguiente forma:

$$p_0 = m_0 \oplus m_1 \quad (1.49)$$

$$p_1 = m_1 \oplus m_2 \quad (1.50)$$

$$p_2 = m_0 \oplus m_2 \quad (1.51)$$

Matriz generadora

Ahora, considérense las ecuaciones 1.46, 1.47 y 1.48; dichas ecuaciones al ser una combinación lineal pueden ser expresadas en forma matricial, tal como se describe en la ecuación 1.52.

$$[p_0 \ p_1 \ p_2] = [m_0 \ m_1 \ m_2] \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad (1.52)$$

En este caso, la matriz de unos y ceros es la que corresponde a determinar el valor de los bits de paridad p . Por este motivo a dicha matriz se la denominará matriz de paridad P . Ahora, recuérdese de la Figura 1.21 que la palabra código formada por el bloque lineal codificador es de la forma $[m \ p]$. La ecuación 1.53 expresa dicho término en forma matricial basado en los cálculos de la ecuación 1.52.

$$[m_0 \ m_1 \ m_2 \ p_0 \ p_1 \ p_2] = [m_0 \ m_1 \ m_2] \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \quad (1.53)$$

Así, la matriz obtenida de unos y ceros está compuesta por dos partes: la primera que es una matriz identidad que corresponde al mensaje m compuesto por los tres bits de información m_1, m_2 y m_3 que se transmitirán, y la segunda que corresponde a la matriz de paridad P . Dicha matriz es conocida como matriz generadora de rango k , donde k expresa el número de bits que se codificarán en el bloque codificador (n, k) y es conocida así porque contiene la información básica para generar una palabra código lineal de la forma $[m \ p]$ en forma matricial [39], tal como se puede ver en su forma general en la ecuación 1.54.

$$G = [I_k \ P] \quad (1.54)$$

Matriz de verificación de paridad

Dado el esquema de codificación de paridad (6,3), considérese la operación que se realizará con el uso de los elementos de matriz generadora de la palabra código c y la propia palabra código. Para esto, primero se debe tomar en cuenta una matriz compuesta por la matriz transpuesta de la matriz de paridad P y la matriz identidad I_k . Para este caso particular, la matriz de verificación de paridad H de este esquema de codificación viene dada por la ecuación 1.55 [40].

$$H = [P^T \ I_3] \quad (1.55)$$

Hay que recordar que, para este caso, la palabra código tiene la forma extendida $c = [m_0 \ m_1 \ m_2 \ p_0 \ p_1 \ p_2]$.

Tal como lo muestra la ecuación 1.56 y 1.57, si se realiza la operación $H \cdot c^T$ se tendrá como resultado una matriz nula. Esto se da porque, si se recuerda, una de las propiedades de la codificación lineal de bloques es que la combinación lineal entre bits para formar los bits de paridad p pertenecen a un subespacio ortogonal de orden k . Por este motivo, la mencionada multiplicación es la que permite comprobar la ortogonalidad existente en el mencionado subespacio.

$$[P^T \ I_3] \cdot c^T = \bar{0} \quad (1.56)$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} m_0 \\ m_1 \\ m_2 \\ p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (1.57)$$

Tome en cuenta que la matriz de verificación de paridad presente en la ecuación 1.58 tiene una dimensión de 3×6 . De hecho, este caso se puede generalizar y es posible decir que una matriz H es de dimensiones $n - k \times k$ con la siguiente forma:

$$[P^T \ I_k] \cdot c^T = \bar{0} \quad (1.58)$$

Codificación lineal de bloques por medio del uso de la matriz de verificación de paridad

La matriz de verificación de paridad H es útil, ya que a partir de ella algunos esquemas de codificación lineal como LDPC permiten calcular los bits de paridad y así formar la palabra código c [26], [40]. Para esto, observe que la ecuación 1.58 que detalla la verificación de la ortogonalidad para un esquema de codificación (6,3) puede reescribirse en función de los términos de m y p , tal como se describe en la ecuación 1.59.

$$[p^T \ I_3] \cdot \begin{bmatrix} m^T \\ p^T \end{bmatrix} = \bar{0} \quad (1.59)$$

Donde m^T y p^T son las transpuestas del vector mensaje m y paridad p , respectivamente. Si se desarrolla la ecuación 1.59 se tendrá una expresión en función de la suma de la multiplicación de dos matrices.

$$p^T \cdot m^T + p^T = \bar{0} \quad (1.60)$$

Observe que, la operación $p^T \cdot I_3$ se simplifica a solo p^T , ya que es la operación de un vector por una matriz identidad. En este punto, nótese que si despeja el valor de p^T por propiedades lineales se tendrá la ecuación 1.61.

$$p^T = p^T \cdot m^T \quad (1.61)$$

Nótese que, para la ecuación propuesta, tanto P^T como m^T son matrices de valor conocido, ya que se conoce el bloque de mensaje a transmitir. Por lo tanto, para un sistema de comunicación con codificación (6,3), los valores están dados en la ecuación 1.62 y es algo replicable para cualquier otro tipo de codificación lineal.

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} \quad (1.62)$$

1.3.7 INTRODUCCIÓN A LOS ESQUEMAS DE CODIFICACIÓN LDPC Y POLAR PARA SISTEMAS *NEW RADIO* DE QUINTA GENERACIÓN

Desde el final de la década de los años 40 del siglo pasado, varios modelos de codificación de canal han sido modelados e introducidos para sistemas de comunicación inalámbrica, claro está, la aplicación de dichas técnicas depende de la tecnología disponible en el momento. En el caso de la telefonía celular, podemos destacar el uso y estudio de los turbo-códigos [41], que fueron los esquemas de codificación adoptados para los sistemas celulares 3G UMTS [42] y 4G LTE [43]. Actualmente, la Asociación de Tercera Generación (3GPP por sus siglas en inglés), encargado de la definición de estándares de radio celular, se encuentra definiendo las especificaciones de la generación 5G, a la que se le ha denominado *New Radio (NR)*, mismas que se han liberado por medio de versiones. Para propósito e interés de desarrollo de este trabajo, el análisis se centrará en la Versión 15 [44].

Escenarios de aplicación para sistemas de radio de quinta generación e inconvenientes para su implementación

De acuerdo con el reporte técnico TR 38.913 – V15.0.0 emitido por la Asociación 3GPP, se definen tres escenarios de aplicación para la nueva versión del interfaz de radio *New Radio 5G* [45]:

- Ancho de banda móvil avanzado (eMBB): admite acceso de radio en ambientes internos y externos de alto rendimiento y movilidad (hasta 500 km/h) con latencia de plano de usuario de 4 ms para UL y DL.
- Comunicaciones ultra confiables y de baja latencia (URLLC): proporciona un intercambio de datos urgente y confiable para aplicaciones de misión crítica con una latencia de plano de usuario de 0,5 ms en UL y DL [46].
- Comunicaciones masivas de tipo máquina (mMTC): transmisiones masivas y cortas de paquetes infrecuentes para aplicaciones de detección, medición y monitoreo con aplicación especialmente en el Internet de las cosas (con latencia en plano de usuario de 10 s en UL y DL) [47].

Si bien la creación de estos tres escenarios supone un gran avance en cuanto a la búsqueda de establecer comunicaciones de tiempo real en alta calidad para escenarios específicos, también impone una serie de nuevos requisitos igualmente de alta exigencia que hace obligatorio la selección de un modelo de codificación de canal diferente a la

adoptada en 4G-LTE [43]. A continuación, se detallan de mejor manera algunos motivos que obligan a este cambio [26]:

- *Throughput*³: se considera que los turbo-códigos son incapaces de lograr de manera eficiente el rendimiento de procesamiento de múltiples Gbps requerido para eMBB. En este escenario, se busca que la tasa pico sea de 20Gbps, cuando 4G-LTE solo ofrece tasas de transmisión de hasta 1Gbps.
- Zona de cobertura: uno de los problemas que se deriva de la búsqueda de una mayor velocidad de transmisión es que 5G permite lograr una cobertura mejorada con un intercambio de datos más rápido y de manera más confiable.
- Latencia: como se mencionó anteriormente, 5G también se dirige a aplicaciones de comunicación de muy baja latencia confiables (URLLC). Debido a que las aplicaciones que se manejan en este escenario son consideradas de “misión crítica”, tales como la conducción de vehículos autónomos, en donde se debe ofrecer una capacidad significativamente mejorada de corrección de errores y latencia (0,5ms para el plano de usuario en UL y DL), lo que hace que la codificación turbo sea ineficaz.
- Flexibilidad: los escenarios diseñados para 5G tienen en consideración el gran número de servicios y tipos de dispositivos. Por este motivo, las palabras código deben ser flexibles a los servicios que prestan, ya que de esta forma aprovecharán de manera más eficiente el hardware que tienen los equipos, el canal asignado y sus características.

Selección de los estándares de codificación *New Radio* LDPC y Polar para sistemas de radio de quinta generación

Los estudios para establecer los esquemas de codificación para el estándar *New Radio* 5G se extendieron entre abril a noviembre del 2016 [48] y tienen como conclusión el Reporte Técnico TR 138.912 [10], mismo que establece que, el esquema de codificación NR-LDPC se usará para los canales de datos de *New Radio*, mientras que, se seleccionan los códigos

³ *Throughput*: tasa promedio de éxito en la entrega de un mensaje sobre un canal de comunicación. Este dato puede ser entregado sobre un enlace físico o lógico, o a través de un cierto nodo de la red

NR-Polar para los canales de control, reemplazando la codificación turbo-código y convolucional tipo *Tail-Biting* de 4G-LTE, respectivamente.

El motivo para esta selección es debido a que la codificación LDPC soporta de manera eficiente múltiples tasas de código, longitudes de bloque y solicitud de repetición automática híbrida (HARQ por sus siglas en inglés), con una mejor latencia de decodificación, rendimiento y complejidad de implementación que otros códigos [49]. Por otro lado, se seleccionan los códigos polares para los canales de control porque ofrecen la mejor capacidad de corrección de errores en las longitudes de bloque de información cortas, mismas que se utilizan para la información de control [50], [51].

En el año 2018, la 3GPP definió los lineamientos de funcionamiento de los esquemas de codificación LDPC y Polar para los canales de datos y control de NR, lo cual se especificó en el Reporte Técnico TS 38.212 [1]. Sin embargo, el alcance del mencionado Reporte Técnico solo incluye las operaciones de codificación realizadas en una estación base transmisora o equipo de usuario (UE), sin explicación esquemática o ejemplos y sin una discusión de la operación de decodificación realizada dentro de un receptor [52]. Por este motivo, para esta parte, el presente documento se apoyará de investigaciones recientes sobre el tema de decodificación.

1.3.8 CODIFICACIÓN LDPC (LOW DENSITY PARITY CHECK CODES) PARA SISTEMAS DE RADIO NEW RADIO 5G

Introducción a la codificación LDPC

Fueron introducidos por Robert Gallager en el año 1963 [53], pero olvidados debido a las limitaciones tecnológicas de la época, hasta los años 90's cuando fueron reintroducidos por David MacKay y Radford Neal [54], los códigos de paridad de baja densidad (LDPC por sus siglas en inglés) se definen como el esquema de codificación lineal de bloques que utiliza una matriz de verificación de paridad H tipo *sparse*⁴ para determinar los bits de paridad para un mensaje que se desea transmitir [55], un método que se especificó en capítulos previos y que se resume en la ecuación 1.62. Además, se debe mencionar que este tipo de codificación es también conocido, tal como puede verse en la Figura 1.20, como aquel que puede ser representado de forma gráfica a través de grafos de Tanner [56].

⁴ Se denomina matriz tipo *sparse* a aquellas que están compuestas en su gran mayoría por ceros.

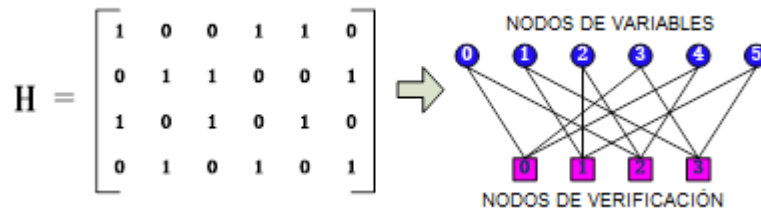


Figura 1.20. Grafo de Tanner para Matriz de Verificación de Paridad H

En este caso, la matriz de verificación binaria H para LDPC debe cumplir la condición dada en la ecuación 1.63 en función de la longitud del mensaje k y de la palabra código n [57].

$$\text{Número de 1's en matriz } H \ll k \cdot (k - n) \tag{1.63}$$

Recuérdese de la ecuación 1.57 y 1.58, que las dimensiones de la matriz de verificación de paridad H son $(n - k) \times k$, por lo tanto, la multiplicación de la ecuación 1.63 corresponde al número total de elementos que existen en la matriz. Debido a que este es un tema muy amplio y que es tratado y detallado a fondo en otro trabajo de titulación [58], el presente documento se centrará en describir las características de la codificación LDPC para *New Radio 5G* [1], por lo que a continuación solamente se tratarán los temas esenciales de LDPC.

Construcción protográfica de una matriz de verificación de paridad

Las matrices de verificación de Paridad H en LDPC para el estándar *New Radio 5G* se construyen por medio de la construcción protográfica⁵ [59], [60]. Este método trabaja con una matriz base B , denominado también grafo base (*base graph*), que a su vez se expande para obtener la matriz de paridad H mediante matrices identidad desplazadas o matrices nulas, un proceso conocido de forma técnica como permutación de desplazamiento a la derecha [57]. Para explicar de mejor manera el proceso de expansión de esta matriz considérese como ejemplo la matriz indicada en la ecuación 1.64.

$$B = \begin{pmatrix} 1 & -1 & 3 & 1 & 0 & -1 \\ 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 4 & 2 & 1 & -1 & 0 \end{pmatrix} \tag{1.64}$$

Note que en el ejemplo dado la matriz base tiene valores entre -1 hasta 4. El número de valores que están dentro del rango mencionado es el que determina el tamaño de expansión de la matriz Z , en este caso $Z = 5$ [61]. A su vez este valor determina la dimensión de la matriz identidad E o nula N con la cual se hará la expansión.

⁵ La codificación protográfica combina la idea de los códigos tipo multilínea con un caso particular de baja complejidad estructural de decodificadores por hardware simplifica el proceso de decodificación paralela por hardware.

Matemáticamente, esto puede ser descrito de forma general por las siguientes expresiones matemáticas:

$$z = \max(B) + 1 \quad (1.65)$$

$$E = I_z \quad (1.66)$$

$$N = \overline{0}_z \quad (1.67)$$

Por lo tanto, para un tamaño de expansión $Z = 5$, se tiene una matriz identidad y nula para realizar la expansión respectiva dada por la ecuación 1.68 y 1.69, respectivamente.

$$E_5 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1.68)$$

$$E_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.69)$$

Ahora, el proceso de expansión para hallar la matriz de verificación de paridad H consiste en reemplazar cada valor de la matriz base B por matrices identidad desplazadas o nulas del tamaño de expansión z en función del valor de cada entrada de la matriz base B [57], [61]. En este caso, si el valor de la entrada de la matriz base B es de -1, entonces la matriz de verificación de paridad en dicha entrada se reemplaza por la matriz nula de dimensión Z , caso contrario, si la entrada de la matriz base B tiene otro valor, la entrada de la matriz de verificación de paridad H en dicha entrada usa la matriz identidad de dimensión Z ; pero desplazada a la derecha el número de posiciones que la entrada de la matriz base B especifica. Matemáticamente esto puede ser descrito de forma general por medio de la ecuación 1.70.

$$H_{ixj} = \begin{cases} \overline{0} & \text{si } B_{ixj} = -1 \\ E_{z+d} & \text{si } B_{ixj} = 0 \end{cases} \quad (1.70)$$

Donde la notación $z + d$ denota un desplazamiento de la matriz identidad de d posiciones a la derecha dada por la entrada ixj de la matriz B [57], [61]. Por ejemplo, note que el valor del elemento que está en la posición $1x2$ de la matriz B de la ecuación 1.63 es -1, por lo tanto, la matriz será nula. Pero, si el valor es 3, como el que se tiene en la entrada $1x3$, entonces el valor de la matriz de verificación de paridad H en dicha entrada es la matriz

identidad de la ecuación 1.66, pero desplazada en 3 posiciones a la derecha, tal como se muestra en la ecuación 1.71.

$$H_{1 \times 3} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (1.71)$$

Como observación, note que el proceso del desplazamiento hacia la derecha es cíclico [59]. Si se realiza el proceso completo con todas las entradas de la matriz se obtendrá la matriz de verificación de paridad H en base a una construcción protográfica [61]. En el caso del ejemplo mostrado con la matriz base B de la ecuación 1.63, el resultado final de la matriz de verificación de paridad H corresponde al de la Figura 1.21 [62].



Figura 1.21. A la izquierda se presenta la matriz/grafico base B de la Ecuación 1.64 y a la derecha su matriz de verificación de paridad H dada

Además, veamos que cuando se trabaja con matrices base, cada entrada de la matriz de verificación de paridad es un bloque; este término será muy usado de forma común de ahora en adelante en el documento.

Factor de expansión Z_c para el estándar *New Radio 5G*

El estándar *New Radio 5G*, define a través de la especificación TS 38.212, el valor del tamaño o factor de expansión que se notará como Z_c [57]. Así mismo, el estándar define igualmente que este valor se determinará en función de unos parámetros que se definen por medio de la ecuación 1.72.

$$Z_c = a \cdot 2^j \quad (1.72)$$

Los parámetros definidos previamente están especificados en el estándar y se encuentran descritos en la Tabla 1.6.

Tabla 1.6. Parámetros para el cálculo del factor o tamaño de expansión Z_c

iL_s	Índice del conjunto de elevación	0, 1, 2, 3, 4, 5, 6, 7
a	Tamaño de expansión	2, 3, 5, 7, 9, ..., 15
j_a	Exponente de expansión	7, 7, 6, 5, 5, 5, 4, 4
Z_c	Factor de expansión	$a \cdot 2^j$

De la Tabla 1.7 hay que manejar ciertas consideraciones para el correcto cálculo de Z_c :

- La primera consideración es que debe notarse que el índice de elevación iL_s está asociado de forma directa con el tamaño de expansión a y con el exponente de expansión j_a , en función al orden que se presenta cada elemento [57]. Por ejemplo, el índice del conjunto de elevación $c = 2$ está en la tercera posición, por lo tanto, de acuerdo con el orden, el tamaño de expansión será $a = 5$ y el exponente de expansión será $j_a = 6$.
- La segunda consideración es el correcto uso de j_a porque si se observa por la ecuación 1.72 que se habla de un exponente j y no de uno j_a . Esto es porque j_a es el máximo valor que puede tener el exponente j . Dicho de forma matemática esto se puede describir mediante la ecuación 1.73.

$$j = 0, 1, \dots, j_a \quad (1.73)$$

Nótese que, a partir de estas dos consideraciones existen 51 valores distintos para Z_c . La Tabla 1.7. recoge dichos valores en función del índice del conjunto de elevación [61].

Tabla 1.7. Valores de tamaño de expansión Z_c en función del índice del conjunto de elevación iL_s

Índice del conjunto de elevación (iL_s)	Z_c
0	2, 4, 8, 16, 32, 64, 128, 256
1	3, 6, 12, 24, 48, 96, 192, 384
2	5, 10, 20, 40, 80, 160, 320
3	7, 14, 28, 56, 112, 224
4	9, 18, 36, 72, 144, 288
5	11, 22, 44, 88, 176, 352
6	13, 26, 52, 104, 208
7	15, 30, 60, 120, 240

Se debe subrayar que esta gran capacidad de seleccionar los valores de tamaño de expansión depende de la longitud del mensaje que se envía cumpliendo uno de los nuevos requisitos básicos para el estándar *New Radio* 5G especificados anteriormente como es la flexibilidad de la palabra código a la hora de ofrecer distintas tasas de codificación [48]. Así mismo, recuérdese que, por lo visto previamente los elementos de la matriz de expansión tendrán valores en un rango entre $[-1; Z_c - 1]$.

Matrices base para codificación NR-LDPC

El estándar *New Radio* 5G a través de la especificación TS 38.212 define el uso de dos matrices bases o grafos base para la codificación LDPC conocidos como BG1 de dimensión 46×68 y BG2 de dimensión 42×52 [57]. Pese a que son de dimensiones distintas, ambas matrices tienen una estructura similar que viene dada por la Figura 1.22.

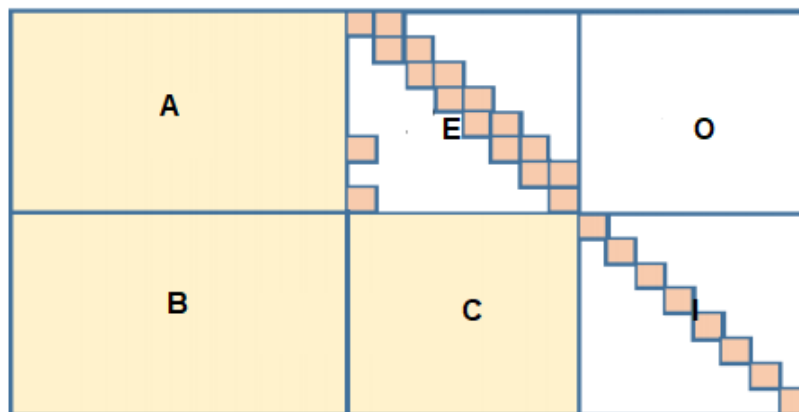


Figura 1.22. Estructura de una matriz base BG1 y BG2 para codificación LDPC en el estándar *New Radio* 5G.

Hay que recalcar que el mismo Reporte Técnico define el valor de las entradas para cada matriz [2]. Dentro de la estructura, cada parte tiene su respectiva función [63], como por ejemplo la submatriz *A* corresponde a los bits sistemáticos o de información. El proceso de codificación se la realiza por medio de la estructura de la doble diagonal que se encuentra en la submatrices *E* para el cálculo de los primeros bits de paridad, mientras que el resto se determina por la submatriz identidad *I* con la ayuda de la submatriz *C*, que con *B* corresponden a las filas para los códigos de verificación de paridad única (*Single Parity Check* en inglés). Por otro lado, la submatriz *O* es de valor nulo. El estándar también define las dimensiones de cada submatriz mencionada, tal como puede apreciarse en la Tabla 1.8 [57].

Tabla 1.8. Dimensionamiento de las partes que forman las matrices base BG1 y BG2 definidos en el estándar New Radio 5G

Parte	BG1	BG2
A	4x22	4x10
E	4x4	38x10
B	42x22	4x4
C	42x4	38x4
O (Matriz Nula)	4x42	4x38
I (Matriz Identidad)	42x42	38x38

Recuerde que, por la teoría revisada, la matriz nula tiene un valor de -1 en sus entradas para una matriz base BG1 o BG2.

Palabra código del esquema de codificación LDPC *New Radio 5G*

El estándar *New Radio 5G* a través de la especificación TS 38.212 menciona que la palabra código se formará en función del grafo base a utilizar (BG1 o BG2) y el valor del tamaño de expansión Z_c [57]. Dicho esto, y tomando de base la nomenclatura usada en el bloque codificador de la Figura 1.9, el número de n bits que tiene la palabra código viene dado por la ecuación 1.74, que es el producto entre el número de columnas de la matriz base y la expansión.

$$n = Z_c \cdot \#columnas_{BG} \quad (1.74)$$

La longitud del mensaje k está en función de los bits sistemáticos, valor que está dado por el número de columnas de la submatriz A . Recuérdese que, por lo visto previamente, cada mensaje de k bits se expande a una matriz cuadrada de dimensión Z_c formando un bloque, por lo tanto, el número de bits del mensaje a transmitir viene dado por la expresión de la ecuación 1.75.

$$k = k_b \cdot Z_c \quad (1.75)$$

Donde k_b es el número de las columnas o bits sistemáticos. Dicho esto, el valor de la longitud de los bits de paridad viene dado por la ecuación 1.76 y el valor de la tasa de codificación está dado por la ecuación 1.27.

$$\text{bits de paridad} = n - k \quad (1.76)$$

Rate matching usando puntuación (*Puncturing*) para transmisión de la palabra código

Se denomina *Rate Matching* al conjunto de técnicas que se utilizan para hacer a un mismo código flexible en términos de tasa de codificación R [63]. Si bien ya se ha definido, la forma como se constituye la palabra código usando codificación LDPC para *New Radio 5G*, el estándar define claramente la forma como debe transmitirse la palabra código. El mismo menciona que la palabra código c no debe ser transmitida con sus dos primeras columnas porque ambas son puntuadas.

Se define a la puntuación (*puncturing*) como la técnica de *Rate Matching* donde a las columnas se le asigna un valor de alto peso, mismo que de acuerdo con la nomenclatura usada es de -1, y expandido es igual a una matriz nula cuadrada de dimensión Z_c [8]. Para ejemplificar esta situación, considérese que se transmite un mensaje codificado con un esquema de codificación NR-LDPC que utiliza la matriz base BG1 (46×68) y un tamaño de expansión de Z_c . Para este caso, en la Figura 1.23 se muestra la forma como debe estar puntuada la matriz BG1 para ser transmitida.

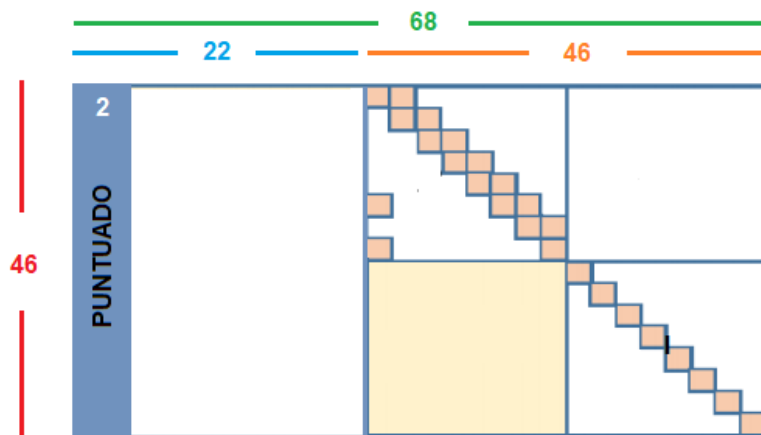


Figura 1.23. Dimensiones de la matriz BG1 usando puntuación (*Puncturing*) y un tamaño de expansión Z_c

Tome en consideración que la puntuación solamente aplica a la palabra código que se transmite, por lo tanto, la cantidad de bits contenidos en la palabra código definidos por la ecuación 1.75 se mantiene. Pero, tal como se menciona, el cambio sí afecta al número de bloques que se transmitirán y en este sentido, la ecuación 1.74 que describe el número de bits totales se usará para caracterizar el número de bits transmitidos, que se expresa por medio de la ecuación 1.77.

$$n = Z_c \cdot \#columnas_{BG} - 2 \quad (1.77)$$

Donde $\#columnas_{BG}$ representa el número de columnas que tiene el grafo base. Así, si se considera el ejemplo, el número de bloques de mensajes para BG1 sería de 22, mientras que la palabra código a transmitir se constituye de 66. De esta manera, si se aplica la ecuación 1.27, se obtiene la tasa más baja de codificación usando una matriz base BG1 y su valor es de $R = \frac{1}{3}$.

RATE MATCHING USANDO ACORTAMIENTO (*SHORTENING*) Y PUNTUACIÓN (*PUNCTURING*)

El acortamiento y la puntuación se utilizan para obtener diferentes tasas de codificación en LDPC para *new radio* 5G. Considérese que ahora, el mensaje que se desea transmitir solamente contiene A bits que están disponibles para ser transmitidos por el canal. Matemáticamente, la ecuación 1.78 describe los bits no usados, si se maneja una matriz base BG1 para la expansión a una tasa de codificación $\frac{1}{3}$.

$$bits\ no\ usados = 22Z_c - A \quad (1.78)$$

Aquí, tómese en consideración que existe un desperdicio de recursos cuando la tasa de codificación es de $\frac{1}{3}$, ya que no es utilizada en un 100%. De hecho, tal como se lo mencionó, uno de los requisitos impuestos por la 3GPP en el estudio de códigos de canal es la flexibilidad para presentar distintas longitudes de la palabra código c [26]. Para esto, LDPC para *New Radio* 5G utiliza *Rate Matching* usando acortamiento en los bloques de mensaje combinado con puntuación en los bloques de paridad [63]. La técnica de acortamiento consiste en que el número de bits que no serán usados, dados por la ecuación 1.78, sea igual a cero, es decir que, sin expandir, la matriz base tenga un valor de -1 [8], tal como puede verse en la Figura 1.24.

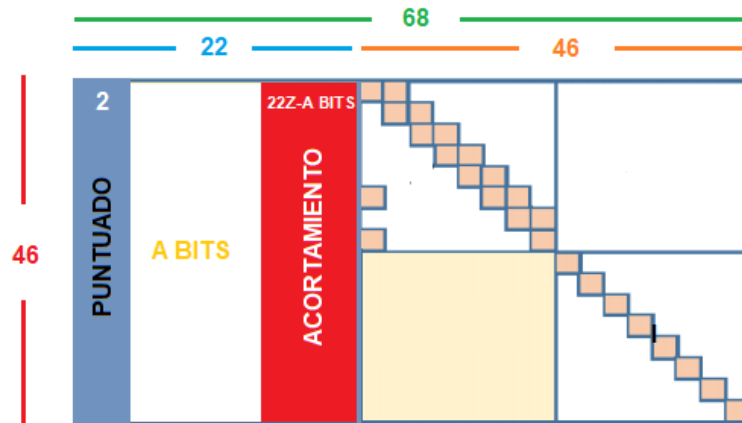


Figura 1.24. Rate Matching con acortamiento en una matriz BG1

Nótese que el número de bits de paridad por medio de esta propiedad también disminuye. Si se tienen E bits a transmitir y considerando que las dos primeras columnas están puntuadas, el número de bits de paridad viene dado por la ecuación 1.79.

$$\text{bits de paridad} = E - (A - 2Z) \quad (1.79)$$

Gráficamente, la Figura 1.25 da una representación gráfica de las dos técnicas combinadas.

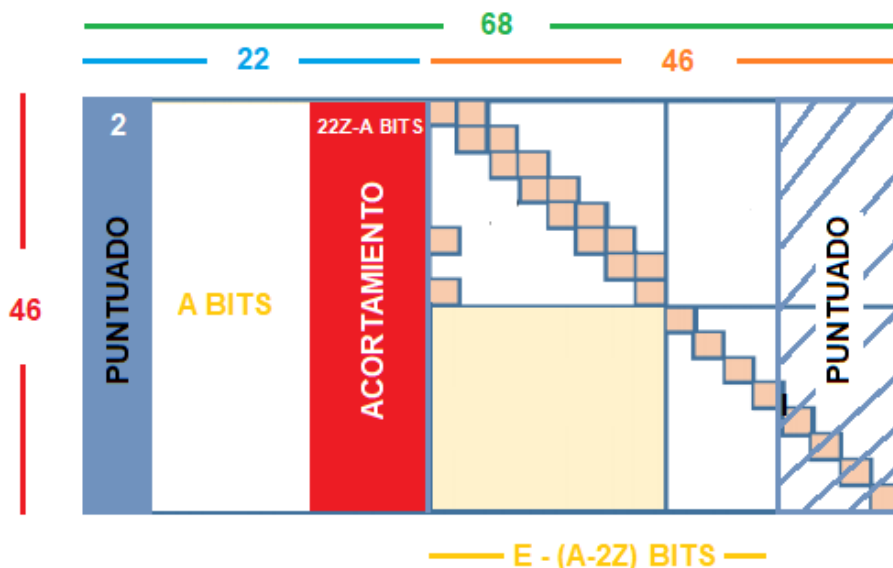


Figura 1.25. Rate Matching con puntuado en una matriz de expansión BG1

De esta forma, obsérvese que se pueden obtener diferentes tasas de codificación y diferentes longitudes para la palabra código c , tal como se describe a continuación [63]:

- La matriz o grafo base BG1, cuyas dimensiones son 46x68, está diseñada para bloques de mensajes entre 300 a 8448 bits, ya que tiene $k_b = 22$ columnas o bits

sistemáticos. En este caso, las tasas de codificación están entre $\frac{1}{3}$ (22 columnas o bits sistemáticos con respecto a 66 columnas a transmitir) y aproximadamente $\frac{8}{9}$ (22 columnas sistemáticas y 25 columnas a transmitir), aunque cabe aclarar que pueden existir algunas tasas más bajas si se utiliza repetición [63].

- La matriz o grafo base BG2, cuyas dimensiones son 42×52 , está diseñada en cambio para bloques de mensajes entre 40 a 3840 bits. Se diferencia de la matriz base BG1 porque el número de bits sistemáticos puede ser variable y puede tener valores tal que $k_b = \{6,8,9,10\}$. Por tanto, las tasas de codificación están entre $\frac{2}{3}$ (6 columnas para bits sistemáticos o de información con respecto a 9 columnas a transmitir) y $\frac{1}{5}$ (8 columnas para bits sistemáticos con respecto a 40 columnas a transmitir) [63].

Cálculo de los bits de paridad para la codificación LDPC *New Radio 5G*

El proceso primero utiliza la estructura de la doble diagonal presente en la submatriz E para el cálculo de los 4 primeros bits de paridad $[p_1, p_2, p_3, p_4]$ [57]. Considérese para la explicación de este paso, una matriz de verificación de paridad H que corresponde a un sistema de codificación (8,4) con un tamaño de expansión $z = 5$.

$$H = \begin{bmatrix} I_1 & 0 & I_3 & I_1 & I_2 & I & 0 & 0 \\ I_2 & I & 0 & I_3 & 0 & I & I & 0 \\ 0 & I_4 & I_2 & I & I_1 & 0 & I & I \\ I_4 & I_1 & I & 0 & I_2 & 0 & 0 & I \end{bmatrix} \quad (1.80)$$

En la codificación las primeras 4 columnas pertenecen a $[m_1, m_2, m_3, m_4]$ y los 4 restantes a los bits de paridad $[p_1, p_2, p_3, p_4]$. Aplicando los conceptos de álgebra lineal vistos previamente sobre subespacios ortogonales de orden k y que concluyeron en la Ecuación 1.57, la matriz puede ser reescrita por medio del sistema de ecuaciones, tal como se describe en la Ecuación 1.60.

$$\begin{cases} I_1 m_1 + I_3 m_3 + I_1 m_4 + I_2 p_1 + I p_2 = 0 \\ I_2 m_1 + I m_2 + I_3 m_4 + I p_2 + I p_3 = 0 \\ I_4 m_2 + I_2 m_3 + I m_4 + I_1 p_3 + I p_4 = 0 \\ I_4 m_1 + I_1 m_2 + I m_3 + I p_4 = 0 \end{cases} \quad (1.81)$$

Recuerde que, por conceptos de álgebra lineal, la doble diagonal cancela todo en cuanto a paridad a excepción del bloque que corresponde a p_1 [38]. De esta manera se conoce el valor de p_1 y si se reemplaza en la ecuación 1, se obtiene p_2 , y así sucesivamente se sigue

el proceso para hallar p_3 y p_4 en la ecuación 2 y 3 del sistema planteado. Este proceso es conocido como sustitución hacia atrás en bloque (*block-wise back-substitution* en inglés).

Una vez calculados los 4 primeros bits de paridad se procede a calcular el resto de los bits y para ello se utiliza la submatriz C y la matriz identidad simple I [57]. Nótese por la Figura 1.22 que, para este paso, cada bit de paridad desde p_5 se calcula en función de los 4 bits de paridad previamente calculados. Además, hay que caer en cuenta que la submatriz C pertenece a las filas de códigos de verificación de paridad simple (*Single Parity Check*). Este concepto es importante entenderlo y se volverá a usar más adelante en el proceso de decodificación.

Introducción a la decodificación LDPC para el estándar *New Radio 5G* - Decodificadores SISO (*Soft Input – Soft Output*) para códigos de repetición

Una vez que se ha visto la forma como se codifica un mensaje usando el esquema LDPC para el estándar *New Radio 5G*, ahora se revisará la forma de decodificar el mismo. Tal como se había dicho en capítulos anteriores, los decodificadores que presentan un mejor rendimiento son los denominados tipo *soft* porque aprovechan todos los niveles de energía, un ejemplo de ello fueron los denominados *soft decision* que trabajan con la distancia euclidiana. Ahora, se revisará un tipo de decodificador *soft* que trabaja con la teoría de probabilidad y que se denominan SISO (*Soft-Input Soft-Output*) con paso de mensaje o de decodificación iterativa [64], [65]. Se debe mencionar que el estándar *New Radio 5G* no especifica el uso de esta clase de decodificadores si bien su literatura hace referencias a que se use el mismo.

Para hacer más fácil la introducción del tema, considérese como ejemplo un mensaje que ha sido codificado con el esquema (3,1), modulado en BPSK y enviado a través del canal AWGN. En este caso, considérese que el decodificador trabajará con teoría de probabilidad. La primera consideración cuando se trabaja con probabilidades consiste en que los valores recibidos del vector r son representados como una “creencia”, esto por la naturaleza de probabilidad que el símbolo sea -1 o +1 [65]. Por ejemplo, considérese que el vector recibido es $r = [0.01 - 2.2 - 0.5]$, por lo tanto, el valor -2.2 y -0.5 representan una “creencia fuerte” de que el símbolo sea $L_2 = L_3 = -1$, en donde L_i es el valor del vector “creencia” L . Esto se da porque existe una gran probabilidad de que ambos valores sean -1. Pero, la pregunta es ¿qué pasa con el valor 0.01? nótese de que en este caso, el valor está muy cercano a la regla de decisión del demodulador BPSK. En consecuencia, si se trabaja con una probabilidad simple, no hay una “creencia” contundente de que el símbolo sea -1 o +1, por lo tanto, se lo denominará como “valor solitario”. Para evitar casos como

el último, la teoría de probabilidad trabaja mediante la suma de valores extrínsecos e intrínsecos [64]. El valor intrínseco es definido como la ganancia que registra un valor respecto a los valores extrínsecos. Como ejemplo, considérese el cálculo de L_1 que se determina usando r_1, r_2 y r_3 . Matemáticamente esto se puede expresar mediante la ecuación 1.82.

$$L_1 = r_1 + (r_2 + r_3) \quad (1.82)$$

Donde r_1 es el valor intrínseco y r_2, r_3 son los valores extrínsecos⁶. De esta manera, L_2 y L_3 serán calculados de la siguiente manera.

$$L_2 = r_2 + (r_1 + r_3) \quad (1.83)$$

$$L_3 = r_3 + (r_1 + r_2) \quad (1.84)$$

Este tipo de decodificación que se basa en la suma de valores intrínsecos y extrínsecos es conocida como SISO (*Soft Input – Soft Output*) [64]. La Figura 1.26 es una representación de este tipo de decodificadores en una forma general.

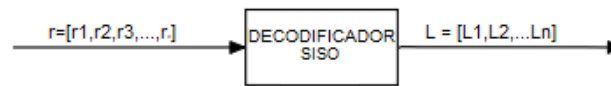


Figura 1.26. Decodificador tipo SISO

De forma general se puede decir que la suma de valores intrínsecos y extrínsecos permite calcular un valor L_i en base al índice de probabilidad logarítmica [66]. El mismo puede ser calculado por medio la relación entre probabilidades de encontrar un valor decodificado 0 y 1 dados los valores del vector r .

$$L_i = \frac{\Pr(c_i=0 | r_1, r_2, r_3)}{\Pr(c_i=1 | r_1, r_2, r_3)} \quad (1.85)$$

Si se desarrollan los cálculos en base a las dos probabilidades dadas que están en función de su densidad espectral de potencia dada por la ecuación 1.1, se tendrá la ecuación 1.86.

$$L_i = e^{\frac{2}{\sigma^2}(r_1+r_2+r_3)} \quad (1.86)$$

Si se obtiene el logaritmo de esta expresión, se tendrá la ecuación 1.87.

$$L_i = \frac{2}{\sigma^2}(r_1 + r_2 + r_3) \quad (1.87)$$

⁶ En probabilidad, se denomina valor intrínseco o principal al valor que tiene en cuenta la suma de todos los componentes que influyen alrededor de él denominados valores intrínsecos. En este caso r_1 es el valor principal de la creencia L_1 y es influenciada por el resto de los componentes del sistema que son r_2 y r_3 , que en este caso son los valores intrínsecos. Recuérdese que el valor extrínseco está dado en función de la creencia L_i que se está analizando.

El término $\frac{2}{\sigma^2}$ es un factor α positivo que puede ser despreciable, tal que puede ser de valor 1. De esta manera se tendrá el índice de probabilidad logarítmica.

$$L_i = (r_1 + r_2 + r_3) \quad (1.88)$$

Observe que esta expresión coincide con las ecuaciones 1.82, 1.83 y 1.84. El entendimiento de la suma de valores intrínsecos y extrínsecos es vital y además este esquema en particular servirá de ayuda para la decodificación LDPC en *New Radio 5G*.

Introducción a decodificadores LDPC para el estándar *New Radio 5G*

En esta parte se revisa el tipo de decodificador SISO para una detección con *single parity check*. Recuérdese que, en secciones anteriores, se mencionó por medio de la Figura 1.26 que parte de la estructura de la matriz o grafo base corresponde a las filas para los códigos de verificación de paridad simple (*Single Parity Check* en inglés) [52], [63]. De hecho, este aspecto del esquema de codificación es muy importante para la decodificación de la palabra código LDPC en el estándar *New Radio 5G*, por lo que, es necesario revisar en conjunto el esquema mencionado, tanto en codificación [67] como en la decodificación [68].

Se define a un esquema de codificación de verificación de paridad como aquel que toma un mensaje de $k = n - 1$ bits y lo convierte en una palabra código de n bits por medio de la añadidura de un bit de paridad p . En la ecuación 1.91 se incluye la suma lógica módulo 2 para determinar el bit de paridad dado el vector de mensaje m de la ecuación 1.89.

$$m = [m_1, m_2, \dots, m_{n-1}] \quad (1.89)$$

$$c = [m_1, m_2, \dots, m_{n-1}, p] \quad (1.90)$$

$$p = m_1 \oplus m_2 \oplus m_3 \oplus \dots \oplus m_{n-1} \quad (1.91)$$

Como ejemplo, considérese que se maneja un esquema de codificación SPC (3,2). En este caso, el diccionario de codificación A viene dado por la Tabla 1.9.

Tabla 1.9. Esquema de Codificación SPC (3,2)

m	c
[0 0]	[0 0 0]
[0 1]	[0 1 1]
[1 0]	[1 0 1]
[1 1]	[1 1 0]

Ahora, recuerde que, por lo visto, los decodificadores tipo SISO son la suma de valores intrínsecos y extrínsecos para determinar las creencias del vector L que en este caso son $[L_1, L_2, L_3]$. Antes considérese que los valores intrínsecos están dados por el vector $[l_{ext,1}, l_{ext,2}, l_{ext,3}]$ y los valores extrínsecos por el vector $[l_1, l_2, l_3]$, tal como puede apreciarse en la Figura 1.27.

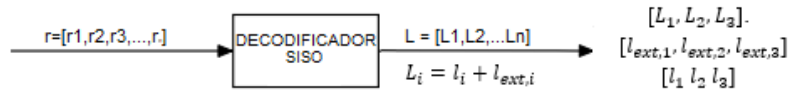


Figura 1.27. Decodificador tipo SISO para SPC

Ahora, en el caso particular de L_1 y como se había explicado en el ejemplo anterior, se conoce que valor intrínseco es proporcional a r_1 y extrínsecos a r_2 y r_3 , pero en este caso la palabra código c se forma a partir de una suma lógica mod 2. Por este motivo, se puede concluir que la búsqueda de uno de los valores extrínsecos de la palabra código estimada \hat{c} , está en función de la suma lógica de los dos valores restantes que son intrínsecos, tal como se indica para el caso de c_1 por medio de la ecuación 1.92.

$$c_1 = c_2 \oplus c_3 \quad (1.92)$$

Así mismo, considérese los índices de probabilidad logarítmica para L_2 y L_3 , tal como se realizó en la ecuación 1.85.

$$l_2 = \frac{\Pr(c_2=0 | r_2)}{\Pr(c_2=1 | r_2)} \quad (1.93)$$

$$l_3 = \frac{\Pr(c_3=0 | r_3)}{\Pr(c_3=1 | r_3)} \quad (1.94)$$

La expresión $\Pr(c_2 = 0 | r_2)$ hace referencia a la probabilidad p_2 de que la palabra código sea 0 dado r_2 , mientras que $\Pr(c_2 = 1 | r_2)$ vendría a ser la probabilidad $(1 - p_2)$. Si se toma en cuenta la tabla lógica presentada en la Tabla 1.9, la probabilidad p_1 de que el bit $c_1 = 0$ viene dada por la ecuación 1.95. Considérese el desarrollo de esta ecuación en la ecuación 1.96.

$$p_1 = p_2 p_3 + (1 - p_2)(1 - p_3) \quad (1.95)$$

$$p_1 = p_2 p_3 + 1 - p_3 - p_2 + p_2 p_3$$

$$(1 - p_1) = p_2(1 - p_3) + (1 - p_2)p_3 \quad (1.96)$$

$$p_1 - (1 - p_1) = p_2(p_3 - (1 - p_3)) + (1 - p_2)((1 - p_3) - p_3)$$

$$p_1 - (1 - p_1) = [p_2 - (1 - p_2)][p_3 - (1 - p_3)]$$

Donde la ecuación 1.96 representa la probabilidad que el bit $c_1 = 1$ que se puede deducir a partir de la ecuación 1.95. Si se realizan ciertas operaciones con el resultado de la última ecuación se tendrá como resultado la ecuación 1.97.

$$\frac{p_1 - (1 - p_1)}{p_1 + (1 - p_1)} = \frac{p_2 - (1 - p_2)}{p_2 + (1 - p_2)} \frac{p_3 - (1 - p_3)}{p_3 + (1 - p_3)} \quad (1.97)$$

La misma ecuación puede ser reescrita, tal como se muestra en la ecuación 1.98.

$$\frac{1 - \frac{1 - p_1}{p_1}}{1 + \frac{1 - p_1}{p_1}} = \frac{1 - \frac{1 - p_2}{p_2}}{1 + \frac{1 - p_2}{p_2}} \frac{1 - \frac{1 - p_3}{p_3}}{1 + \frac{1 - p_3}{p_3}} \quad (1.98)$$

Donde la relación $\frac{1 - p_1}{p_1}$ es el valor extrínseco de L_1 en función de los valores intrínsecos L_2 y L_3 . Ahora considérese que se tiene la función $\tanh(x)$, misma que viene descrita en la ecuación 1.99.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (1.99)$$

Recuerde que la función $\tanh(x)$ es impar, por lo tanto, si $x > 0$ dicha función será positiva. Ahora, así se puede adaptar la Ecuación 1.98 a la Ecuación 1.99. De esta manera, se obtendrá la ecuación 1.100.

$$\tanh\left(\frac{l_{ext,1}}{2}\right) = \tanh\left(\frac{l_2}{2}\right) \tanh\left(\frac{l_3}{2}\right) \quad (1.100)$$

Por lo tanto, si se aplica el principio de paridad de la función para solamente obtener el signo de este, se tendrá como resultado la ecuación 1.101.

$$\text{sgn}(l_{ext,1}) = \text{sgn}(l_2) \text{sgn}(l_3) \quad (1.101)$$

La ecuación 1.101 es importante para la decodificación SISO para SPC, ya que marca el signo del valor extrínseco y es parte de lo que se busca. Ahora, si se determina el valor absoluto de la ecuación 1.100 se presenta el siguiente desarrollo desde la ecuación 1.102.

$$\tanh\left(\frac{|l_{ext,1}|}{2}\right) = \tanh\left(\frac{|l_2|}{2}\right) \tanh\left(\frac{|l_3|}{2}\right) \quad (1.102)$$

$$\log\left(\tanh\left(\frac{|l_{ext,1}|}{2}\right)\right) = \log\left(\tanh\left(\frac{|l_2|}{2}\right) \tanh\left(\frac{|l_3|}{2}\right)\right) \quad (1.103)$$

$$\log\left(\tanh\left(\frac{|l_{ext,1}|}{2}\right)\right) = \log\left(\tanh\left(\frac{|l_2|}{2}\right)\right) + \log\left(\tanh\left(\frac{|l_3|}{2}\right)\right) \quad (1.104)$$

Ahora, considere una función $f(x) = \log\left(\tanh\left(\frac{|x|}{2}\right)\right)$ misma que cumple la propiedad $f(x) = f^{-1}(x)$, donde $f^{-1}(x)$ es la función inversa de $f(x)$. Bajo esta consideración se obtiene la ecuación 1.105.

$$f(|l_{ext,1}|) = f(|l_2|) + f(|l_3|) \quad (1.105)$$

De esta manera, se puede definir al valor extrínseco de L_1 por medio de la ecuación 1.106.

$$l_{ext,1} = (S - f(|l_1|) \cdot P) \operatorname{sgn}(l_1) \quad (1.106)$$

Donde las variables S y P vienen dadas, ya para una forma general cuando se tiene un vector r de n bits, por las ecuaciones 1.107 y 1.108, respectivamente [69].

$$S = f(|l_1|) + f(|l_2|) + \dots + f(|l_n|) \quad (1.107)$$

$$P = \operatorname{sgn}(l_1) \operatorname{sgn}(l_2) \dots \operatorname{sgn}(l_n) \quad (1.108)$$

Se debe mencionar que esto se puede realizar de la misma forma, tanto para $l_{ext,2}$ y $l_{ext,3}$. Ahora, dos consideraciones finales: la primera, el valor del término intrínseco l_i en parte ya se lo revisó en el capítulo anterior y corresponde al valor de la entrada de r_i multiplicado por el factor α , tal como puede apreciarse en la ecuación 1.109.

$$l_i = \frac{2}{\sigma^2} r_i \quad (1.106)$$

Y la segunda consideración tiene que ver con la ecuación 1.105, en donde se tiene que la expresión $f(|l_2|) + f(|l_3|)$ es una aproximación de la función de valor mínimo entre dos variables, conocida en inglés como *MINSUM APPROXIMATION*. De tal manera, matemáticamente se tendrá el siguiente desarrollo.

$$f(|l_2|) + f(|l_3|) \approx f(\min(|l_2|, |l_3|)) \quad (1.107)$$

$$f'(f(|l_{ext,1}|)) \approx f'(f(\min(|l_2|, |l_3|))) \quad (1.108)$$

$$|l_{ext,1}| = \min(|l_2|, |l_3|) \quad (1.109)$$

El motivo de realizar esta aproximación es porque el cálculo de la función $f(x) = \log\left(\tanh\left(\frac{|x|}{2}\right)\right)$ es computacionalmente muy pesada y consume muchos recursos de memoria.

Decodificación LDPC SISO de paso de mensaje iterativo para el estándar *New Radio 5G*.

Considérese que se tiene un vector de la palabra código que obedece a la expresión $c = [c_1 \ c_2 \ \dots \ c_n]$ que a su vez ha sido transmitido usando modulación BPSK y ha recorrido por un canal tipo AWGN, convirtiéndose en un vector $r = [r_1 \ r_2 \ \dots \ r_n]$. En este caso, para una entrada r_i , el canal AWGN puede ser tomado en cuenta tal como se indica en la ecuación 1.110.

$$r_i = S_i + n \quad (1.110)$$

Donde S_i es el símbolo modulado y transmitido en BPSK. Ahora, basado en el índice de probabilidad logarítmica, la decodificación basada en la creencia l_i para una de las entradas de r_i viene dada por la ecuación 1.111.

$$l_i = \log \left(\frac{\text{Pr}(r_i | c_i = 0)}{\text{Pr}(r_i | c_i = 1)} \right) \quad (1.111)$$

Por lo tanto, el índice de probabilidad logarítmica para todo el vector de creencias L viene dado por la ecuación 1.112.

$$L_i = \log \left(\frac{\text{Pr}(c_i = 0 | r)}{\text{Pr}(c_i = 1 | r)} \right) \quad (1.112)$$

Se debe tener presente que para conocer el valor intrínseco del vector de creencia L se debe analizar todo el vector r . Además, el uso de BPSK, se vuelve un proceso complejo debido a la gran cantidad de bits que maneja LDPC, lo cual consume recursos de memoria y tiempo en el procesamiento y hace el proceso ineficaz. Por este motivo, de acuerdo a [70] y [71] la decodificación tipo SISO maneja iteraciones con paso de mensaje para obtener un resultado aproximado a través de la repetición de un proceso en una o varias veces hasta tener una respuesta lo más aproximada posible a la respuesta dada por la ecuación 1.112, disminuyendo así la carga computacional y la complejidad de implementación del algoritmo. Para esto, la decodificación NR-LDPC se hace en base a un concepto denominado “estructura local” que se encuentra dentro de la matriz de verificación de paridad H tipo *sparse*. Parte de dicha estructura se encuentra en la Figura 1.28 [72].



Figura 1.28. Parte de la estructura local en una matriz de verificación de paridad H

Ahora observe el ejemplo en el que se desea conocer el valor de L_1 . Para este caso, recuérdese que un decodificador tipo SISO realiza la suma de valores intrínsecos y extrínsecos. En este caso, al usar iteraciones, la decodificación trabaja con estimaciones, donde la primera estimación es el valor intrínseco de la creencia denominada l_{1-0} , mismo

que viene dado por la expresión de la ecuación 1.106. Desde la segunda estimación se trabajará con los valores intrínsecos y con las filas de la matriz de verificación de paridad H , esto con el objetivo de usar verificación de paridad para involucrar al bit que se desea con respecto a otros valores y conocer el valor de la creencia, en este caso, L_1 . Aplicando conceptos previos sobre la matriz de paridad, se tienen los siguientes resultados en las filas 1, 5 y 9.

$$F_1: c_1 + c_4 + c_8 + c_{12} = 0 \rightarrow l_{1-1} = LLR | r_4, r_8, r_{12} \quad (1.113)$$

$$F_2: c_1 + c_2 + c_{10} + c_{20} = 0 \rightarrow l_{1-5} = LLR | r_2, r_{10}, r_{20} \quad (1.114)$$

$$F_3: c_1 + c_6 + c_{16} + c_{18} = 0 \rightarrow l_{1-9} = LLR | r_6, r_{16}, r_{18} \quad (1.115)$$

Para este caso, cada fila representa una nueva estimación del valor de r_1 , este valor se determina por el índice de probabilidad logarítmico de los otros valores que son diferentes de cero en la fila (LLR por sus siglas en inglés) que a su vez se puede calcular mediante el uso de un decodificador SISO SPC. Así, en este caso, se tendrá una estimación inicial dado por el valor intrínseco del bit r_1 y tres más dados por los valores extrínsecos. Nótese que la suma de valores fila por fila es un proceso muy similar al esquema de decodificación SISO Repetitiva que fue presentado en la ecuación 1.88. Es por este motivo que la decodificación LDPC para el estándar *New Radio* 5G puede realizarse mediante una combinación entre decodificación SISO Repetitiva y SPC donde la aproximación *MINSUM* definido en la ecuación 1.109 jugará un papel fundamental en la estimación de la palabra código [35].

Decodificación tipo MINSUM para NR-LDPC

El esquema de decodificación tipo *MINSUM* es aquel que se utiliza para la estimación de la palabra código \hat{c} cuando se utiliza el esquema de codificación LDPC [70], [71], [35]. Tal como puede apreciarse en la Figura 1.31, este tipo de decodificación maneja iteración con paso de mensaje por medio de múltiples operaciones repetitivas entre filas y columnas que se realizan en un matriz L tipo *sparse* y de iguales dimensiones que la matriz de verificación H para estimar la palabra código. De hecho, esta matriz es la denominada “estructura local” antes mencionada.

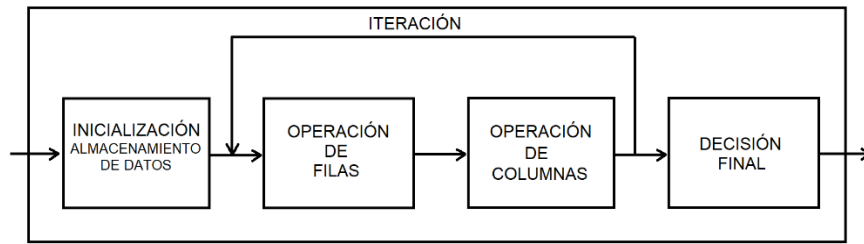


Figura 1.29. Decodificador NR-LDPC tipo MINSUM

La inicialización de datos consiste en la creación de la ya mencionada matriz de almacenamiento L , que tal como se mencionó, tiene la misma dimensión de la matriz de verificación H . Por ejemplo, si la tasa de codificación R usando un esquema de codificación NR-LDPC es $\frac{1}{2}$ y se utiliza un grafo base BG1 con tamaño de expansión $Z_c = 16$, tanto H como L corresponderán a una matriz tipo *sparse* de dimensión $(22 \times 46) * 16$. Ahora bien, la matriz de almacenamiento L se crea en función de dos parámetros: la matriz de verificación de paridad H y el vector recibido r del canal AWGN. En primera instancia, el valor de las entradas de la matriz de verificación de paridad H determina las entradas de la matriz de almacenamiento L en función de la condición dada en la ecuación 1.116.

$$\begin{cases} L(i, j) = 0 \text{ si } H(i, j) = 0 \\ L(i, j) \neq 0 \text{ si } H(i, j) = 1 \end{cases} \quad (1.116)$$

Los valores que se definen como $L(i, j) \neq 0$ son reemplazados por el valor de r_j . Esto quiere decir que el valor de r_j se reemplazará en toda la columna. Como ejemplo, considérese la matriz de verificación de paridad H de dimensiones 4×7 dada por la ecuación 1.117 y su matriz de almacenamiento L expresada en la ecuación 1.119 cuando se parte de un vector recibido del canal AWGN $r = [r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6 \ r_7]$.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (1.117)$$

$$L = \begin{bmatrix} r_1 & r_2 & r_3 & 0 & r_5 & 0 & 0 \\ 0 & r_2 & r_3 & r_4 & 0 & r_6 & 0 \\ r_1 & r_2 & 0 & r_4 & 0 & 0 & r_7 \\ r_1 & 0 & r_3 & 0 & r_5 & r_6 & r_7 \end{bmatrix} \quad (1.118)$$

Una vez creada la matriz de almacenamiento, se procede a realizar la operación con las filas. Para esto, observe que se debe aplicar el principio de decodificación SISO para SPC utilizando la aproximación MINSUM. Basado en la ecuación 1.106 y 1.109 y, al ser la fila todo un vector, el decodificador primero escoge los dos valores absolutos más pequeños y calcula el signo de la fila. Posterior a eso se reemplazan todos los valores de la fila por el

primer mínimo absoluto, a excepción de los ceros y de la posición que contiene a dicho valor, que es reemplazado por el segundo mínimo absoluto. Como ejemplo, considérese que el vector recibido luego de su transmisión por el canal AWGN es $r = [0.2, -0.3, 1.2, -0.5, 0.8, 0.6, -1.1]$. En este caso, si se maneja la matriz de almacenamiento L de la Ecuación 1.117, la primera fila de esta matriz viene dado por el vector $L = [0.2, -0.3, 1.2, 0, 0.8, 0, 0]$. Realizando las operaciones en la primera fila se tendrá una expresión como la de la ecuación 1.119.

$$L_{1,j} = [-0.3, 0.2, -0.2, 0, -0.2, 0, 0] \quad (1.119)$$

Una vez finalizada la operación en todas las filas de la matriz L , se procede a realizar la operación con cada una de las columnas. Para esto, se usa un decodificador SISO para códigos de repetición. Recuérdese que este tipo de decodificadores, tal como se indicó en la ecuación 1.88, la creencia \tilde{L} (se denota así para evitar confusiones con la notación que se está usando en este momento) es la suma del valor intrínseco con la de los extrínsecos. En este caso, el valor intrínseco viene dado por el valor de r_j del vector r y los valores extrínsecos que corresponden a la sumatoria de todos los elementos de la columna de la matriz. Matemáticamente esto puede ser descrito por medio de la ecuación 1.120.

$$\widetilde{L}_{1-1} = r_j + \sum L_{i,1-1} \quad (1.120)$$

Observe que el valor \widetilde{L}_1 no es más que la palabra código estimada para la primera entrada luego de la primera iteración. Para esto, si se considera la primera columna del ejemplo propuesto se tendrá la ecuación 1.121.

$$L_{i,1-2} = \begin{bmatrix} -0.3 \\ 0 \\ -0.3 \\ -0.6 \end{bmatrix} \quad (1.121)$$

En el caso del ejemplo propuesto, se tiene que si se realiza el procedimiento que se expone, la suma será -1.0 y la palabra decodificada y demodulada en dicha entrada cambia de 0 a 1 con respecto a su original.

Para realizar nuevas iteraciones, el nuevo vector r se forma de las sumas de cada columna, que son las creencias \tilde{L}_j , menos el valor de cada entrada por columna. Matemáticamente, para la primera columna de la matriz L del ejemplo, la segunda iteración se describe por medio de la ecuación 1.122.

$$L_{i,1-2} = \tilde{L}_1 - L_{i,1-1} \quad (1.122)$$

Que para el ejemplo tendrá el siguiente resultado:

$$L_{i,1-2} = \begin{bmatrix} -1 - (-0.3) \\ -1 - (0) \\ -1 - (-0.3) \\ -1 - (-0.6) \end{bmatrix} = \begin{bmatrix} -0.7 \\ -1 \\ -0.7 \\ -0.4 \end{bmatrix} \quad (1.123)$$

Una vez realizado el número de iteraciones requerido para estimar el mensaje se puede decir que el decodificador ha convergido.

Decodificador NR-LDPC tipo MINSUM a través de corrección de variables mediante el uso de *OFFSET* y *LAYERING*

Pese a que la decodificación tipo MINSUM tiene un gran rendimiento en términos de tiempo de procesamiento por ser un método paralelo de decodificación y ya que se usa previamente la cuantificación arreglada de punto fijo, unos de los grandes inconvenientes que presentan este tipo de decodificación vienen dados por el apareamiento de valores que en algunos casos son demasiado pequeños, además de un alto número de iteraciones para alcanzar un valor que se asemeje al del índice de probabilidad logarítmico y la existencia de grandes dimensiones de las matrices que se trabajan. Es por esto, que el decodificador, pese a ser robusto, puede mejorar y por esto, el estándar NR-LDPC ofrece dos técnicas/alternativas que se las denominada corrección de variables [73].

La primera mejora es conocida como *OFFSET* y es una técnica muy usada en decodificadores tipo SISO MINSUM. Esta técnica consiste en multiplicar los valores de la función MINSUM realizada sobre el vector r por una constante a y luego se suma con otra constante b , tal como puede verse en la ecuación 1.124.

$$x = a(\text{MINSUM}(r)) + b \quad (1.124)$$

Esto ayuda bastante a que las operaciones entre filas y columnas no sea con valores que posiblemente puedan estar cerca de la regla de decisión, algo que ya se mencionó en la introducción a decodificadores SISO bajo el término de valores “solitarios”, valores que por probabilidad tienen la más alta incertidumbre de no saber si al estimar sean correctos o no y pueden empeorar el rendimiento del sistema en términos de BER.

La segunda mejora es conocida como decodificación por capas o *Layered* [72]. El método consiste en que la matriz de verificación de paridad H se divida en grupos de filas consecutivas para desarrollar el proceso de decodificación, en el resultado de la decodificación de cada capa es usada para el cálculo de la siguiente. Para una mejor explicación, considérese la matriz de verificación de paridad H dada por la ecuación 1.123.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (1.125)$$

La división por capas es homogénea y está en función del peso de las columnas. El peso es determinado por la cantidad de entradas de valor 1 por columna. En este caso, el estándar *New Radio 5G* dicta que se debe tener un peso de valor 1 por capa. Dicho esto, la matriz H de la ecuación 1.125 se divide en dos partes (en la ecuación 1.126 esto puede notarse por los colores azules y rojo). Note que cada capa tiene un peso de valor 1 y que cada iteración será de dos partes.

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (1.126)$$

Considérese ahora que se recibe un vector que viene dado por $r = [0.2, -0.3, 1.2, -0.5, 0.8, 0.6, -1, 1]$. Para este caso, a diferencia del primer caso presentado en el anterior literal, la matriz de almacenamiento L solo se realiza para la primera capa, de tal manera que dicha matriz se presenta en la ecuación 1.127.

$$L_{1-1} = \begin{bmatrix} 0.2 & -0.3 & 1.2 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & -0.5 & 0 & 0.6 & -1.1 \end{bmatrix} \quad (1.127)$$

Si se realizan las operaciones respectivas de decodificación tanto en filas como en columnas se tendrá la creencia dada por la ecuación 1.128.

$$L_{1-1} = [-0.1, -0.1, 1, -1.1, 0.6, 1.1, -1.6] \quad (1.128)$$

Ahora, dentro de la primera iteración es el turno de la segunda capa. Tal como se describe en la ecuación 1.129, la matriz de almacenamiento en este caso se forma en función del vector creencia L_{1-2} y de la segunda parte de la matriz H de la ecuación 1.126.

$$L_{1-2} = \begin{bmatrix} -0.1 & -0.1 & 0 & 1 & 0 & 0 & -1.6 \\ 0 & 0 & 1 & 0 & 0.6 & 1.1 & 0 \end{bmatrix} \quad (1.129)$$

Realizando la decodificación, la creencia obtenida viene dada por la ecuación 1.130.

$$L_{1-2} = [-0.2, -0.2, 1.6, -1.2, 1.6, 1.7, -1.7] \quad (1.130)$$

Ahora para la segunda iteración, recuerde que se había mencionado que debe restarse la creencia menos la antigua entrada, que, en este caso, es la capa 1 de la última iteración, pero realizado el proceso de decodificación SISO SPC. De esta manera, al ejecutar la sustracción se obtiene la ecuación 1.131.

$$L_{2-1} = \begin{bmatrix} 0.1 & -0.4 & 1.8 & 0 & -0.2 & 0 & 0 \\ 0 & 0 & 0 & -0.6 & 0 & 0.5 & -0.5 \end{bmatrix} \quad (1.131)$$

Caiga en cuenta que el proceso de decodificación por capas es muy útil porque disminuye las iteraciones y disminuye el tiempo de convergencia. Normalmente, basado en estudios técnicos y prácticos con este método, se ha determinado que el número de iteraciones necesario para que el decodificador haga convergencia con buenos resultados en términos de BER es de entre 6 y 8 [7]. Tal como puede apreciarse, el esquema de codificación LDPC para el estándar *New Radio 5G* es robusto e intenta reducir tiempos de procesamiento por medio de métodos como la decodificación SISO MINSUM por capas; sin embargo, se debe acotar que esto aumenta la complejidad de implementación. Finalmente, la Figura 1.30 representa el sistema de comunicación con decodificación SISO MINSUM con capas y con OFFSET.

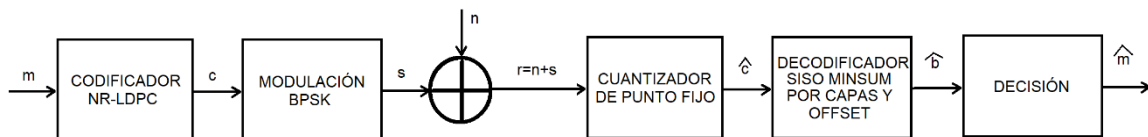


Figura 1.30. Sistema de Comunicación con codificación NR-LDPC, modulación BPSK, decodificación SISO MINSUM sobre un canal AWGN

1.3.9 CODIFICACIÓN NR-POLAR

Los códigos polares son un concepto relativamente nuevo en el campo de la codificación. Fueron inventados e introducidos por Erdal Arikan en el año 2008 y basados en la idea de polarización del canal por medio de una matriz de transformación o *Kernel* [74], mismos que parten de la idea del parámetro de Bhattacharya [75]. Esto hace que este esquema de codificación se clasifique como un código secuencial, algo que trae algunos inconvenientes y por lo cual se dudó en un inicio que este tipo de codificación vaya a ser práctico [76]. El primer inconveniente de esta codificación es que a diferencia de LDPC, la decodificación en códigos polares debe ser basada en cancelación sucesiva o bit a bit, algo que disminuye el rendimiento del sistema de comunicación por el aumento del retardo. El segundo inconveniente se da por la misma polarización de los bits⁷ que afecta el rendimiento en términos del BER, característica que se considera como “mediocre” para palabras código de tamaño n pequeñas o medianas y solo presenta un gran rendimiento en palabras código que tienen más de un millón de bits ($\#bits > 2^{20}$) [76]. Por este motivo se introdujeron algunas mejoras para hacer una decodificación más rápida; entre ellas se encuentra la introducción de listas por el uso de códigos de redundancia cíclica (CRC en sus siglas en

⁷ Un bit polarizado es aquel que ha sido transmitido con la ayuda del concepto de transformación polar. De esta manera, el bit que se transmite contiene parte de la información de todo el vector transmitido.

inglés), este último cambio hizo que el esquema de codificación sea elegido por parte de la 3GPP como parte de los esquemas de codificación para la serie de especificaciones *New Radio 5G* y viene definido por medio del Reporte Técnico TS 38.212 [1]. Si bien, el esquema de codificación no está descrito a detalle dentro del estándar, en este trabajo de titulación se dará una descripción básica de cómo se realiza la codificación y decodificación Polar.

TRANSFORMACIÓN POLAR

Considere ahora que se envían dos bits de forma paralela a través de un canal AWGN, tal como se indica en la Figura 1.31.

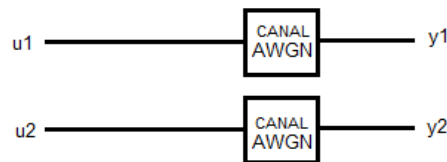


Figura 1.31. Esquema de transmisión donde se envían dos bits sobre un canal AWGN

Caiga en cuenta que no existen motivos por los cuales creer que haya información confiable sobre el valor de u_1 y u_2 . Ahora, considérese que se transmiten los dos mencionados bits utilizando el esquema que se presenta en la Figura 1.32, donde x_1 denota una suma lógica entre u_1 y u_2 considerado como el principio de codificación polar [76]. A su vez, si se observa la misma figura, x_2 solamente será el bit u_2 .

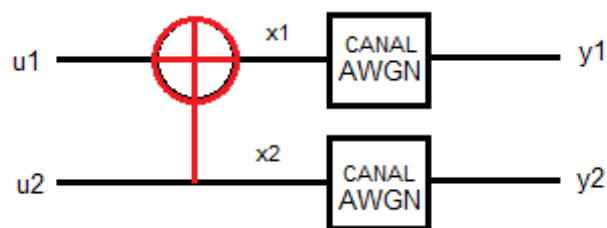


Figura 1.32. Esquema básico para transformación polar

De donde, la siguiente Tabla indica las salidas correspondientes del sistema.

Tabla 1.10. Salidas del esquema básico de transformación polar

Entrada $[u_1 \ u_2]$	$y_1 [u_1 + u_2]$	$y_2 [u_2]$
$[0 \ 0]$	0	0

[0 1]	1	0
[1 0]	1	0
[1 1]	0	1

En este caso, existe alguna forma de tener cierta confiabilidad de que el bit u_1 recuperado en la recepción sea correcto debido al uso de u_2 en y_1 , pero, aun así, existe la posibilidad de que el bit sea erróneo debido a las alteraciones del canal AWGN sobre los bits u_1 y u_2 . De hecho, este es el motivo por el cual el BER suele ser elevado en sistemas que manejan codificación polar, algo que se mencionó se reduce de manera proporcional a la longitud del mensaje. Ahora, si se expresa tanto x_1 y x_2 como un vector de longitud 2 en términos de u_1 y u_2 , se tendrá la ecuación 1.132 que se describe a continuación.

$$u^{(2)} = [u_1 + u_2, u_2] \quad (1.132)$$

Ahora, se va a reescribir el vector en función de una multiplicación matricial; la ecuación 1.133 describe dicha multiplicación [52].

$$u^{(2)} = [u_1 \ u_2] \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (1.133)$$

La matriz cuadrada de dimensión 2 es conocida como matriz de transformación polar o *Kernel* y está denotada como G_2 ; se debe tomar en cuenta que toda transformación polar parte de esta matriz. En forma general para un mensaje de longitud k , tal que su vector mensaje viene dado por $m = [u_1, u_2, u_3, \dots, u_k]$, la matriz de transformación polar de orden $N = k$ viene dada por la n -ésima potencia de Kronecker [52], tal como se describe en la ecuación 1.134.

$$G_N = G_2^{\oplus n} \quad (1.134)$$

Donde n se calcula por medio de la ecuación $n = \log_2(k)$, donde k debe ser una potencia de 2. Al partir de la ecuación 1.134 que se encuentra en función de la potencia de Kronecker, la expresión puede ser derivada a una forma más comprensible por medio de la ecuación 1.135.

$$G_N = \begin{bmatrix} G_2^{\oplus n-1} & 0 \\ G_2^{\oplus n-1} & G_2^{\oplus n-1} \end{bmatrix} \quad (1.135)$$

Como ejemplo, considérese que se desea transmitir un mensaje de longitud $k = 4$. En este caso se tendrá una matriz de transformación cuadrada de dimensión 4 dada por la ecuación 1.136.

$$G_4 = \begin{bmatrix} G_2^{\oplus 1} & 0 \\ G_2^{\oplus 1} & G_2^{\oplus 1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (1.136)$$

Al continuar con el ejemplo, se tiene que la transformación polar $u^{(k=N)}$ será dada por la ecuación 1.137.

$$u^{(4)} = [u_1, u_2, u_3, u_4] \cdot G_4 = [u_1 + u_2 + u_3 + u_4, u_2 + u_4, u_3 + u_4, u_4] \quad (1.137)$$

Tenga en cuenta que la transformación polar no es lo mismo que la codificación polar. La transformación polar es la operación que permite obtener la codificación polar (n, k) , pero esta última contiene algunos detalles extras. Note además que la transformación polar, tal como puede verse en la Figura 1.32, es un proceso paralelo porque todos los bits involucrados en la transformación afectan en alguna medida al vector, algo que puede evidenciarse de mejor forma en la Figura 1.33 para una transformación polar de $N = 8$ [76].

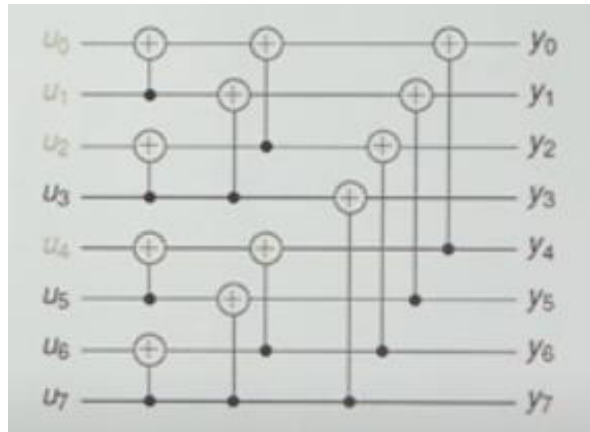


Figura 1.33. Proceso paralelo de transformación polar con $N=8$ bits

Diagrama de árbol binario

Este tipo de diagrama es una forma de representar gráficamente la transformación polar de orden N por medio del esquematizado de nodos [77]. Para esto, se representa cada transformación polar de orden 2, G_2 , como la suma de dos nodos de orden inferior y que se combinan en un solo nodo de orden superior [52]. En la Figura 1.34 se presenta la representación de una transformación polar de orden 2 de acuerdo a la ecuación 1.130.

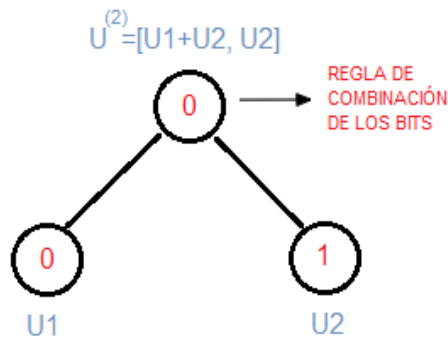


Figura 1.34. Diagrama de árbol binario para una transformación polar de orden 2

De esta forma, la representación puede ser notada para una mayor rapidez en función del nodo y la profundidad del nodo que se desea evaluar dentro del árbol tal que se forma el par $(nodo, profundidad)$. Esto es útil cuando se manejan transformaciones polares de orden superior a 2. En este caso, la Figura 1.35 es la representación gráfica a través del diagrama de árbol binario de una transformación polar de orden 4, tal como se presentó en la ecuación 1.137.

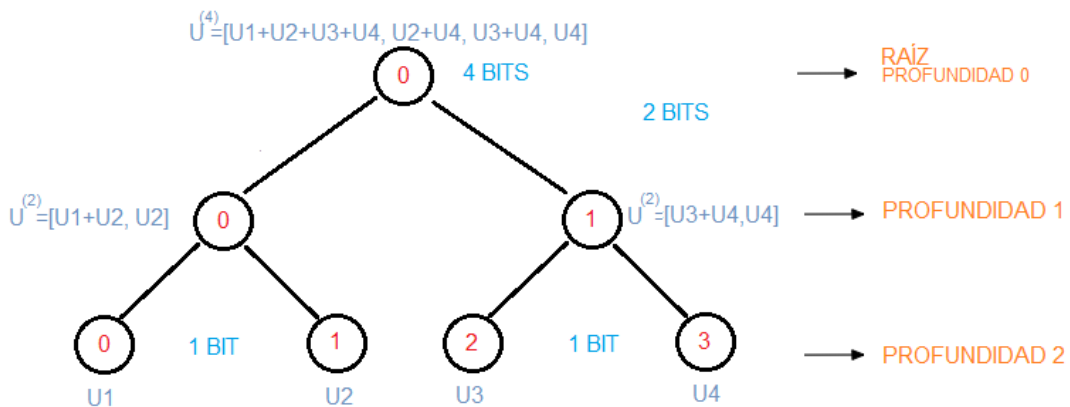


Figura 1.35. Diagrama de árbol binario para una matriz de transformación polar G_4

De los ejemplos propuestos se puede concluir que los árboles binarios tienen una profundidad igual al valor de $n = \log_2(k)$. Así mismo, se puede observar que la profundidad máxima del diagrama es aquella que contiene el mensaje que se desea transformar.

Polarización del canal

Considérese un caso propuesto similar al de la Figura 1.32, pero donde ahora se realiza la transformación polar de un mensaje para $N = k$ bits y se los transmite por el canal AWGN por medio del uso de BPSK. La Figura 1.36 es una representación esquemática inicial del sistema.

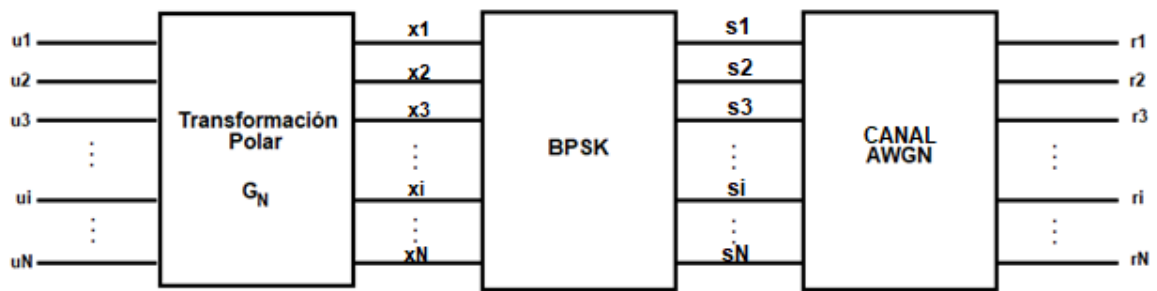


Figura 1.36. Transmisión de un mensaje con transformación polar G_N con modulación BPSK y sobre un canal AWGN

En este caso el vector en la recepción corresponde a $r^{(N)} = [r_1, r_2, r_3, \dots, r_N]$; adicionalmente, recuérdese que de manera previa se mencionó que la transformación polar es un proceso paralelo, algo también visto en la Figura 1.32. Esto quiere decir que, se considerará que en vez de tener una sola secuencia de transmisión en el canal AWGN, se supondrá que cada uno de los $N = k$ bits se transmite en uno de los $N = k$ canales, tal como puede verse en la Figura 1.37 [74].

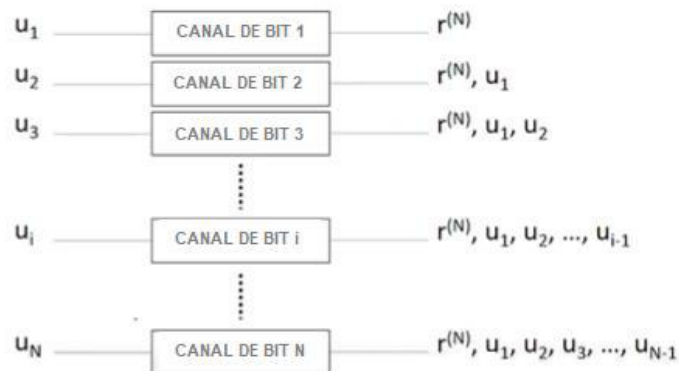


Figura 1.37. Representación de la polarización de canal

Observe que en este caso cada canal recibe el vector $r^{(N)}$ y tal como se revisó anteriormente, esto se debe a que en transformación polar cada bit afecta al otro de alguna manera y eso ocasiona que, por ejemplo, el primer bit sea afectado por todo el vector $r^{(N)}$.

Ahora, considere de manera teórica que el canal de alguna manera genera, a más del vector $r^{(N)}$, los $N - 1$ bits previos al canal en sus salidas. Se debe mencionar que a diferencia de la Figura 1.36, la Figura 1.37 describe el comportamiento real de cómo actúa cada canal con polarización, ya que si el valor de $N = k$ aumenta, los canales de bit de orden superior tienen más valores involucrados en su salida y experimentarán una estimación del bit que será cada vez mejor o peor, pero sin un punto intermedio debido a la propagación del error [52], [75], [78]. Este concepto es lo que se conoce como

polarización de canal y es un concepto útil para entender y realizar la codificación y decodificación.

Secuencia de confiabilidad

Ahora, se debe recordar que, cuando se utiliza una transformación de orden N muy grande, la polarización del canal tenderá a ser mayor y las estimaciones tenderán cada vez más a ser mucho mejores o peores, sin un punto intermedio. Pero ahora la pregunta que surge es: ¿qué canales son los mejores para transmitir la información debido a la polarización del canal?

Pues para esto, se ideó un concepto denominado secuencia de confiabilidad, que se basa en un índice notado por la letra Q que ordena del peor al mejor canal de bit para la transmisión de la información [78]. Por ejemplo, si el mensaje transmitido es de $N = 8$ bits, la secuencia de confiabilidad viene dada por $Q = [1\ 2\ 3\ 5\ 4\ 6\ 7\ 8]$, donde el canal de bit 1 es el que menos se recomienda para transmitir un bit y el canal 8 es el más confiable. Existen métodos para crear este tipo de secuencias, pero el presente documento no se extenderá en el mismo porque el estándar *New Radio* 5G, ya establece su propia secuencia definida en los estándares que se define para un $N = 1024$ según el Reporte Técnico TS 38.212 [79].

Codificación polar (N, k)

Matemáticamente la codificación polar viene dada por la multiplicación entre el vector \bar{u} y la matriz de transformación polar de grado N , de acuerdo con la ecuación 1.138.

$$c = \bar{u} \cdot G_N \quad (1.138)$$

La formación del vector \bar{u} se basa en la asignación del denominado *Frozen Bit* o bit de valor 0 a las primeras $N - k$ posiciones menos confiables dadas por la secuencia de confiabilidad Q , mientras que al resto de las posiciones se les asigna los bits del mensaje. En este caso, el orden de la transformación polar cumplirá la condición $N \leq k$, he ahí que como se mencionó en un inicio, la transformación polar no es la única condición para lograr la codificación polar [3]. Como ejemplo, considérese el esquema de codificación (8,4). En este caso, las primeras $N - k = 4$ posiciones de la secuencia de confiabilidad son consideradas frías o de valor 0, mientras que al resto se le asigna el mensaje. Esto quiere decir que el vector \bar{u} es igual a $\bar{u} = [0\ 0\ 0\ m_1\ 0\ m_2\ m_3\ m_4]$. Si se aplica la transformación polar de u dada por la ecuación 1.136 se tendrá la palabra código c según la ecuación 1.139.

$$c = [m_1 + m_2 + m_3 + m_4, m_1 + m_2 + m_4, m_1 + m_3 + m_4, m_1 + m_4, m_2 + m_3 + m_4, m_2 + m_4, m_3 + m_4, m_4] \quad (1.139)$$

Gráficamente, esta situación puede ser representada por medio del diagrama del árbol binario dado por la Figura 1.38.

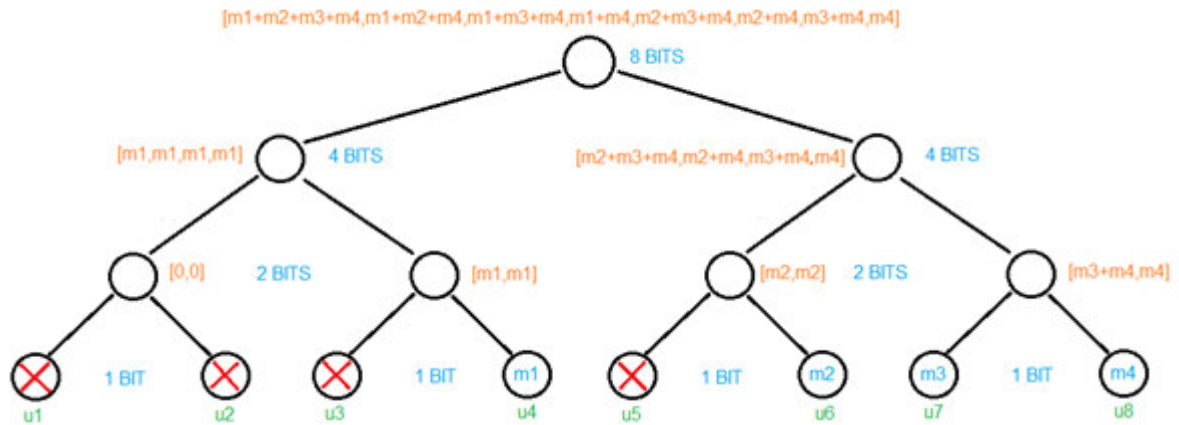


Figura 1.38. Diagrama de árbol binario para un esquema de codificación polar (8,4)

En el caso de *New Radio* 5G, el Reporte Técnico TS 38.212 indica que la longitud de la palabra código c es de longitud de 1024, tal que el mensaje es de longitud 512 bits y se ocupan 512 posiciones frías para un rendimiento óptimo [3]. Tenga en cuenta que, en este caso, la palabra código no es variable, pero pese a eso es usada, algo que es explicado por el uso del esquema de decodificación usado, que introduce ciertas variaciones en el planteamiento del campo de k bits.

Introducción a la decodificación polar (n, k) con cancelación sucesiva (SC)

Se mencionó en capítulos previos que el primer canal de bit contiene la información del vector $r^{(N)}$, mientras que el segundo contiene en su salida la información de $r^{(N)}$ y además u_1 , y así sucesivamente se tiene el contenido del resto de los canales con los $N - 1$ bits previos. Ahora, tal como se puede observar en la Figura 1.39, note que si se cancela la información entre el primer canal de bit y el segundo se obtiene la entrada estimada u_1 , mientras que si se cancela la información entre el segundo canal de bit y el tercero se obtiene la entrada estimada u_2 . Este es el concepto que se conoce como decodificación por cancelación sucesiva [80].

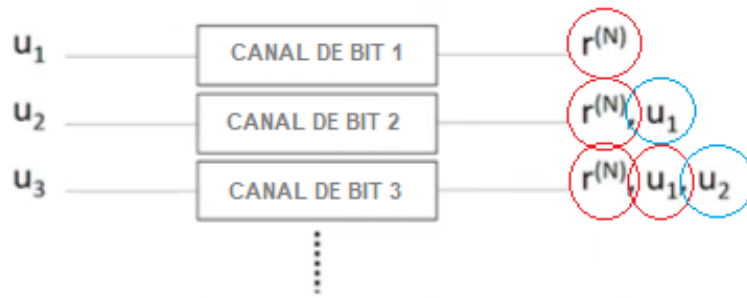


Figura 1.39. Decodificación polar basada en cancelación sucesiva

Lastimosamente en la práctica, los $N - 1$ bits que se mencionan en las Figuras 1.37 y 1.39 no existen, pero, para ventaja, existe un método por el cual se puede recrear el sistema propuesto que se observa en las mencionadas figuras. De tal manera que si se conoce el valor estimado de \hat{u}_1 del vector $r^{(N)}$ del primer canal de bit, se puede realizar la decodificación por cancelación sucesiva, tal como puede verse en la Figura 1.40.



Figura 1.40. Decodificación basada en cancelación sucesiva por medio de la estimación de los bits u_i

Para el caso propuesto, recuerde que en el esquema de codificación polar (N, k) , las primeras $N - k$ entradas de la secuencia de confiabilidad eran conocidas como *Frozen Bits* o bits de valor cero y el resto de las entradas eran asignadas al mensaje [81]. Por lo tanto, si el bit en la posición i dado por la secuencia de confiabilidad se lo considera “congelado”, el valor de u_i tendrá valor cero tanto en transmisión como en recepción. En cambio, en el caso contrario, donde de acuerdo con la secuencia de confiabilidad se tiene que en la posición i se asigna un bit de mensaje m , se deberá estimar el bit aplicando la decodificación del tipo SISO. Esto puede ser descrito de mejor manera por medio de la ecuación 1.140 considerando sus posibles valores.

$$\hat{u}_i = \begin{cases} 0 & \text{si el bit es congelado} \\ \text{otro valor} & \text{si el bit no es congelado} \end{cases} \quad (1.140)$$

Partiendo de lo anterior, debe tomarse en cuenta que, por el tipo de decodificación de cancelación sucesiva, si se estima mal cualquier valor u_i el error se propaga al resto de bits.

Decodificación polar (2,2) por cancelación sucesiva simplificada (SSC)

Ahora, considérese que se envía un mensaje bajo un esquema de codificación polar (2,2) con modulación BPSK y que se transmite por un canal AWGN. Para este caso el sistema viene dado por la Figura 1.43 [81].

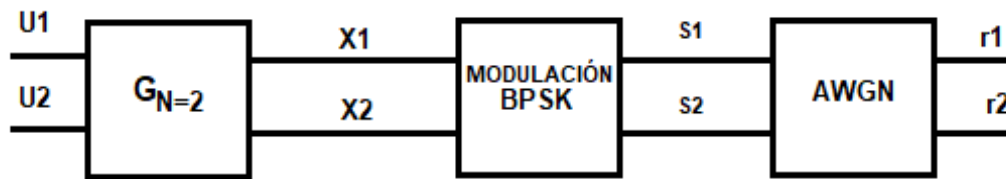


Figura 1.41. Proceso de codificación polar (2,2) con modulación BPSK sobre un canal AWGN

Al aplicar la transformación polar se obtienen los vectores $x_1 = [u_1 + u_2]$ y $x_2 = [u_2]$ que juntos forman el vector x . A partir de la figura se debe mencionar que el vector r_1 es el punto de inicio de la decodificación. En este punto, recuérdese que se había mencionado que el esquema de decodificación SISO con verificación de paridad simple (SPC por sus siglas en inglés), el cual debe ser revisado a fondo en LDPC, es el candidato más adecuado para este caso y, de hecho, es usado para estimar el bit \hat{u}_1 . Por lo tanto, la ecuación 1.141 describe el proceso matemático para la estimación de u_1 en la decodificación SISO SPC [82].

$$\begin{aligned}
 \text{sea } L(u_1) &= f(r_1, r_2) = \text{sgn}(r_1)\text{sgn}(r_2)\min(|r_1| \cdot |r_2|) \\
 \rightarrow \hat{u}_1 &= \begin{cases} 0 & \text{sí } L(u_1) \geq 0 \\ 1 & \text{sí } L(u_1) < 0 \end{cases}
 \end{aligned} \tag{1.141}$$

Donde r_1 y r_2 son los bits recibidos del canal AWGN, $L(u_1)$ representa la creencia que se tiene con respecto al bit transmitido u_1 y \hat{u}_1 es el bit estimado. Se debe mencionar que la creencia en este caso está en función de r_1 y r_2 y que se utiliza la regla del umbral de BPSK para determinar el valor de u_1 . Ahora, para estimar el bit \hat{u}_2 se utiliza la función conocida como g , que viene descrita por la ecuación 1.142.

$$g(r_1, r_2, b) = r_2 + (1 - 2b)r_1 \tag{1.142}$$

Donde b describe al valor del bit \hat{u}_1 . Dada la función, se establecen las condiciones para la búsqueda de \hat{u}_2 por medio de ecuación 1.143.

$$\text{si } \hat{u}_1 = \begin{cases} 0 \rightarrow L(u_2) = r_2 + r_1 \rightarrow x = [u_2 \ u_2] \\ 1 \rightarrow L(u_2) = r_2 - r_1 \rightarrow x = [\bar{u}_2 \ u_2] \end{cases} \quad (1.143)$$

La ecuación 1.141 determina que si el bit estimado \hat{u}_1 es de valor cero, el vector x es una repetición del vector u_2 enviado. Mientras tanto, si el bit estimado es uno, el vector x está constituido por el complemento lógico del bit enviado \bar{u}_2 y el bit u_2 . Note que ambas condiciones tienen bastante lógica si se analiza desde el punto de vista matemático. Por lo tanto, se obtiene el bit estimado \hat{u}_2 por medio de la misma regla de decisión de la creencia. Observe que el proceso descrito corresponde a la decodificación SISO repetitiva, ya que no es más que una suma y resta de vectores recibidos, tal como se describe en este documento en secciones anteriores. Gráficamente, se puede observar que en el diagrama del árbol la estimación de los dos bits u_1 y u_2 se realiza por medio de tres pasos, tal como puede verse en la Figura 1.42. El primer paso donde se trabaja en el nodo de la izquierda con decodificación SISO SPC se determina \hat{u}_1 dados r_1 y r_2 , luego se trabaja en el nodo de la derecha donde se determina el bit estimado \hat{u}_2 y finalmente en el nodo de la combinación de los bits o de la transformación polar donde se estima el valor de \hat{x} .

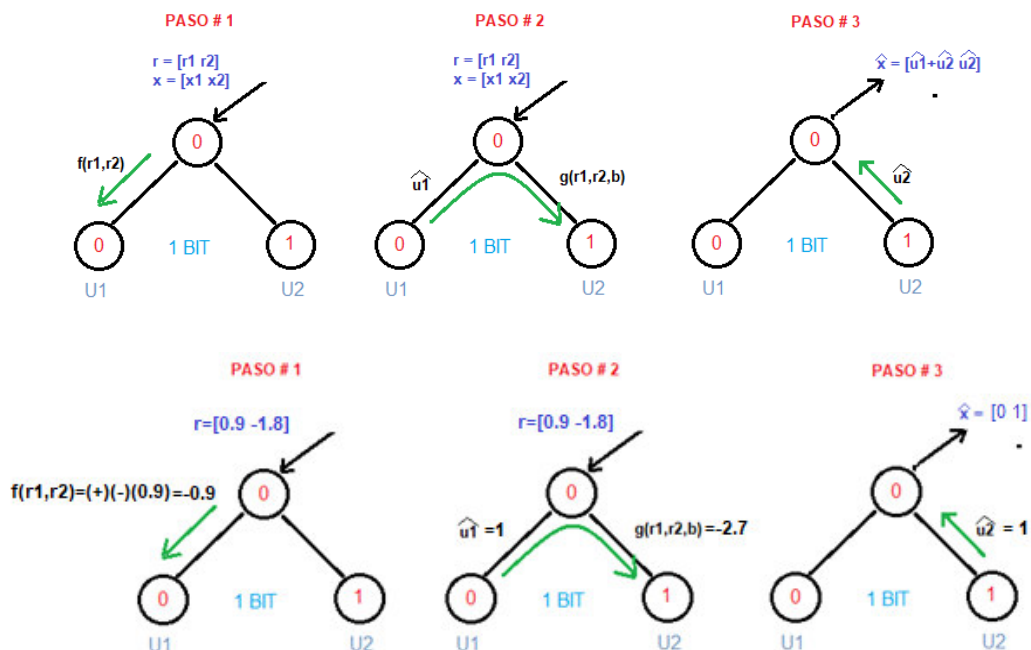


Figura 1.42. Pasos para la estimación de los bits u_1 y u_2 usando codificación polar (2,2)

El proceso explicado puede expandirse hacia un esquema de codificación polar (N, k) , algo que será explicado a continuación.

Decodificación polar (N, k) por cancelación sucesiva simplificada (*Sucesive Simplified Cancellation Decoder – SSC -*)

Considere el ejemplo que se envía un mensaje bajo un esquema de codificación polar (N, k) con modulación BPSK y que se transmite por un canal AWGN [3], [83]. El proceso, para este caso viene dado por el diagrama de la Figura 1.43.

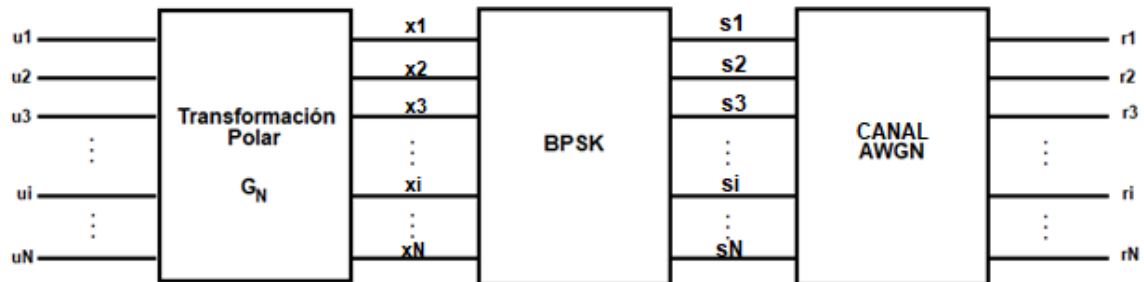


Figura 1.43. Codificación Polar (N, k) con modulación BPSK sobre un canal AWGN

Para este caso se tiene que la codificación depende de la secuencia de confiabilidad, además se debe tomar en cuenta que la transformación polar es función del valor de N y de la longitud k del mensaje que se desea transmitir. El proceso descrito anteriormente para un esquema de codificación $(2,2)$ va a ser muy útil para describir el caso general que es (N, k) . Recuérdese que se había mencionado que la decodificación por medio de cancelaciones sucesivas consiste en tres procesos, tal como se puede observar en la Figura 1.42. Ahora, en el modelo propuesto se realizarán ciertas consideraciones:

- La primera consideración es que el vector recibido es un conjunto de creencias que vienen notadas por el vector $L = [L_1 L_2 L_3 \dots L_M]$, donde M es el número de bits involucrados en el nivel de profundidad de la transformación polar M o regla de combinación de los bits. Así, se tiene el diagrama de árbol binario de la Figura 1.44.

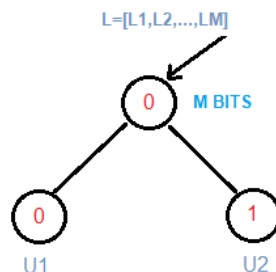


Figura 1.44. Conjunto de creencias que serán evaluadas en un nodo de M bits de un decodificador SISO de N bits

- La segunda consideración es que si se toma en cuenta el primer paso del esquema de codificación (2,2) propuesto anteriormente, la decodificación SISO SPC a la derecha hace que el vector L se divida en dos partes para estimar el bit u_1 , tal que se tiene que la decodificación se evaluará desde L_1 hasta $L_{M/2}$ en una primera parte y luego desde $L_{\frac{M}{2}+1}$ hasta L_M [83]. Dicho esto, la función a evaluar es la descrita en la ecuación 1.144.

$$L(u_1) = f(L_{1:\frac{M}{2}}, L_{1+\frac{M}{2}:L}) \quad (1.144)$$

A diferencia del caso de la codificación (2,2) donde había solo dos valores, ahora se trata de un vector L de varios valores. Para este caso, la función se evalúa, tal como se describe en la ecuación 1.145.

$$f(a_{1:p}, b_{1:p}) = [f(a_1, b_1) f(a_2, b_2) \dots f(a_p, b_p)] \quad (1.145)$$

- La evaluación de un vector L en el decodificador SISO SPC que corresponde a la ecuación 1.139 da como resultado el vector \widehat{u}_1 de la mitad de la longitud de L y que calcula cada entrada de a con su correspondiente de b . En este caso, a_1 sería L_1 y b_1 sería $L_{1+\frac{M}{2}}$. El mismo procedimiento debe ser contemplado para el cálculo del lado derecho que corresponde a un proceso de decodificación SISO de repetición. En este caso, la ecuación 1.146 describe la función que se debe evaluar dado un vector de creencias L .

$$g(L_{1:\frac{M}{2}}, L_{1+\frac{M}{2}:L}, \widehat{u}_1) = [g(L_1, L_{1+\frac{M}{2}}, \widehat{u}_{1,1}) \dots g(L_{\frac{M}{2}}, L_M, \widehat{u}_{1,\frac{M}{2}})] \quad (1.146)$$

- Finalmente, el nodo de orden superior considera la regla de combinación de los bits dada por la transformación polar G_2 que se establece con el vector $[\widehat{u}_1 + \widehat{u}_2, \widehat{u}_2]$. Observe que, hasta ahora no se ha hablado mucho sobre la secuencia de confiabilidad, pero, este juega un papel importante sobre la decisión del valor final que tendrá el o los valores del vector \widehat{u}_i . Se debe indicar que existen dos casos posibles. El primero donde el índice indica una posición fría, en la cual la creencia $L(u_i)$ es ignorada y se asigna el valor de cero a u_i . El segundo escenario corresponde al caso cuando el índice i indica que u_i es un mensaje y es ahí donde se hace una decisión en el umbral que viene dada por la ecuación 1.147 [16].

$$u_i = \begin{cases} 0 & \text{sí } L(u_i) \geq 0 \\ 1 & \text{sí } L(u_i) < 0 \end{cases} \quad (1.147)$$

Como puede observarse, el esquema de codificación polar tanto en codificación como decodificación es muy sencillo, pero, tal como se mencionó al principio, tiene un rendimiento considerado como “mediocre” para palabras código pequeñas y medianas en términos de BER y comienza a mostrar un rendimiento notable para palabras código c de más de un millón de bits. Es por esto, que el estándar *New Radio* 5G utiliza todo lo expuesto como base e incorpora listas, o al menos, este es el método común usado en investigaciones [80]; este es el tema que se verá a continuación.

Introducción a la codificación y decodificación polar (N, k) con cancelación sucesiva simplificada con listas (*Sucesive Simplified Cancelation Decoder with List – SSCL -*)

La decodificación con listas es una idea que está presente desde los años 40's-50's del siglo pasado, sin embargo, de igual manera que en el caso de la codificación LDPC, no pudo ser implementado y reintroducido hasta tiempos modernos debido a las limitaciones tecnológicas de la época; pero, actualmente, gracias a la tecnología está disponible y resulta muy útil para combinarlo con sistemas de decodificación secuenciales o bit a bit.

La idea de la decodificación de listas es sencilla y se basa en producir una lista de posibles palabras código a través de códigos de redundancia cíclica (CRC por sus siglas en inglés) [80], [9]. Es importante que el lector entienda que estos códigos en este caso, no se utilizan como esquema de codificación para detección de errores como comúnmente se ha hecho hasta ahora, sino como una herramienta que permite generar M posibles palabras códigos en función del símbolo recibido r y que luego son evaluadas en función del CRC mencionado para finalmente recuperar la información. De esta forma, esta idea expuesta varía el sistema de comunicación propuesto hasta ahora, tal como puede verse en la Figura 1.45.

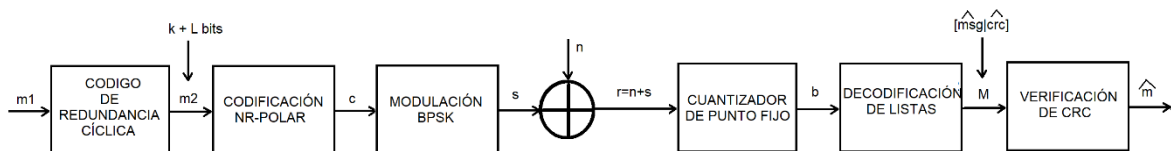


Figura 1.45. Sistema de comunicación con codificación polar, modulación BPSK sobre un canal AWGN y con decodificación en base a la cancelación sucesiva simplificada con lista (SSCL)

Ahora, note que, por la figura, el mensaje m de longitud de $k = 512$ bits que se codificará con el esquema polar no solamente consta del mensaje de longitud de A bits, sino también del código de redundancia cíclica CRC de longitud $L = 12$ bits que de acuerdo al Reporte

Técnico TS 38.212 viene dado por la secuencia de bits $L = [1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1]$ (hay que recordar que esto representa un polinomio) formando la estructura $m = [msg\ | \ crc]$ [84]. En este caso se tiene que el número de posiciones denominadas como frías se mantiene en 512, esto con la intención de mantener la tasa de codificación lo más cercano a $R = 0.5$, y que la polarización del canal y la secuencia de confiabilidad no varíe [80]. Ahora, luego de codificar, modular y transmitir el mensaje por el canal AWGN, el símbolo recibido r pasa a un bloque denominado “Decodificación de Listas”, que se encarga de crear, en función del tamaño de la lista y con el uso de decodificación sucesiva simplificada (SSC), un grupo de M posibles mensajes candidatos de la forma $\hat{m} = [\widehat{msg}\ | \ \widehat{crc}]$, que a su vez se evaluarán en el bloque encargado de la verificación del código de redundancia cíclica. El primer mensaje estimado del conjunto que coincida con el CRC original es seleccionado y se discrimina solamente la parte del mensaje para el usuario final [9]. En este caso, la regla de decisión viene dado por la verificación del CRC.

Ahora, en función a lo descrito, la primera consideración notable es que el rendimiento mejora notablemente en términos de BER por el simple uso de CRC. Pero como desventaja, hay que tomar en cuenta que el sistema presenta mayor complejidad de implementación y necesita más tiempo de procesamiento en la estimación del mensaje porque además de realizar la operación de decodificación sucesiva simplificada (SSC), ahora debe involucrar la generación de listas. Si bien, teóricamente, esto puede presentar un inconveniente, el uso del algoritmo que se verá a continuación para decodificación por listas y el uso del esquema en la práctica han demostrado que esto no presenta un inconveniente mayor para sistemas de comunicación *New Radio* 5G en términos de latencia y velocidad de transmisión [3].

Concepto de métrica decisiva para la generación de listas en la decodificación polar con cancelación sucesiva simplificada con listas (SSCL)

Recuerde que en secciones anteriores se había introducido la idea de decodificación sucesiva simplificada (SSC) por medio del uso de diagramas de árboles binarios, un proceso que puede observarse de mejor forma en la Figura 1.42. Ahora se va a introducir la forma de producir M mensajes candidatos con este tipo de decodificación basado en el concepto de la métrica decisiva o DM por sus siglas en inglés [3], [80], [9]. Para esto, considérese la Figura 1.46 y recuérdese que la decodificación de cancelación sucesiva de un nodo i de profundidad N tiene como entrada el vector de creencias y como salida el vector u_i de valores estimados.

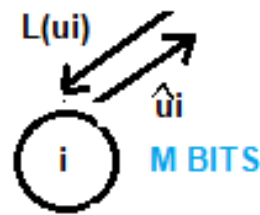


Figura 1.46. Ingreso y salida de datos a través de un nodo i en la decodificación de cancelación sucesiva

Ahora, considere que el mensaje m se transmite en el sistema que se indica en la Figura 1.43 donde el proceso de decodificación llega al nodo i . Debido a la naturaleza secuencial del código, el error suele propagarse de bit a bit, lo que puede ocasionar que el rendimiento del sistema sea afectado notablemente. Frente a esto, una alternativa para asegurar que la palabra decodificada tenga el menor número de errores posibles es almacenar tanto la palabra código estimada a partir de los procesos de decodificación antes descritos en secciones anteriores como su contraparte, tal como se describe en la Figura 1.47. Es decir, si la palabra código estimada en el nodo i por los procesos de decodificación descritos en secciones anteriores es igual al bit 1, el resultado deberá ser almacenado con su contraparte que no fue estimada, el bit 0, generando ya no una sino dos opciones que pasarán al bloque de Verificación de CRC [9].

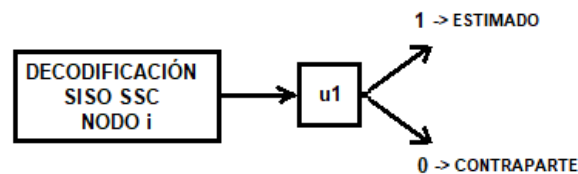


Figura 1.47. Posibles salidas de una decodificación tipo SISO SSC con BPSK

Pero, aquí es natural que el lector se haga la pregunta ¿cómo determinar qué bit o conjunto de bits son la mejor opción para verificar el código de redundancia cíclica?

Recuerde que en decodificación SSC cuando el valor del nodo u_i no está congelado o *freeze* trabaja en función del valor de la creencia que proviene de nodos superiores, en consecuencia, este valor o conjunto de valores son los encargados de determinar si el error se propaga o no. Por lo tanto, si el bit decodificado está en función de los valores de las creencias L , se debe asumir que el proceso de decodificación está bien realizado y no debe sufrir “penalidad”. Pero, en el caso de que el bit que no es elegido en la decodificación, se lo debe conservar de forma paralela para generar la lista u opciones, pero con cierto valor de “penalidad” para recordar que ese bit no fue el correctamente decodificado. Este es el concepto de *métrica decisiva* [80], [9], que viene descrito por la ecuación 1.148.

$$si \begin{cases} L(u_i) \geq 0 : \hat{u}_i = 0 \rightarrow DM_i = 0 \mid u_i = 1 \rightarrow DM_i = |L(u_i)| \\ L(u_i) < 0 : \hat{u}_i = 1 \rightarrow DM_i = 0 \mid u_i = 0 \rightarrow DM_i = |L(u_i)| \end{cases} \quad (1.148)$$

La ecuación anterior describe que en el caso que se asigna una métrica o “penalidad” de valor cero al bit que es producto de la decodificación y que, automáticamente, el bit contrario al de la mencionada decodificación adquiere una métrica denominada penalidad que equivale al valor absoluto de la creencia, algo que puede ser denominado como contra-creencia que se encuentra señalado como rojo en la ecuación 1.148 [9]. Ahora, recuérdese que se había mencionado que el proceso de decodificación con cancelación sucesiva sea cual sea el tipo, es un proceso secuencial bit a bit o vector a vector, por lo tanto, si se decodifica un nodo i que tiene dos opciones o listas decodificadas y dicho término existe en el nivel inferior, entonces en un nodo superior $i + 1$ se generará el doble de opciones o listas, es decir 4 y así en respectivo orden por potencia de 2 [9], tal como puede verse en la Figura 1.48.

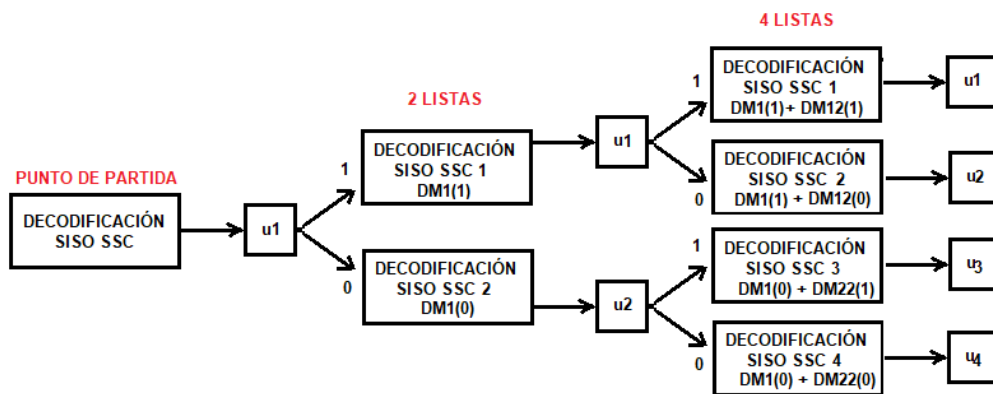


Figura 1.48. Proceso para generación de 4 listas dado un nodo u_i

En este caso, para elegir la mejor decodificación se procede a evaluar el concepto de métrica de camino o *Path Metric*, que tal como se indica en la ecuación 1.150, se basa en la suma de las métricas decisivas de las decodificaciones involucradas⁸ [80].

$$PM = \sum_i DM_i \quad (1.149)$$

⁸ En el ejemplo propuesto de la Figura 1.48, se hace referencia a la suma de “las decodificaciones involucradas” a la suma de las dos métricas decisivas cuando se trabaja con 4 listas. Por ejemplo, el proceso de decodificación SISO SSCL 1 tiene como suma las métricas decisivas $DM1(1)$ y $DM12(1)$

De esta manera, la ecuación 1.150 permite seleccionar el valor más pequeño de estas métricas que se considera la mejor estimación de la decodificación y la primera candidata a ser evaluada en el bloque de verificación CRC.

$$\hat{m}_1 = \text{argmin}(PM) \quad (1.150)$$

Este proceso se repetirá en la lista generada hasta que un mensaje estimado de acuerdo con el orden generado tenga el mismo polinomio CRC en el bloque de verificación [9]. El estándar *New Radio* 5G, a través del Reporte Técnico TS 38.212, no recomienda ningún tamaño de generación, pero las investigaciones realizadas en [80] sugieren que el número de listas debe tener un tamaño de 4 a 8 y debe efectuarse sobre todo el símbolo recibido r y no sobre cada nodo.

2. METODOLOGÍA

El capítulo dos se centra en describir la programación de los bloques de codificación NR-LDPC y NR-Polar con Matlab como software de simulación. Los sistemas de comunicación que se implementarán tienen en común el manejo de modulación BPSK sobre un canal AWGN y cuantización de punto fijo (*fixed-point quantization*) en la recepción y se diferencian por el esquema de codificación que se usará: el esquema de codificación NR-LDPC consta en su recepción de un bloque de decodificación SISO (*Soft-Input Soft-Output*) MINSUM por capas con *Offset* y *Rate Matching*; mientras que el que utiliza codificación NR-Polar consta de un bloque de decodificación SISO SSCL (Cancelación Sucesiva Simplificada con Listas). Además, en el presente capítulo se complementará la descripción de las características técnicas que se tomaron en cuenta y se presentarán los resultados obtenidos.

Para el presente proyecto se utilizó tanto la investigación explicativa como la investigación aplicada. La primera puede observarse en el capítulo uno, donde se explica a fondo las características de los esquemas de codificación NR-LDPC y NR-Polar y de decodificación SISO MINSUM por capas con *Offset* y *Rate Matching* y SISO SSCL. En cambio, el segundo tipo de investigación está presente en la simulación del sistema de comunicaciones inalámbrico.

Debido a que el manejo de los esquemas de codificación se realiza tanto en transmisión como en la recepción, se explicará a detalle cada proceso que se halla involucrado en programación para la implementación de los mencionados esquemas.

2.1. GENERALIDADES

Sistema de comunicación con el esquema de codificación *New Radio* LDPC

Tal como puede verse en la Figura 2.1, la implementación del esquema de codificación *New Radio* LDPC [57], de acuerdo a lo planteado en este estudio, debe incluirse en un sistema de comunicación con modulación BPSK sobre un canal AWGN. El sistema de comunicación además debe contar con cuantización de punto fijo (*fixed-point quantization*) [85] y un decodificador tipo SISO MINSUM de paso mensaje por capas con capacidad de corrección de variables mediante el uso de *OFFSET* [63], [8].



Figura 2.1. Sistema de comunicación a implementarse con el esquema de codificación New Radio LDPC

Sistema de comunicación con el esquema de codificación *New Radio Polar*

Tal como puede apreciarse en la Figura 2.2, de igual manera el esquema de codificación *New Radio Polar* [3] debe implementarse con modulación BPSK sobre un canal AWGN y además debe incorporarse con un bloque de cuantización de punto fijo (*fixed-point quantization*) [85] y un bloque decodificador tipo SISO de Cancelación Sucesiva Simplificada Con Listas (SSCL) [80], [83]. Este bloque de decodificación, tal como se vio en el capítulo anterior, está compuesto del proceso de decodificación como tal y de un bloque de verificación del código de redundancia cíclica.

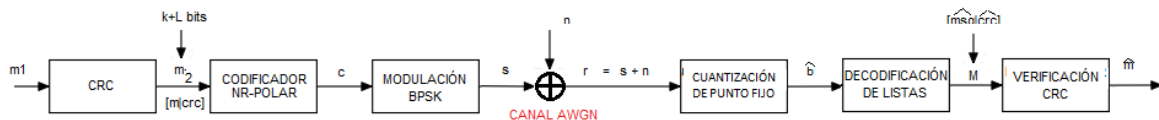


Figura 2.2. Sistema de comunicación propuesto con el esquema de codificación *New Radio*

Software y equipo

La implementación y simulación de los sistemas de comunicación propuestos en la Figura 2.1 y 2.2 se realizarán por medio del software matemático MATLAB R2020a. La plataforma de MATLAB está optimizada para resolver problemas científicos y de ingeniería. El lenguaje de MATLAB, basado en matrices, es la forma más natural del mundo para expresar las matemáticas computacionales. Las gráficas integradas facilitan la visualización de los datos y la obtención de información a partir de ellos. Una vasta biblioteca de herramientas (*Toolboxes*) integradas permiten empezar a trabajar inmediatamente con algoritmos esenciales para su dominio. El entorno de escritorio invita a experimentar, explorar y descubrir. Todas estas herramientas y funciones de MATLAB están probadas rigurosamente y diseñadas para trabajar juntas. La implementación de la simulación de los sistemas de comunicación se hará por medio de un computador con procesador Intel i5-7ma generación con memoria RAM de 4GB

Forma de implementación

Los sistemas de comunicación propuestos en la Figura 2.1 y 2.2 serán implementados por medio de programación de dos *scripts* en lenguaje .m, uno para cada sistema propuesto.

Estos sistemas usarán funciones propias de MATLAB como también algunas creadas de forma propia. Los resultados generados de la simulación serán visualizados en la ventana de comandos (*Command Window*) para que sean posteriormente almacenados de forma manual en un tercer script para que, posteriormente, permita su visualización por medio de gráficas.

Para cada esquema de codificación, el proceso de implementación de los sistemas de comunicación debe ser ordenado en función de la teoría expuesta en el capítulo anterior. La Figura 2.3 y 2.4 muestra el orden de implementación de cada uno de los bloques de los sistemas de comunicación propuestos.

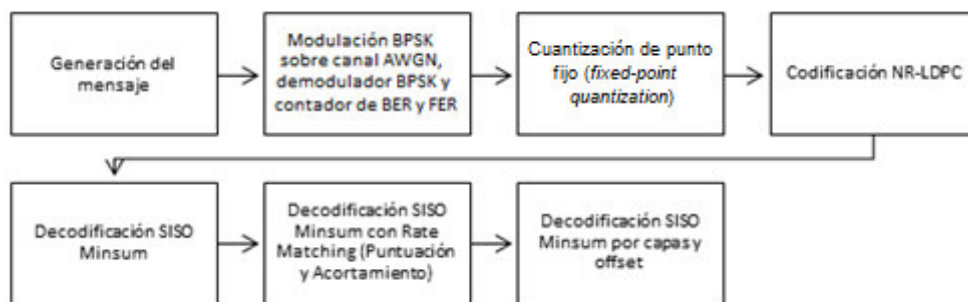


Figura 2.3. Orden de implementación del sistema de comunicación que incorpora el esquema de codificación NR-LDPC

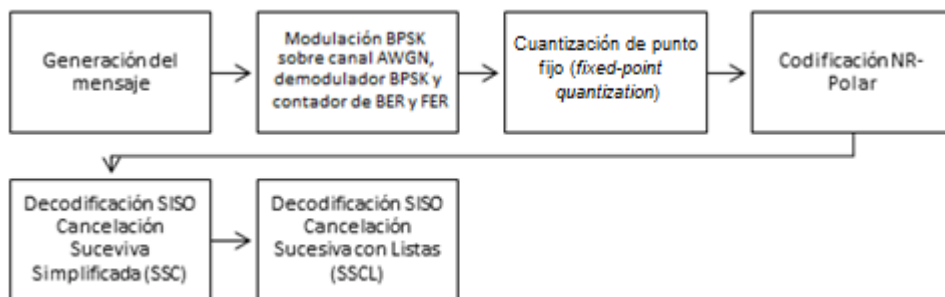


Figura 2.4. Orden de implementación del sistema de comunicación que incorpora el esquema de codificación NR-Polar

2.2. REQUISITOS TÉCNICOS PARA LA SIMULACIÓN DE LOS SISTEMAS DE COMUNICACIÓN PROPUESTOS

Una vez definidos los sistemas a implementar, el software a utilizar y el orden de implementación de los sistemas, el presente segmento realiza una definición de los requisitos técnicos necesarios que deben cumplir los sistemas de comunicación a

implementar. Los mencionados requisitos técnicos están basados en las normativas impuestas por la 3GPP a través del estándar TS 38.212.

2.2.1. GENERACIÓN DEL MENSAJE

Los sistemas de comunicación consideran la creación de N mensajes m de longitud k . La mencionada longitud del mensaje está determinada por las definiciones del estándar técnico TS 38.212 [1].

- En el caso de LDPC, el estándar *New Radio* 5G determina que la longitud del mensaje está en función de la longitud a nivel de columnas de la submatriz A de la matriz/grafó base a utilizar (BG1 o BG2) [2]. En este caso, si se utiliza la matriz base BG1 la longitud del mensaje k será de $22 \cdot Z_c$ bits, caso contrario, si se utiliza la matriz base BG2 la longitud del mensaje k será de $10 \cdot Z_c$ bits.
- En el caso del esquema de codificación Polar, el estándar *New Radio* 5G determina que la longitud del mensaje k es de 512 bits [3]: 12 bits destinados al código de redundancia cíclica y 500 bits al mensaje m como tal. El estándar técnico 38.212 define que el polinomio generador CRC es $L = [1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1]$ [84]. La Figura 2.5 describe el proceso lógico de generación del mensaje m usando códigos de redundancia lógica.

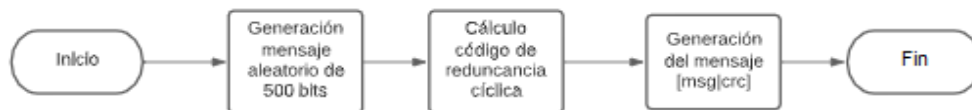


Figura 2.5. Generación de un mensaje de 512 bits para un sistema de comunicación con el esquema de codificación NR-Polar con el uso de códigos de redundancia cíclica (CRC)

2.2.2. MODULACIÓN BPSK Y CANAL AWGN

Para el presente trabajo de titulación el esquema de modulación utilizado será BPSK. La Ecuación 2.1 define la fórmula matemática usada para realizar la codificación en MATLAB.

$$s_i = 1 - 2m_i [w] \quad (2.1)$$

Donde m_i es el bit de la cadena o vector del mensaje a transmitir y que será modulado. Mientras tanto, en demodulación, se usará detector basado por umbral (*Threshold Based Detector*), definido por la Ecuación 1.7 y donde la regla de decisión tiene un valor de 0[W]. Recuérdese que, en el caso del presente trabajo de titulación, el demodulador BPSK está incorporado al final de los decodificadores SISO.

Por su lado, el ruido AWGN está determinado en función del esquema de modulación que se usa, en este caso BPSK. Al ser ruido del tipo aditivo, el canal AWGN en MATLAB no es más que un vector de números generados aleatoriamente menores a uno de misma longitud del vector símbolo transmitido s que está modulado en BPSK multiplicado por la desviación estándar σ que debe sumarse al mencionado vector símbolo s y que da como resultado el vector r . Este proceso asegura además que el ruido AWGN sea independiente de bit a bit, una de las características de este tipo de ruido.

2.2.3. CODIFICACIÓN

El estándar *New Radio* 5G a través del estándar TS 38.212 define la forma como debe ser codificado un mensaje m mediante el uso de los esquemas de codificación LDPC y Polar [1]. Las características se presentan a continuación.

- El estándar técnico TS 38.212 define que para hallar los $n - k$ bits de paridad p del esquema de codificación NR-LDPC se realiza la construcción de una matriz de verificación de paridad H mediante la expansión de dos matrices o grafos base BG1 (46x68) o BG2 (42x52) con el manejo de matrices identidad desplazadas hacia la derecha y de valor nulo dadas por el valor del tamaño de expansión Z_c [57], [2]. Para hallar los bits de paridad, el esquema de codificación NR-LDPC trabaja con la doble diagonal contenida en la submatriz E para determinar los primeros cuatro bloques de bits de paridad y posterior a eso la submatriz C e I para determinar el resto de los bits de paridad.

Además, el estándar exige el uso de *Rate Matching*. En este caso, las dos primeras columnas del grafo base son puntuados por defecto para transmisión y que el uso de acortamiento o *shortening* está en función de la tasa de codificación R que se desee aplicar [8]. La Figura 2.7 define el proceso lógico de la codificación polar para el estándar *New Radio* 5G.

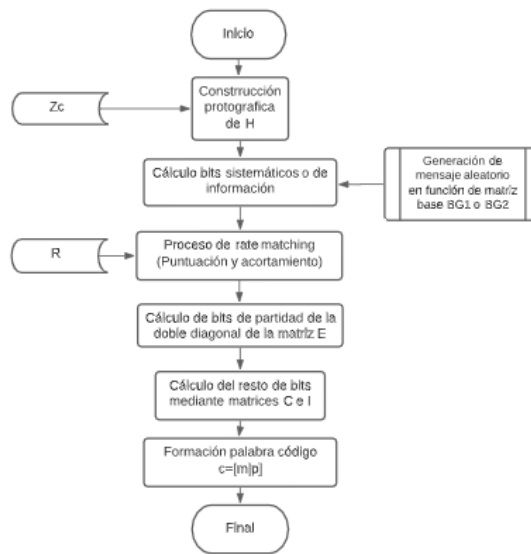


Figura 2.6. Diagrama de flujo del esquema de codificación NR-LDPC

- En el caso de codificación Polar para el estándar *New Radio* 5G, el estándar técnico TS 38.212 define que la codificación se la realizará en base a una secuencia de confiabilidad Q de una longitud de 1024. El mismo estándar determina que las primeras 512 posiciones definidas en la mencionada secuencia son para los bits fríos ($u_i = 0$) y las últimas 512 posiciones están destinadas para el mensaje m compuesto de la información a enviar como tal y el código de redundancia cíclica [79]. La Figura 2.7 define el proceso lógico de la codificación polar para el estándar *New Radio* 5G.

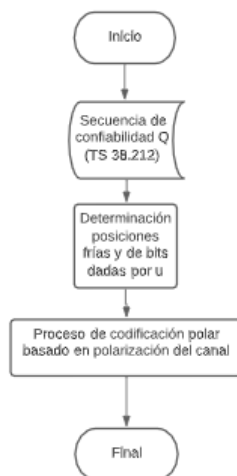


Figura 2. 7. Diagrama de flujo para el proceso de codificación NR-Polar basado en la secuencia de confiabilidad Q del estándar TS 38.212

2.2.4. CUANTIZACIÓN DE PUNTO FIJO (*FIXED-POINT QUANTIZATION*)

En ambos sistemas, la cuantización de punto fijo usará la Ecuación 1.44. Para el caso de los sistemas de codificación propuestos, la mantisa definida para este caso será de $n = 5$ bits y el *Valor Pico* será de $3[W]$. Además, para mejorar la precisión del uso de los datos en la decodificación y estimación de la palabra código \hat{c} en NR-LDPC, se usará una cuantificación de este mismo tipo dentro del decodificador de 7 bits. Este tipo de cuantificación se la justifica debido al uso de iteraciones y para mejorar la precisión en las operaciones en cada una de las iteraciones.

2.2.5. DECODIFICACIÓN

Tal como se explicó en el capítulo anterior, el estándar técnico TS 38.212 no ofrece información o instrucciones de cómo realizar la decodificación de los esquemas de codificación LDPC y Polar para el estándar *New Radio* 5G [3]. Pese a eso, existen varias contribuciones dentro del campo de investigación que sugieren el uso de decodificadores tipo SISO MINSUM combinados con *Rate Matching* [8] y *offset* [73], decodificación por capas y corrección de variables para NR-LDPC y decodificación SISO con cancelación sucesiva simplificada con listas (SSCL) para NR-Polar [80].

- El sistema de comunicación descrito por la Figura 2.1 define el uso de un decodificador MINSUM por capas con offset para el esquema de codificación NR-LDPC, mismo que se caracteriza por ser un proceso iterativo de operación sucesiva de filas y columnas por medio del uso de decodificadores SISO SPC Repetitivo, tal como se describe de forma lógica en la Figura 2.8.

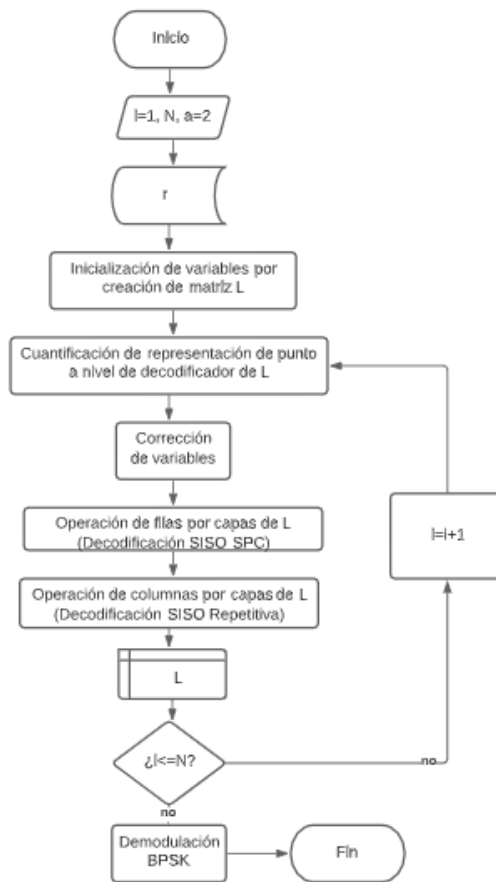


Figura 2.8. Diagrama de flujo para la decodificación SISO MINSUM por Capas con Offset

Nótese que, al ser un proceso iterativo de paso de mensaje, el sistema necesita de forma predefinida el número de veces que se realizará la operación de decodificación. Por medio de diversas investigaciones se recomienda que sea entre 6 a 8 veces [7]. En el caso de corrección de variables, el manejo de *Offset* solo manejará la contante a de la Ecuación 1.122. Para este caso, se usará el valor de corrección $a = 2$, valor que ha sido recomendado en [73].

- El sistema de comunicación descrito por la Figura 2.2 define el uso de un decodificador SISO con cancelación sucesiva simplificada de listas (SSCL) [80], mismo que parte de un proceso de decodificación SISO con cancelación sucesiva simplificada (SSC) que es un proceso que consta de tres pasos descrita de forma general por la Figura 1.42 y matemáticamente desde la Ecuación 1.142 hasta la 1.146 [81]. Por su lado, la generación de listas trabaja con el manejo de creencias y contracreencias definidas en el capítulo previo desde la Ecuación 1.147 hasta la 1.149, que tienen como objetivo crear una palabra código estimada \hat{c} con el bloque de verificación del código de

redundancia cíclica cuyo polinomio generador $L = [1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1]$ [9]. El mencionado proceso se explica de mejor forma mediante el diagrama de flujo de la Figura 2.9, donde observe que el proceso siempre determina una palabra código estimada que cumpla con la condición de verificación CRC.

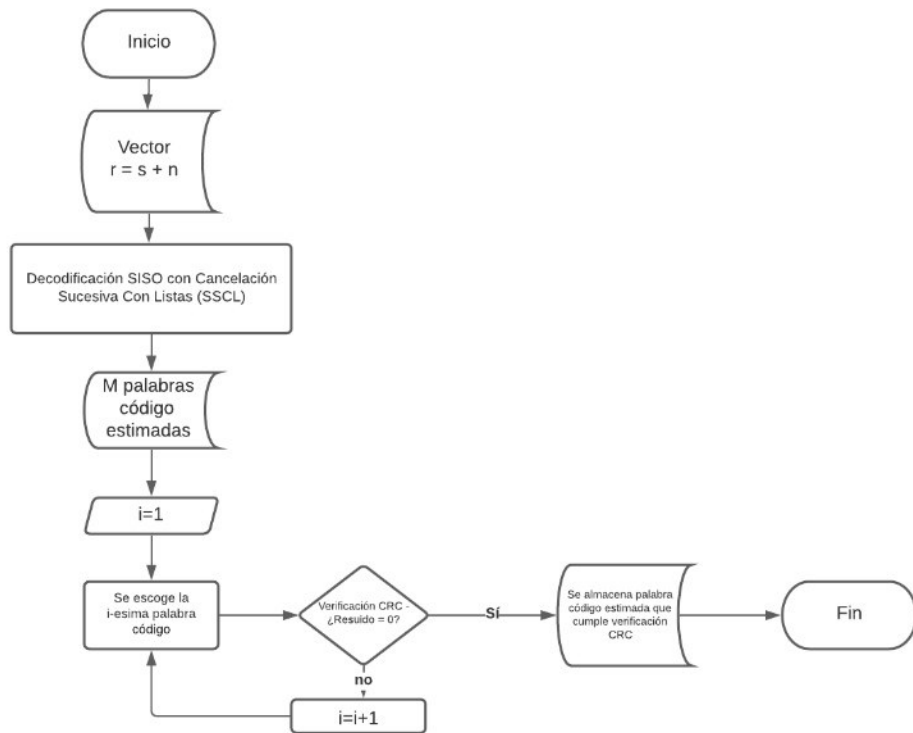


Figura 2.9. Diagrama de flujo para la decodificación NR-Polar mediante el uso de SISO SSCL

2.2.6. CONTEO DE ERRORES Y TIEMPO DE PROCESAMIENTO

Debido a que el sistema tiene como objetivo medir el rendimiento de los esquemas de codificación tanto en tasa de bits de errores (BER), bloques errados (FER) y en términos de tiempo de procesamiento, los sistemas de comunicación de la Figura 2.1 y 2.2 deben incorporar un mecanismo matemático para poder realizar el conteo de errores, tanto a nivel de bits como de bloques errados y calcular el BER y FER. Por su lado, el tiempo de procesamiento debe contar el tiempo de procesamiento total y de codificación y decodificación por bloque por medio del comando TIC-TOC de MATLAB [86].

2.2.7. VISUALIZACIÓN DE RESULTADOS Y ALMACENAMIENTO

Los sistemas simulados en MATLAB que están definidos en la Figura 2.1 y 2.2 deben calcular FER, BER y el tiempo de procesamiento. A más de eso, la simulación debe ser capaz de mostrar en un vector estos resultados junto al E_b/N_o utilizado en [dB] y el número de bloques transmitidos. Este vector debe ser almacenado en el tercer script para que posteriormente puedan ser visualizados de la forma como se muestra en la Figura 2.10 y 2.11.

$\frac{E_b}{N_o}$ [dB]	BER	FER	Tiempo de codificación por bloque	Tiempo de de codificación por bloque	Tiempo de de procesamiento total	Número de bloques [N]	Número de iteraciones
------------------------	-----	-----	-----------------------------------	--------------------------------------	----------------------------------	-----------------------	-----------------------

Figura 2.10. Variables resultantes que serán almacenadas posterior a la simulación del esquema de codificación NR-LDPC

$\frac{E_b}{N_o}$ [dB]	BER	FER	Tiempo de codificación por bloque	Tiempo de de codificación por bloque	Tiempo de de procesamiento total	Número de bloques [N]	Número de listas
------------------------	-----	-----	-----------------------------------	--------------------------------------	----------------------------------	-----------------------	------------------

Figura 2.11. Variables resultantes que serán almacenadas posterior a la simulación del esquema de codificación NR-Polar

Los resultados deben ser almacenados en un tercer script que permitirán la visualización por medio de gráficas bidimensionales con su respectiva leyenda, título y ejes rotulados por medio de gráfico semilogarítmico y lineales, de acuerdo como se considere el caso. La idea con esto es que el presente trabajo de titulación pueda proporcionar información relevante sobre estos esquemas de codificación.

2.3. IMPLEMENTACIÓN DE LOS SISTEMAS DE COMUNICACIÓN

2.3.1. VARIABLES PREDEFINIDAS

El proceso de implementación de los sistemas de comunicación de la Figura 2.1 y 2.2. inicia desde la definición de variables necesarias para su funcionamiento. La Tabla 2.1 y 2.2 clasifica de forma general a las mencionadas variables de entrada dependiendo del esquema que se vaya a utilizar dependiendo el tipo de codificación.

Tabla 2.1. Variables de entrada para el sistema de comunicación con codificación NR-LDPC

Magnitud	Símbolo	Bloque al que pertenece
Número de bloques	N	Transmisión
Relación E_b/N_0	E_b/N_0	Canal AWGN
Grafo Base (1 o 2)	BG	Codificación
Tamaño de expansión	Z_c	Codificación
Mantisa	n	Cuantificación
Valor pico	N/A	Cuantificación
Iteraciones	n	Decodificación
Offset	a	Decodificación / Corrección de variables

Tabla 2.2. Variables de entrada para el sistema de comunicación con codificación NR-Polar

Magnitud	Símbolo	Bloque al que pertenece
Número de bloques	N	Transmisión
Relación E_b/N_0	E_b/N_0	Canal AWGN
Código de redundancia cíclica	CRC	Codificación y decodificación
Secuencia de confiabilidad	Q	Codificación y decodificación
Mantisa	n	Cuantificación
Valor pico	N/A	Cuantificación
Número de Listas	nL	Decodificación

2.3.2. IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN CON MODULACIÓN BPSK SOBRE CANAL AWGN Y CUANTIFICACIÓN CON REPRESENTACIÓN DE PUNTO FIJO

El primer paso para lograr la implementación de los sistemas propuestos es la codificación de un sistema que tenga la capacidad de generar mensajes de N bloques de n bits, que puedan ser modulados en BPSK, transmitidos por un canal AWGN y posteriormente cuantificados y demodulados. La Figura 2.12 muestra los pasos lógicos para la programación de este proceso.



Figura 2.12. Pasos lógicos para la programación de un sistema de comunicación con modulación BPSK y cuantificación con representación de punto fijo sobre un canal AWGN

La programación de este sistema de comunicación viene dada por el segmento de código 2.1, en donde se señalan las posiciones en las que se colocarán los bloques de codificación y decodificación para los esquemas NR-LDPC y NR-Polar, que serán implementados mediante funciones.

```

%Inicio Segmento de Código

clear all
close all
clc

%Inicio contador de tiempo global
tinicio=tic;

R=1; %Defino tasa de codificación
EbNodB=3.5; %Defino Eb/No en [dB]
EbNo=10^(EbNodB/10); %Cambio Eb/No unidimensional
sigma=sqrt(1/(2*R*EbNo)); %Calculo sigma
%Condiciones para cuantización arreglada de punto
rmax=3; %Límite máximo para cuantificar y verificar valor recibido
maxqr=31; %Valor de 5 bits para cuantizar la magnitud

Nblocks=1000; %Defino número de bloques
Nbitrrs=0; %Número de bits errados
Nblkrrs=0; %Número de bloques errados

for i=1:Nblocks %Generación de bloque
    msg=randi([0 1],1,k); %Generación de bits

    %Se coloca bloque de codificación NR-LDPC o NR-Polar

    s = 1 - 2*msg; %Modulación BPSK
    r = s + (sigma*randn(1,N)); %Defino canal AWGN

    %Cuantificación de punto fijo suponiendo que se recibe r
    rq=floor(r/rmax*maxqr); %Determino el valor cuantificado de
    la magnitud - Se convierte en integral
    rq(rq>maxqr)=maxqr; %Si el valor excede maxqr=31
    entonces se asigna el valor de 31 - Clipping
    rq(rq<-(maxqr+1))=-(maxqr+1); %Si el valor excede
    maxqr=-31 entonces se asigna el valor de -31 - Clipping

    %Se coloca bloque de decodificación NR-LDPC o NR-Polar
    msg_cap=(rq<0); %Decisión / Demodulación BPSK

    Nrrs = sum(msg~=msg_cap); %Calculo de número de errores

    if Nrrs>0 %Calculo de número de errores
        Nbitrrs = Nbitrrs + Nrrs;
        Nblkrrs=Nblkrrs+1;
    end
end

end

BER_sim=Nrrs/N/Nblocks; %Calculo BER
FER_sim=Nblkrrs/Nblocks; %Calculo FER

%Cuento tiempo de procesamiento codificador y decodificador por bloque y global
%tcodfinal=sum(tcod)/Nblocks;
%tdecfinal=sum(tdeco)/Nblocks;
tfinal=toc(tinicio);

%Visualizo resultados

```

```
disp([EbNodB BER_sim FER_sim tcodfinal tdecofinal tfinal Nblocks MaxIter/nL])
%Fin de Segmento de Código
```

Segmento de Código 2.1. Sistema de comunicación con modulación BPSK sobre un canal AWGN y cuantificación con representación de punto fijo

2.3.3. IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN CON EL ESQUEMA DE CODIFICACIÓN NR-LDPC

Construcción protográfica⁹ y multiplicación

Primero, recuérdese que se debe realizar la construcción protográfica en función de la matriz base que se tiene [59]. Este proceso trabaja con matrices identidad desplazadas y matrices nulas de dimensión z_c que corresponde al tamaño de expansión [57].

Tomando en cuenta que la dimensión de la matriz identidad desplazada o nula es $z_c \times z_c$ y que la longitud del mensaje como vector es $z_c \times 1$, el resultado de la multiplicación va a ser de dimensiones $z_c \times 1$. Ahora, observe el ejemplo de la Ecuación 2.2 para caracterizar esta operación, donde el desplazamiento es $d = 1$ posición a la izquierda.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_3 \\ x_1 \end{bmatrix} \quad (2.2)$$

Se puede ver que el desplazamiento tiene como resultado el vector mensaje desplazado d posiciones a la izquierda, donde d es el valor del desplazamiento a realizar, si es que existe. Si se lleva este ejemplo de forma generalizada puede ser implementado a manera de código tal como puede verse en el Segmento de Código 2.2.

```
%Inicio Segmento de código

function y = mul_sh(x,k)

%x es el bloque de entrada
%k es el movimiento de la matriz identidad
%y es la salida

if(k== -1)
    y= zeros(1,length(x)); %Multiplicación por matriz nula
else
    y=[x(k+1:end) x(1:k)]; %Multiplicación por la matriz I desplazada
end
%Fin segmento de código
```

Segmento de Código 2.2. Construcción Protográfica mediante la implementación de la función mul_sh

⁹ La codificación protográfica combina la idea de los códigos tipo multilínea con un caso particular de baja complejidad estructural de decodificadores por hardware simplifica el proceso de decodificación paralela por hardware.

Implementación del bloque de codificación NR-LDPC

El esquema de codificación NR-LDPC realiza el cálculo de bits de paridad por el uso de la estructura de la doble diagonal E y de la matriz identidad I [3].

Si se verifica la estructura de los grafos base BG1 y BG2 en NR-LDPC definidos en el estándar técnico TS 38.212 se puede observar que si se realiza la cancelación de todos los elementos de la estructura de la doble diagonal E , tal como se realizó en la Ecuación 1.81, el cálculo del bit de paridad p_1 está en función de la segunda o tercera fila de la primera columna, ya que uno de estos elementos siempre será nulo. De esta manera, los elementos de la doble diagonal correspondientes a la primera y cuarta fila están hechos para simplificarse entre ellos a propósito para el cálculo de p_1 [2]. De manera lógica, la Figura 2.13 muestra el proceso de codificación de este segmento.

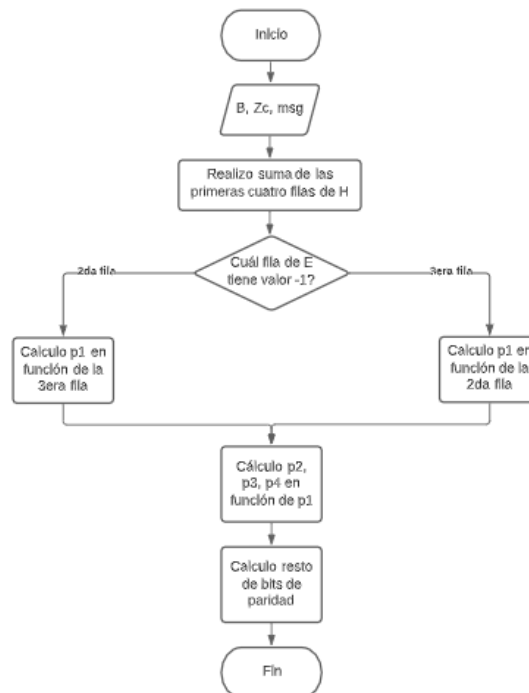


Figura 2.13. Diagrama de cálculo de los bits de paridad en la codificación NR-LDPC

El cálculo de p_1 es considerado el proceso crítico en la codificación NR-LDPC, ya que este permite calcular el resto de los bits de paridad p_2, p_3, p_4 y de la submatriz I [2]. Para esto se procede a realizar un proceso de reemplazo de variables. El Segmento de Código 2.3 define una función para codificación la LDPC del estándar *New Radio* 5G.

```

%Inicio Segmento de Código

function cword=nrlldpc(B,z,msg)
%B es la matriz base
%z es el factor de expansión
%msg: vector de mensaje, length= (#columnas(B)-#filas(B))*z
%cword: vector de la palabra de código, longitud=#columnas(B)*z

[m,n]=size(B);

cword=zeros(1,n*z); %Formo el vector de la palabra código
cword(1:(n-m)*z)=msg; %Asigno el mensaje de 1 a hasta el tamaño del mensaje (en
función de z y de los bloques de información)

%Codificación de la doble diagonal: se procede a encontrar p1, p2, p3 y p4

temp=zeros(1,z); %Creo una matriz basado en z

%Se procede a hacer la suma de las 4 primeras filas de la doble diagonal de la
matriz base. La siguiente operación lo que realiza es realizar la suma de la
información de las primeras 4 filas para crear un sistema de ecuaciones
lineales.

for i=1:4
for j=1:n-m
temp=mod(temp + mul_sh(msg((j-1)*z+1:j*z),B(i,j)),2);
end
end

%Se verifica la segunda o tercera columna para hallar p1
if B(2,n-m+1)==-1
p1_sh=B(3,n-m+1);
else
p1_sh=B(2,n-m+1);
end

%Se procede a encontrar p1: recuérdese que esto está en función de z y del
desplazamiento.
cword((n-m)*z+1:(n-m+1)*z) = mul_sh(temp,z-p1_sh);

%Para encontrar p2 y p3 recuérdese que se debe hacer en función de p1
for i=1:3
temp=zeros(1,z);
for j=1:n-m+i
temp=mod(temp+mul_sh(cword((j-1)*z+1:j*z),B(i,j)),2);
end
cword((n-m+i)*z+1:(n-m+i+1)*z)=temp;
end

%Las paridades restantes se realizan el cálculo desde la fila y columna 5 de
paridad en la matriz generadora
for i=5:m
temp=zeros(1,z);
for j=1:n-m+4
temp=mod(temp+mul_sh(cword((j-1)*z+1:j*z),B(i,j)),2);
end
cword((n-m+i-1)*z+1:(n-m+i)*z)=temp;
end

%Fin del Segmento de Código

```

Segmento de Código 2.3. Bloque de Codificación NR-LDPC mediante la implementación de la función nrlldpc

Implementación del bloque de decodificación SISO MINSUM por capas, *Offset* y *Rate Matching*

Recuérdese que los decodificadores SISO trabajan con creencias, en este caso, la creencia inicial viene dada por el vector recibido que proviene del canal AWGN. Así mismo, se tiene que la decodificación en NR-LDPC es un proceso que se debe realizar por n ocasiones, correspondiente al número de iteraciones y por capas, algo que obliga al uso de al menos dos bucles condicionales *while* o *for*. En este caso, se usará un bucle general *while* para contar el número de iteraciones y para los procesos internos se usará el bucle *for*.

Ahora, recuérdese que el proceso de decodificación incluye un proceso de iniciación donde se necesita crear una matriz de almacenamiento denominada L ; lastimosamente, si esto se codifica en software de simulación, las dimensiones de H son muy grandes y el tiempo de procesamiento aumenta, motivo por lo cual es mejor trabajar con valores distintos de -1 debido a la naturaleza propia de H en LDPC (matrices tipo *sparse*). Así mismo, el proceso exige la creación de una matriz de almacenamiento de creencias definido como R , donde se actualicen los resultados temporales de cada iteración. Definidas las variables y las consideraciones, el proceso de decodificación SISO MINSUM viene dado por tres operaciones que se repiten de manera iterativa [7] y que se describen a continuación:

- Restar R de L que corresponde a la actualización del registro. Este registro como se puede apreciar parte del conjunto inicial de creencias.
- Realizar la operación de MINSUM: correspondiente al proceso de decodificación SISO SPC.
- Sumar las dos operaciones previas: correspondiente al proceso de decodificación SISO Repetitivo. Actualización de la creencia.

Ahora, se considerará la introducción de la corrección de variables para lo cual es necesario recordar que el valor de *offset* es igual a 2 [73]. En este caso, la operación de corrección se realizará sobre los valores mínimos establecidos en la operación MINSUM. La operación viene dada por la Ecuación 2.3 para el valor mínimo 1.

$$min1 = min1 - offset \quad (2.3)$$

Nótese que, al ser valores absolutos, si el valor de la operación es menor a 0, el estándar asigna el valor 0 automáticamente para $min1$. Esto ocurre de manera similar para $min2$.

Finalmente, la introducción de *Rate Matching* con el uso de puntuación y acortamiento propuesto en el estándar TS 38.212 considera que se utiliza puntuación en las dos primeras columnas de la matriz de verificación H, las cuales no se transmiten [8]. Esto se soluciona en la implementación asignando valor 0 a dichas columnas. En el caso del proceso de puntuación, este viene dado por la tasa de codificación y para ambos casos se debe tener en cuenta el número de filas y columnas que se transmiten. En forma lógica, los procesos descritos pueden ser detallados de mejor forma a través de la Figura 2.14.

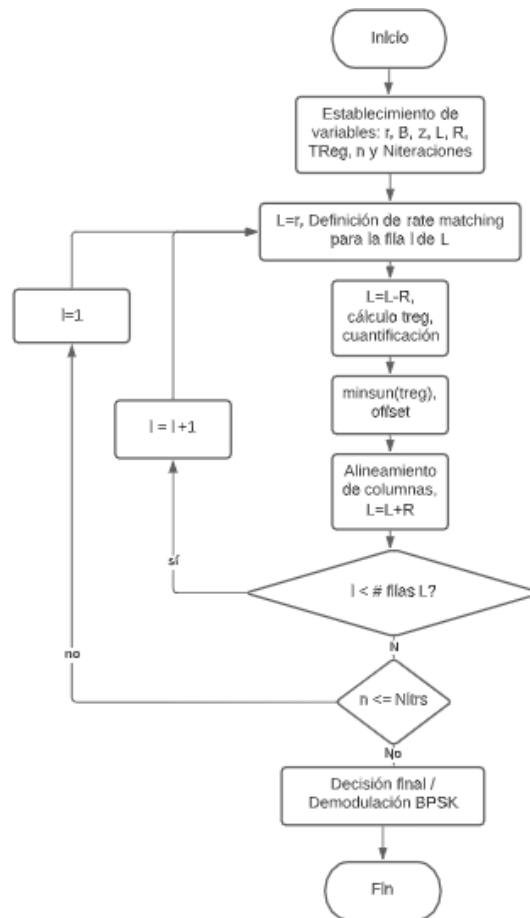


Figura 2.14. Diagrama de flujo de la decodificación SISO MINSUM con capas, OFFSET y con Rate Matching

El Segmento de Código 2.4 define el proceso de decodificación SISO MINSUM por capas, OFFSET y Rate Matching.

```

%Inicio Segmento de Código
function
msg_tilde=sisoiterative(B,offset,maxqr,maxqL,k,mbRM,nbRM,Slen,treg,r,z,MaxIter)
%- Se realiza decodificación por repetición y SPC
%Se ponen condiciones para la decodificación
R=zeros(Slen,z); %Creación del vector de almacenamiento para cada entrada
de B diferente a -1
L=r; %Creencia total
itr=0; %Contador de iteraciones

while itr<MaxIter
    %Contador del vector de almacenamiento
    Ri=0;
    %Cada fila se considera una capa
    for lyr=1:mbRM %Se crea codificación por capas
        ti=0;
    %Se crea un contador para las filas del registro temporal distintas a -1
        for col=1:nbRM
    %Hacemos un mapeo elemento por elemento de cada fila viendo si es diferente de
    -1
            if B(lyr,col)~=-1
                ti=ti+1;
                Ri=Ri+1;
                %Substracción L-R
                L((col-1)*z+1:col*z)=L((col-1)*z+1:col*z)-R(Ri,:);
                %Realizo L-R en grupos de z elementos por fila
                %Alineamiento y almacenamiento en treg
                temp=mul_sh(L((col-1)*z+1:col*z),B(lyr,col));
                %Adapto el dato a la identidad trasladada de la matriz base
                temp(temp>maxqr)=maxqr; %Realizo
    cuantización si el valor excede maxqr, esto asegura que los valores almacenados
    estén entre -32 y 31
                temp(temp<-(maxqr+1))=-(maxqr+1);
                treg(ti,:)=temp;
            end
        end
        %Operación Minsum -> Decodificación SISO SPC
        for j=1:z
            [min1,pos]=min(abs(treg(1:ti,j)));
    %Primer mínimo y su posición del registro temporal
            min2=min(abs(treg([1:pos-1 pos+1:ti],j)));
    %Segundo mínimo y su posición del registro temporal
            S = 2*(treg(1:ti,j)>=0)-1;
    %Calculo signo de la fila
            parity=prod(S);
    %Calculo paridad -> -1 o 1
            min1=min1-offset;
    %Realizo la disminución de valor de DC en la cuantización

            if min1<offset
                min1=0;
            end

            min2=min2-offset;
            if min2<offset
                min2=0;
            end

            treg(1:ti,j)=min1;
    %Coloco valor abs min1 a todos los valores distintos a min1
            treg(pos,j)=min2;
    %Coloco valor abs min2 en donde está valor min1
            treg(1:ti,j)=parity*S.*treg(1:ti,j);
    %Asignación de signos a la matriz
        end
    end
    %Alineamiento de columnas, suma y almacenamiento en R

```

```

%Es decir (L-R)+minsum(valor)
Ri=Ri-ti;
ti=0;

for col=1:nbRM
    if B(lyr,col)~=-1
        Ri=Ri+1;
        ti=ti+1;
        %Alineamiento de columnas
        R(Ri,:)=mul_sh(treg(ti,:),z-B(lyr,col));
        %Adición de columnas y cuantización arreglada
        temp=L((col-1)*z+1:col*z)+R(Ri,:);
        temp(temp>maxqL)=maxqL;
        temp(temp<-(maxqL+1))=-(maxqL+1);
        L((col-1)*z+1:col*z)=temp;
    end
end
end
msg_tilde=real(L(1:k))<0;
%Decision
itr=itr+1; %Avanzo a la siguiente
iteración

end
end
%Fin del Segmento de Código

```

Segmento de Código 2.4. Decodificación SISO tipo MINSUM por capas con Rate Matching y uso de corrección de variables con *Offset* mediante la implementación de la función *sisoiterative*

Implementación

La implementación del sistema de comunicación en código viene dada por la combinación del sistema de comunicación con modulación BPSK sobre un canal AWGN (Segmento de Código 2.1) como eje fundamental de la codificación. El segmento de código debe acondicionarse para trabajar con los bloques de codificación NR-LDPC definido por la función *nrlldpc* (Segmento de Código 2.3) y el bloque de decodificación SISO tipo MINSUM por capas, *Rate Matching* y *OFFSET* definido por la función *sisoiterative* (Segmento de Código 2.4). El resultado final de esta implementación se presenta el Anexo A.

2.3.4. IMPLEMENTACIÓN DEL SISTEMA DE COMUNICACIÓN CON EL ESQUEMA DE CODIFICACIÓN NR-POLAR

Implementación del esquema de codificación NR-Polar

Tal como se vio en el capítulo uno, la codificación NR-Polar se realiza a partir de la idea de polarización de canal, misma que define el uso de una secuencia de confiabilidad Q encargada de establecer las posiciones frías o menos confiables para transmitir un bit de mensaje (bits de valor 0) y las posiciones donde se los puede transmitir. Para el estándar

NR-LDPC, la secuencia, cuya longitud es de 1024 bits, viene establecida dentro del estándar técnico 38.212 [79]. El Segmento de Código 2.5 establece la forma de selección de las mencionadas posiciones en función de los valores de bits de información a transmitir y de la longitud del mensaje.

```

%Inicio Segmento de Código

function [Q1,F]=reability(N,K)

    %Q1 son las posiciones asignadas al mensaje
    %F son las posiciones frías
    %N es el tamaño de la palabra código
    %K es el tamaño del mensaje
    %Q es la secuencia de confiabilidad establecido por el TS 38.212

    Q=[0 1 2 4 8 16 32 3 5 64 9 6 17 10 18 128 12 33 65 20 256 34 24 36 7 129
66 512 11 40 68 130 ...];

    Q1=Q(Q<=N); %Selecciono los elementos menores a N en la matriz de
confiabilidad en función de la longitud del mensaje
    F = Q1(1:N-K); %Selecciono posiciones frías

end

%Fin del Segmento de Código

```

Segmento de Código 2.5. Establecimiento de las posiciones frías y de mensaje mediante el uso de la secuencia de confiabilidad Q mediante la implementación de la función *reability*

Ahora, tómese en cuenta la Figura 2.15 para la implementación de la codificación. Nótese que el vector $u^{(4)}$ en sus primeros dos elementos es una suma lógica de las dos primeras entradas que forman los dos vectores $u^{(2)}$ del nivel inferior, mientras tanto, el segundo elemento es una suma lógica de las dos entradas que forman los dos vectores $u^{(2)}$. Finalmente, los dos elementos restantes que complementan el vector corresponden de forma exclusiva al vector $u^{(2)}$ del lado derecho [83].

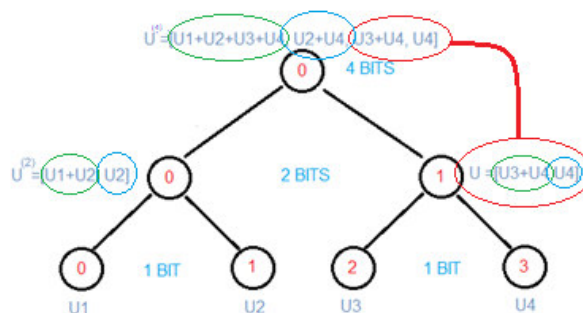


Figura 2.15. Explicación del diagrama de árbol binario para codificación NR-Polar en MATLAB

Se puede notar, por lo mencionado, que se tiene una combinación de hasta 2^i bits por cada nivel, donde i es el nivel más bajo y este se incrementa, mientras más cerca estemos de la raíz. Así mismo, obsérvese que el número de bits no cambia más allá de que se combinan de forma progresiva. Esta consideración es importante, ya que solamente se debe trabajar con un vector U de longitud fija. La Figura 2.16 hace alusión al orden lógico de implementación del esquema de codificación NR-Polar denominado codificación por pares.

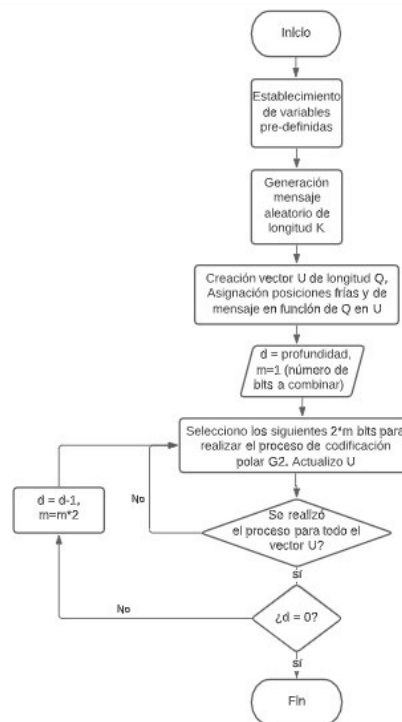


Figura 2.16. Diagrama de Flujo para codificación NR-Polar

La implementación del presente diagrama viene dada por el Segmento de Código 2.6.

```

%Inicio Segmento de Código - Codificación NR-Polar

function cword=nrpolar(N,n,K,Q1,msg)

u=zeros(1,N); %Creo
matriz u, la correspondiente a crear contener la codificación
u(Q1(N-K+1:end))=msg;
m=1; %Número
de bits que se combinarán
%Realizo la codificación por pares y con el uso de XOR

for d=n-1:-1:0
    for i=1:2*m:N
        a=u(i:i+m-1);
        b=u(i+m:i+2*m-1);
        u(i:i+2*m-1)=[mod(a+b,2) b];
    end
    m=m*2;
end
end
  
```

```

%Creo la palabra código en función de la matriz u
cword=u;

end
%Fin del Segmento de Código

```

Segmento de Código 2.6. Función que permite la codificación NR-Polar

Implementación del esquema de decodificación SISO por cancelación sucesiva simplificada (SSC)

El proceso de decodificación SISO implica que, a más del uso de decodificación SISO SPC y repetitivo, el vector inicial $r = U^{(1024)}$ con 1024 creencias, que está en la raíz, se divide en su primer nivel de profundidad en dos nodos $U^{(512)}$ con 512 creencias cada uno y así sucesivamente hasta llegar al nivel 10 donde habrá 1024 nodos con un bit $U^{(1)}$. Esto significa que los N bits deben almacenarse en un vector de $d + 1$ niveles. Nótese que todo este proceso crea la necesidad de implementar un vector de dos dimensiones que permita conocer el nodo, profundidad y número de creencias que se están trabajando y que permita crear un índice para un correcto manejo del vector de creencias [79]. Esto se puede dar por medio de la Ecuación 2.4 para un nodo i a una profundidad d .

$$Creencias\ por\ nodo = [d, 2^{n-d} \cdot i, 2^{n-d} \cdot i - 1] \quad (2.4)$$

Lo anterior permite que cada nodo $u^{(i)}$ a una profundidad d tenga asignado su número de creencias. Ahora nótese por la Figura 2.23 que el proceso descrito ordenado por pasos recorre el árbol ciertos nodos y niveles del árbol binario más de una vez, algo que se puede observar en la Figura 1.17 puede ocasionar problemas si no se toman las precauciones necesarias.

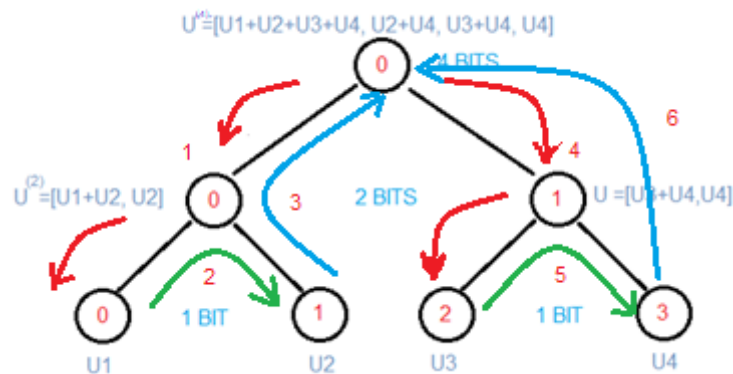


Figura 2.17. Pasos para la decodificación por SISO SSC utilizando decodificadores SISO SPC y repetitivo para una transformación polar de 4 bits

Para esto, es necesario que la codificación reconozca que pasos ya se han ejecutado en el árbol y si es necesario que se ejecute uno más. Esto se puede realizar mediante el uso de estados numéricos en un vector que contenga todos los nodos que pertenecen al sistema en el cual se señale que paso se ha realizado. Para esto, se debe implementar un vector con el número de nodos de un árbol binario, mismo que está en función del número de niveles de profundidad d de acuerdo con la Ecuación 2.5.

$$\# \text{ de nodos} = 2^{d+1} - 1 \quad (2.5)$$

Estas consideraciones son básicas y deben ser previas a la codificación como tal. En cuanto a los estados utilizados, la Tabla 2.3 los describe

Tabla 2.3. Estados que se deben manejar en la implementación de la decodificación SSC

Número	Estado
0	Se debe realizar la operación de decodificación SISO SPC
1	Se debe realizar la operación de decodificación SISO Rep.
2	Se debe realizar la estimación en el nodo central

El Segmento de Código 2.7 se trata de la implementación de la decodificación SISO SSC para la codificación NR-Polar.

```
%Inicio Segmento de Código
function msg_tilde=sisossc(n,N,K,Q1,F,r)
%Condiciones
L=zeros(n+1,N);           %Creencias
ns=zeros(1,2*N-1);       %Vector de estados del nodo
u_tilde=zeros(n+1,N);    %Decisiones

L(1,:)=r;                 %Creencias de la raíz o del nodo 0/profundidad 0

%Establecimiento de funciones f y g para codificación SPC y Repetitiva
f=@(a,b) (1-2*(a<0)).*(1-2*(b<0)).*min(abs(a),abs(b));
g=@(a,b,c) b+(1-2*c).*a;

%Condiciones iniciales del árbol en la raíz
nodo=0;
profundidad=0;

done=0;
while done==0             %Condición que indica si todos los bits
    fueron decodificados
        if profundidad==n %Condición que indica si hemos o no
            llegado a la profundidad más baja
                if any(F==(nodo+1)) %Crea condición si el nodo está congelado =
                    0
                    u_tilde(n+1,nodo+1)=0;
                elseif L(n+1,nodo+1)>=0
                    u_tilde(n+1,nodo+1)=0;
                elseif L(n+1,nodo+1)<0
                    u_tilde(n+1,nodo+1)=1;
```



```

end

if nodo==(N-1) %Caso en indicar si todo los bits fueron decodificados
done=1; %Termina la decodificación
else
nodo=floor(nodo/2); %Acude al siguiente nivel
profundidad=profundidad-1; %Disminuye un nivel de profundidad para
continuar con la deco
end

else
%Caso contrario si el vector todavía no llega a la profundidad máxima
npos=(2^profundidad-1)+nodo+1; %Nodo en el cuál se está trabajando
actualmente
%Inicio operaciones en el nodo izquierdo inferior
if ns(npos)==0 %Verifico si en el nodo no se ha
realizado ninguna operación
temp=2^(n-profundidad);
Ln=L(profundidad+1,temp*nodo+1:temp*(nodo+1));
%Extraigo creencias en función de la profundidad y nodo
a=Ln(1:temp/2); %Divido al nodo en dos para ejecutar la función f
b=Ln(temp/2+1:end);
%Trabajo condiciones para las creencias del nodo izquierdo
nodo= nodo*2; %Cambio al nodo de la izquierda
profundidad=profundidad+1; %Bajo de profundidad
temp=temp/2; %Actualizo el índice de selección
L(profundidad+1,temp*nodo+1:temp*(nodo+1))=f(a,b); %Ejecuto
la función minsum y almaceno el resultado
ns(npos)=1; %Cambio la operación del nodo de la derecha
elseif ns(npos)==1 %Verifico si en el nodo se necesita hacer la
operación a la derecha
temp=2^(n-profundidad);
Ln=L(profundidad+1,temp*nodo+1:temp*(nodo+1)); %Creencias
que vienen
a=Ln(1:temp/2); %Divido al nodo en dos para
ejecutar la función f y g y ejecutar operaciones a la izquierda y derecha
b=Ln(temp/2+1:end); %Divido al nodo en dos para
ejecutar la función f de minsum a la izquierda
%Establezco valores para importar los valores del nodo de la izquierda
nodoi=2*nodo;
profundidadi=profundidad+1;
tempi=temp/2;
u_tilde_i=u_tilde(profundidadi+1,tempi*nodoi+1:tempi*(nodoi+1));
%Se accede al registro de las decisiones tomadas en el nodo izquierdo
%Trabajo condiciones para el nodo de la derecha
nodo= nodo*2 + 1; %Cambio al nodo de la derecha
profundidad=profundidad+1; %Bajo la profundidad al nodo
original
temp=temp/2; %Coloco el tamaño del nodo del nodo
L(profundidad+1,temp*nodo+1:temp*(nodo+1))=g(a,b,u_tilde_i);
%Ejecuto la función minsum y almaceno el resultado
ns(npos)=2; %Cambio al nodo central
%Procedo a unir izquierda con derecha y vuelvo al patrón
elseif ns(npos)==2
temp=2^(n-profundidad);
%Establezco valores para importar los valores del nodo de la izquierda y de
derecha
nodoi=2*nodo;
nodod=2*nodo+1;
profundidadc=profundidad+1;
tempc=temp/2;
u_tilde_izq=u_tilde(profundidadc+1,tempc*nodoi+1:tempc*(nodoi+1));
%Se accede al registro de las decisiones tomadas en el nodo izquierdo
u_tilde_der=u_tilde(profundidadc+1,tempc*nodod+1:tempc*(nodod+1));
%Se accede al registro de las decisiones tomadas en el nodo derecho
u_tilde(profundidad+1,temp*nodo+1:temp*(nodo+1))=[mod(u_tilde_izq+u_tilde_der,2
) u_tilde_der]; %Combino nodo izquierdo y derecho

```

```

        %Actualizo valores de nodo y profundidad
        nodo=floor (nodo/2);
        profundidad=profundidad-1;
    end
end
end
msg_tilde=u_tilde(n+1,Q1(N-K+1:end));
end
%Fin Segmento de Código

```

Segmento de Código 2.7. Decodificación SISO con cancelación sucesiva simplificada (SSC) para el esquema de codificación NR-Polar

Implementación del esquema de decodificación SISO por cancelación sucesiva simplificada con listas (SSCL)

La implementación de los decodificadores SISO con cancelación sucesiva simplificada con listas (SSCL) [80], [9] añaden una serie de requisitos extras a ser considerados a la implementación realizada en el Segmento de Código 2.7. El primero es el cálculo del código de redundancia cíclica (CRC) en transmisión y la verificación de la misma en recepción. El proceso de generación del código de redundancia cíclica se logra mediante la función $conv(u,v)$ de Matlab. Recuérdese que el uso del CRC es para la generación de nL listas, algo que obliga a añadir una nueva dimensión a todos los vectores que fueron trabajados en la decodificación SSC. Esto permitirá el almacenamiento y navegación de los datos en la lista que se está trabajando en dicho momento. Finalmente, el uso de listas obliga a crear un vector o vectores que permitan el almacenamiento de los valores de métrica de camino o *Path Metric* que se van generando en cada lista creada [82].

Tomando en cuenta las consideraciones técnicas previas vistas en cuanto al uso de CRC y las mencionadas en esta sección, el Segmento de Código 2.8 presenta la implementación final para un sistema de comunicación con modulación BPSK sobre un canal AWGN con el esquema de codificación NR-Polar y decodificación SISO SSCL.

```

%Inicio codificación
function msg_tilde=sisosscl(n,N,K,Q1,F,rq,nL,maxqr,crc_pol,A)

%Establecimiento de funciones
f=@(a,b) (1-2*(a<0)).*(1-2*(b<0)).*min(abs(a),abs(b));
satx=@(x,th) min(max(x,-th),th);
g=@(a,b,c) satx(b+(1-2*c).*a,maxqr);

L=zeros(nL,n+1,N); %Creencias en función del uso de librerías
ns=zeros(1,2*N-1); %Vector de estado del nodo
u_tilde=zeros(nL,n+1,N); %Vector de decisiones
PML=Inf*ones(nL,1); %Creo un vector para activar las métricas -
Todas desactivadas
PML(1)=0; %Activo la primera métrica

L(:,1,:)=repmat(rq,nL,1,1); %Creencias de la raíz o del nodo 0/profundidad
0

```

```

DML = zeros(nL,N);
PMLL = zeros(nL,N);

%Condiciones iniciales del árbol
nodo=0;
profundidad=0;
done=0;

while done==0
    if profundidad==n %Condición que verifica si todavía hemos o no
    hecho la operacion R-L
        DM=squeeze(L(:,n+1,nodo+1));
        DML(:,nodo+1) = DM;
        PMLL(:,nodo+1) = PML;
        if any(F==(nodo+1)) %Si el nodo está congelado =0
            u_tilde(:,n+1,nodo+1)=0;
            PML=PML+abs(DM) .* (DM<0);
        else
            %Se hace la regla de decisión en función de la decisión de la métrica
            decision=DM<0; %La toma en cuenta la regla de decisión en
            función de la métrica
            PM2=[PML;PML+abs(DM)]; %Se toma en cuenta la métrica contraria y
            se aumenta el valor de PM
            [PML,pos]=mink(PM2,nL); %Se considera los valores máximos y
            mínimos. Los 4 primeros son los valores que pertenecen a la matriz de decisión
            mientras que los últimos son la oposición a ellos
            pos1=pos>nL;
            pos(pos1)=pos(pos1)-nL;
            decision=decision(pos);
            decision(pos1)=1-decision(pos1);
            L=L(pos, :, :);
            u_tilde=u_tilde(pos, :, :);
            u_tilde(:,n+1,nodo+1)=decision;
        end

        if nodo==(N-1)
            done=1;
        else
            nodo=floor(nodo/2);
            profundidad=profundidad-1;
        end
    else
        npos=(2^profundidad-1)+nodo+1;
        %Posición del nodo en el vector del estado del nodo
        %Inicio operaciones en el lado izquierdo del nodo central (Step L)
        if ns(npos)==0
            temp=2^(n-profundidad);
            Ln=squeeze(L(:,profundidad+1,temp*nodo+1:temp*(nodo+1)));
            %Creencias que vienen
            a=Ln(:,1:temp/2);
            %Divido al nodo en dos para ejecutar la función f y g y ejecutar operaciones a
            la izquierda y derecha
            b=Ln(:,temp/2+1:end);
            %Divido al nodo en dos para ejecutar la función f de minsum a la izquierda
            %Trabajo condiciones para las creencias del nodo izquierdo
            nodo= nodo*2;
            %Cambio al nodo de la izquierda
            profundidad=profundidad+1;
            %Bajo de profundidad
            temp=temp/2;
            %Coloco el tamaño del nodo del nodo
            L(:,profundidad+1,temp*nodo+1:temp*(nodo+1))=f(a,b);
            %Ejecuto la función minsum y almaceno el resultado
            ns(npos)=1;
            %Cambio al nodo de la derecha
            elseif ns(npos)==1
                temp=2^(n-profundidad);

```

```

Ln=squeeze(L(:,profundidad+1,temp*nodo+1:temp*(nodo+1)));
%Creencias que vienen
a=Ln(:,1:temp/2);
%Divido al nodo en dos para ejecutar la función f y g y ejecutar operaciones a
la izquierda y derecha
b=Ln(:,temp/2+1:end);
%Divido al nodo en dos para ejecutar la función f de minsum a la izquierda
%Establezco valores para importar los valores del nodo de la izquierda
nodoi=2*nodo;
profundidadi=profundidad+1;
tempi=temp/2;

u_tilde_i=squeeze(u_tilde(:,profundidad+1,tempi*nodoi+1:tempi*(nodoi+1)));
%Se accede al registro de las decisiones tomadas en el nodo izquierdo
%Trabajo condiciones para el nodo de la derecha
nodo= nodo*2 + 1;
%Cambio al nodo de la derecha
profundidad=profundidad+1; %Bajo la profundidad al nodo
original
temp=temp/2;
%Coloco el tamaño del nodo del nodo
L(:,profundidad+1,temp*nodo+1:temp*(nodo+1))=g(a,b,u_tilde_i);
%Ejecuto la función minsum y almaceno el resultado
ns(npos)=2;
%Cambio al nodo de la derecha
%Procedo a unir izquierda con derecha y vuelvo al patrón
elseif ns(npos)==2
temp=2^(n-profundidad);
%Establezco valores para importar los valores del nodo de la izquierda y de
derecha
nodoi=2*nodo;
nodod=2*nodo+1;
profundidadc=profundidad+1;
tempc=temp/2;

u_tilde_izq=squeeze(u_tilde(:,profundidadc+1,tempc*nodoi+1:tempc*(nodoi+1)));
%Se accede al registro de las decisiones tomadas en el nodo izquierdo

u_tilde_der=squeeze(u_tilde(:,profundidadc+1,tempc*nodod+1:tempc*(nodod+1)));
%Se accede al registro de las decisiones tomadas en el nodo derecho

u_tilde(:,profundidad+1,temp*nodo+1:temp*(nodo+1))=[mod(u_tilde_izq+u_tilde_der
,2) u_tilde_der]; %Combino nodo izquierdo y derecho
%Actualizo valores de nodo y profundidad
nodo=floor(nodo/2);
profundidad=profundidad-1;
end
end
end

%Verificación de CRC

msg_tilde=squeeze(u_tilde(:,n+1,Q1(N-K+1:end)));
cout=1;

for i=1:nL
[ql,r1]=gfdeconv(fliplr(msg_tilde(i,:)),crc_pol); %Realizo
la convolución entre los M mensajes generados con el polinomio CRC
if isequal(r1,0) %Realizo
la verificación del CRC
cout = i; %Si la
verificación es correcta, asigno el valor del contador a cout y termino la
operación
break
end
end
end

```

```
msg_tilde=msg_tilde(cout,1:A);  
  
end  
%Fin Segmento de Código
```

Segmento de Código 2. 8. Programación del bloque de decodificación SISO SSCL para el esquema de codificación NR-Polar

Implementación

La implementación del sistema de comunicación en código viene dada por la combinación del sistema de comunicación con modulación BPSK sobre un canal AWGN (Segmento de Código 2.1) como eje fundamental de la codificación. El segmento de código debe acondicionarse para trabajar con los bloques de codificación NR-Polar definido por la función `nrpolar` (Segmento de Código 2.6) y el bloque de decodificación SISO tipo MINSUM por capas, *Rate Matching* y *OFFSET* definido por la función `sisoiterative` (Segmento de Código 2.8). Adicional a eso, se debe considerar añadir la función `reability`, misma que permite el cálculo de las posiciones frías dada la secuencia de confiabilidad Q establecida en el estándar técnico TS 38.212. Esta implementación se tiene en el Anexo B y C. El Anexo B describe la implementación de un sistema de comunicación con codificación NR-Polar con decodificación SISO SSC mientras que el Anexo C contiene el mismo sistema de comunicación, pero con un bloque de decodificación SISO SSCL.

2.3.5. IMPLEMENTACIÓN PARA VISUALIZACIÓN DE RESULTADOS

La implementación final está vinculada con la selección y visualización de los resultados. El orden lógico de implementación viene establecido por la Figura 2.18.

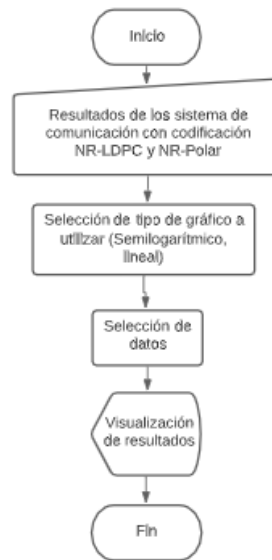


Figura 2.18. Selección y Visualización de Resultados

En este caso la selección del tipo de gráfico está en función de las unidades que utilizan los datos a ser visualizados. En este caso, los gráficos semilogarítmicos, tal como se muestran en la Figura 2.19, se usarán cuando uno de los dos resultados que se visualizarán tengan como unidades los [dB], en este caso exclusivo, cuando se desee conocer un resultado en función del E_b/N_0 usado. Para este caso, MATLAB ofrece el uso del comando *semilogy* para la visualización de resultados donde x es el eje lineal, mientras que y es el logarítmico.

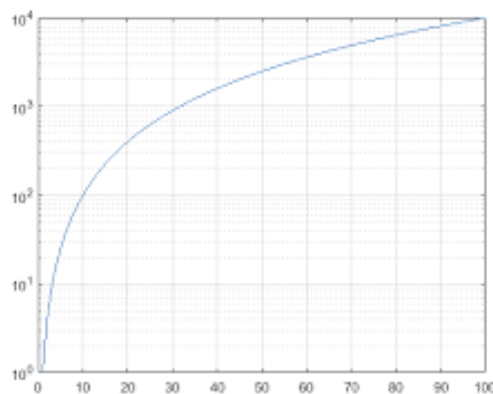


Figura 2.19. Gráfico de referencia que se obtiene con el comando *semilogy*

Mientras tanto, el manejo de gráficas lineales, tal como se muestra en la Figura 2.19, se dará cuando ambas variables sean de esta naturaleza. En este caso, MATLAB utiliza el comando *plot*, mismo que permite la visualización de resultados tanto con el eje x y y como lineales.

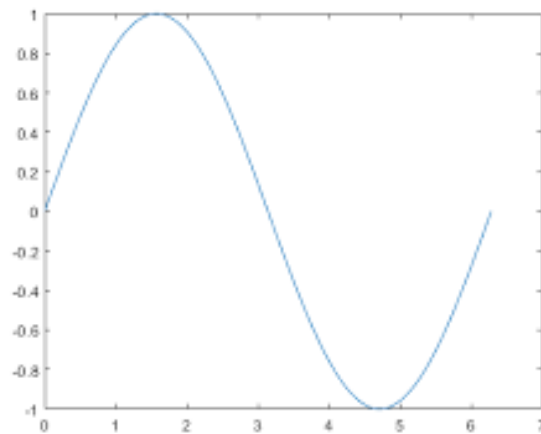


Figura 2.20. Gráfico de referencia que se obtiene con el comando `plot`

Por su lado, en función de cómo se ha dispuesto el almacenamiento de datos en la sección 2.2.7, la selección de estos y su posterior visualización en gráficas bidimensionales será mediante la selección de dos columnas que se deseen relacionar en función de los comandos `semilogy` y `plot`. En este caso, la visualización de resultados puede mostrar por gráfica una como varias curvas, en función de los parámetros que se pida analizar.

3. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados del rendimiento de los sistemas de comunicación en las simulaciones mediante: curvas de BER Vs. E_b/N_0 , FER Vs. E_b/N_0 , BER Vs. Número de Bloques Enviados, tiempo de procesamiento Vs. Número de Bloques Enviados, tiempo promedio de codificación por bloque Vs. número de bloques enviados y tiempo promedio de decodificación por bloque Vs. número de bloques enviados para los sistemas de comunicación implementados con codificación NR-LDPC y NR-Polar. Para un análisis comparativo objetivo, las simulaciones que se han realizado con una misma tasa de codificación R (aprox. $1/2$) y usando distintos valores de E_b/N_0 en [dB], número de iteraciones, listas y número de bloques N a transmitir para obtener las curvas antes mencionadas. Además, de manera extra se incorporarán los resultados de decodificación SISO con cancelación sucesiva simplificada o SSC por sus siglas en inglés con el objetivo de tener un criterio técnico por el cual la codificación Polar sin el uso de listas no es aceptada para usos prácticos. La validación de los resultados obtenidos se dará mediante la comparación con los resultados presentados en el documento técnico “*D3.1 Physical layer of the 5G-RANGE – Part I*” que forma parte del proyecto conjunto entre la Unión Europea y el Ministerio de Telecomunicaciones del Gobierno Federal Brasileño denominado “*5G-RANGE: Remote Area Access Network for the 5th Generation*”

3.1 ESCENARIO DE SIMULACIÓN

Para la obtención de resultados, se simularán los sistemas de comunicación con codificación NR-LDPC y NR-Polar propuestos en los segmentos de código considerando los siguientes valores de E_b/N_0 en dB: 0, 0.5, 1, 1.5, 2, 2.5, 3 y número de bloques¹⁰ transmitidos de: 100, 100, 1000, 5000, 10000, 15000 y 20000, respectivamente, a una tasa de codificación común $R \approx \frac{1}{2}$.

El motivo de escoger el valor de tasa de codificación se debe a que NR-Polar en Uplink tiene una tasa de codificación fija con una palabra código de 1024 bits que está compuesta de un mensaje de 500 bits y un código de redundancia cíclica de 12 bits, por lo tanto, si se desea comparar ambos esquemas en términos de rendimiento debe ser bajo una igualdad de condiciones en términos de R , es decir aproximadamente 0,5 [3]. Por lo tanto, para NR-LDPC, recuérdese que, por lo visto en capítulos previos, el esquema de codificación maneja dos tipos de matrices base conocidas como BG1 o BG2 [2]. Si se aplica la tasa de codificación respectiva con *Rate Matching*, cada grafo/matriz base es capaz de generar

¹⁰ Cada bloque tiene un tamaño de 1024 bits para NR-Polar, 1056 bits para NR-LDPC con una matriz o grafo base BG1 y 1040 bits con una matriz o grafo base BG2.

una palabra código de $44z_c$ o $20z_c$ bits. Se debe tomar en cuenta que el objetivo es obtener una palabra de longitud similar a la que maneja NR-Polar que es 1024 bits, por lo tanto, z_c debe ser de tamaño 24 y 52 para grafos base BG1 y BG2, respectivamente y así obtener una tasa de codificación $R \approx \frac{1}{2}$. Estas matrices base han sido obtenidas en formato .txt [87] [88], mismas que han sido extraídas de [2] y tal como se explicó en la implementación de la codificación, se exportan y se definen como variable predefinida. En el caso de la codificación NR-Polar, se exporta la secuencia de confiabilidad Q de la Tabla 5.3.1.2-1 del estándar técnico TS 38.212 de forma directa [79].

Así mismo, el uso de decodificación SISO MINSUM para NR-LDPC debe considerar el número de iteraciones que se va a utilizar, en este caso, el valor seleccionado será ocho [7] y además, se debe mencionar que el uso de decodificación SISO SSCL para NR-Polar debe establecer de manera predefinida el número de listas que se van a utilizar para el proceso que, en este caso, será cuatro [80]. De forma extra se añade un análisis mediante curvas con diferentes números de iteraciones para SISO MINSUM (2, 4, 6, 8, 10, 12, 16, 20) y de listas para SISO SSCL (2, 4, 8) que intentan explicar y validar el por qué estos textos técnicos recomiendan un cierto número de iteración o lista. Así mismo se añade el análisis con el bloque de decodificación SISO SSC para poder respaldar de forma técnica el por qué este tipo de decodificación no es usado en la práctica debido a su mal rendimiento en términos de BER y FER.

Finalmente, recuérdese que, por lineamientos del método científico, se debe realizar una gran cantidad de pruebas que permitan establecer un patrón de comportamiento de los sistemas implementados para obtener una conclusión válida. En este caso, se considera a una prueba como el envío, a través del canal AWGN, de un solo bloque de información codificado y modulado desde el transmisor hasta el receptor donde se decodificará, demodulará y analizará los parámetros requeridos. Por lo tanto, el sistema al enviar un mayor número de bloques entre transmisor y receptor permitirá obtener mejores resultados tanto en BER, FER y tiempos de procesamiento. Este es uno de los motivos por el cual se decide trabajar con distintos valores del número de bloques a transmitir y los cuales fueron definidos en párrafos anteriores de este literal.

3.2 PRUEBAS DE FUNCIONAMIENTO

La comprobación de que el código implementado es correcto tanto para codificación NR-LDPC como NR-Polar con sus respectivos bloques de decodificación se realiza por medio de la comparación con otros resultados de estudios similares realizados. En este caso, el presente proyecto de titulación comparará los resultados obtenidos con los del documento

“D3.1 Physical layer of the 5G-RANGE – Part I” que forma parte del proyecto conjunto entre la Unión Europea y el Ministerio de Telecomunicaciones del Gobierno Federal Brasileño denominado “5G-RANGE: Remote Area Access Network for the 5th Generation” [89]. El informe realiza una comparación gráfica BER Vs. Es/No entre los esquemas de codificación NR-LDPC y NR-Polar con bloques de decodificación SISO Minsum por Capas y SISO con cancelación sucesiva con listas (SSCL por sus siglas en inglés), respectivamente, a diferentes tasas de codificación por medio del uso de distintas herramientas disponibles en línea [87], [90]. Las condiciones de la simulación realizada en el documento se presentan en la Tabla 3.1.

Tabla 3.1. Condiciones de la simulación propuestas en el documento “D3.1 Physical layer of the 5G-RANGE – Part I” para sistemas de comunicaciones con los esquemas de codificación NR-LDPC y NR-Polar

Canal	AWGN
Modulación	BPSK
Número de Bloques	100000
Longitud de la palabra código	1056 para NR-LDPC/1024 para NR-Polar
Número de Listas / Iteraciones	8/20

Tal como se puede observar, este documento técnico ha sido seleccionado en comparación a otros que realizan estudios similares debido a que las condiciones de la simulación coinciden con los requisitos especificados en la sección 3.1, a excepción del número de iteraciones para el sistema de comunicación que usa el esquema de codificación NR-LDPC. Esto es porque para propósitos de prueba de rendimiento de canal, el uso de 20 iteraciones es una elección común en LDPC y 8 para NR-Polar [89]. La Figura 3.1 del documento mencionado hace referencia al gráfico BER Vs. Es/No, mismo que se presenta a continuación.

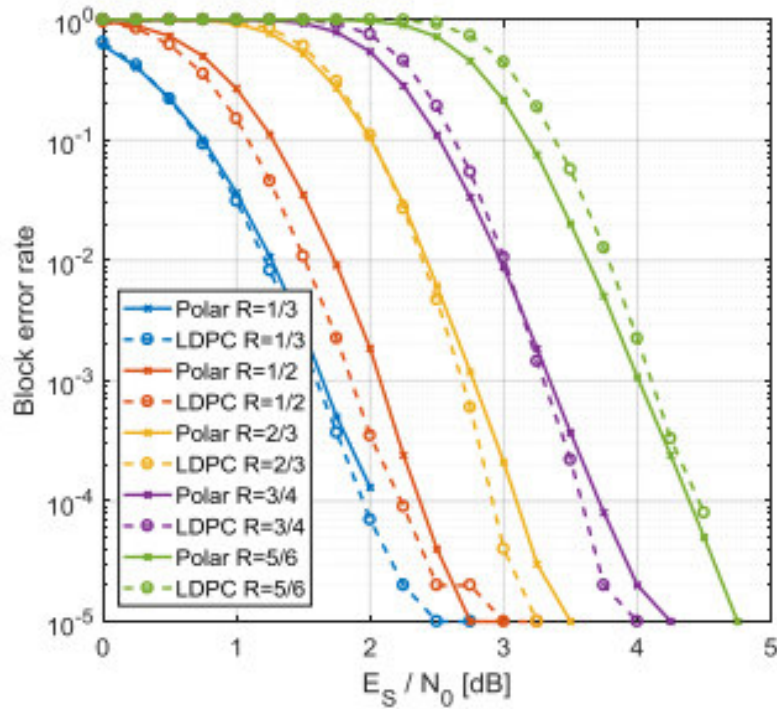


Figura 3.1. Comparación de rendimiento mediante BLER de la codificación NR-LDPC y NR-Polar para 5G en un sistema con modulación BPSK sobre un canal AWGN, datos publicados en el documento “D3.1 Physical layer of the 5G-RANGE – Part I”

Las curvas que interesan en este caso son las correspondientes a NR-LDPC y NR-Polar con una tasa de codificación de 0.5. La Figura 3.2 es el resultado de la simulación con las características presentadas en la Tabla 3.1 con los modelos implementados en el código desarrollado a través de una curva BLER Vs. E_b/N_0 y que se presentan en el Anexo D.

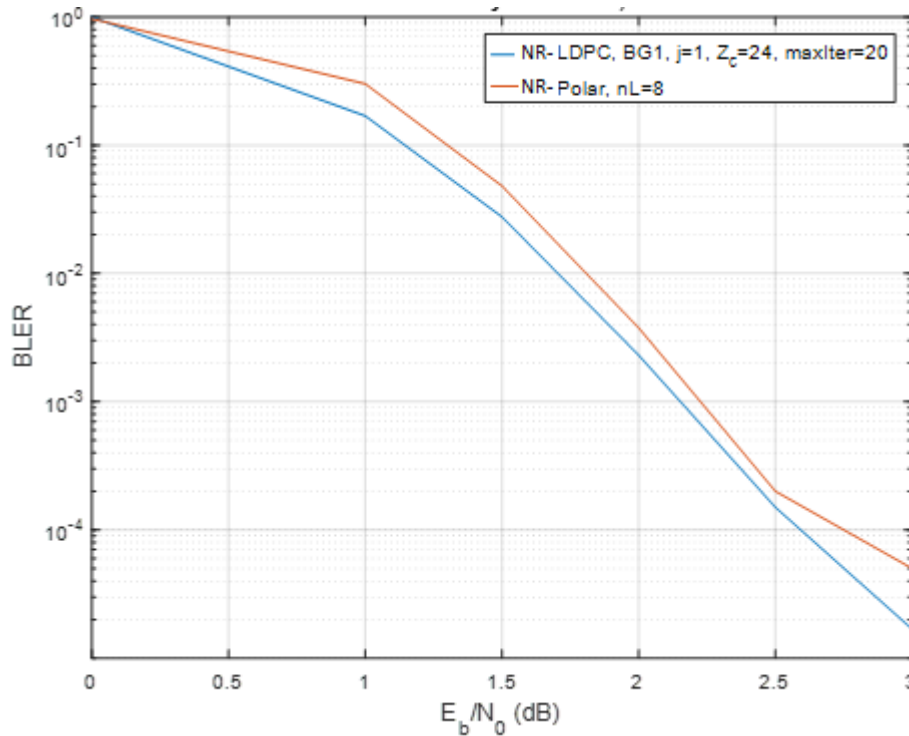


Figura 3.2. Rendimiento de la codificación NR-Polar (*New Radio – Polar Codes*) y NR-LDPC (*New Radio – Low Density Parity Check Codes*) a una tasa de 1/2 y longitud de palabra código 1024/1056, respectivamente

Tal como se puede apreciar entre la Figura 3.1 y 3.2, existe una gran similitud entre ambas curvas, tanto en resultados como en apariencia. Recuérdese que los resultados varían por la naturaleza aleatoria del canal AWGN. Bajo este criterio, se puede comprobar que las implementaciones de los sistemas de comunicación son válidas.

3.3 RENDIMIENTO DE LOS ESQUEMAS DE CODIFICACIÓN

Este segmento del trabajo se dedica de forma especial a mostrar de forma gráfica los resultados de interés obtenidos previamente en las simulaciones y se encuentran en el Anexo E. Para esto, esta parte del documento primero se enfocará en mostrar el rendimiento individual de ambos esquemas para respaldar ciertos criterios técnicos que han sido adoptados para la implementación en sistemas reales, tales como el número de iteraciones para el proceso de decodificación SISO MINSUM con *Offset*, *Layering* y *Rate Matching* para codificación NR-LDPC, o el número de listas para el proceso de decodificación SISO SSCL para codificación NR-Polar.

3.3.1 RENDIMIENTO DEL ESQUEMA DE CODIFICACIÓN NR-LDPC

El sistema de comunicación implementado con el bloque de codificación NR-LDPC emplea un esquema de decodificación iterativo SISO MINSUM con características añadidas, tales como *Rate Matching*, *Offset* y *Layering*. La Figura 3.3 y 3.4 muestran un análisis gráfico del Tiempo de Procesamiento Total Vs. Número de Bloques Transmitidos con distintos números de iteración en decodificación (2, 4, 6, 8, 10, 12, 16, 20), tanto con los grafos base BG1 como BG2 [2].

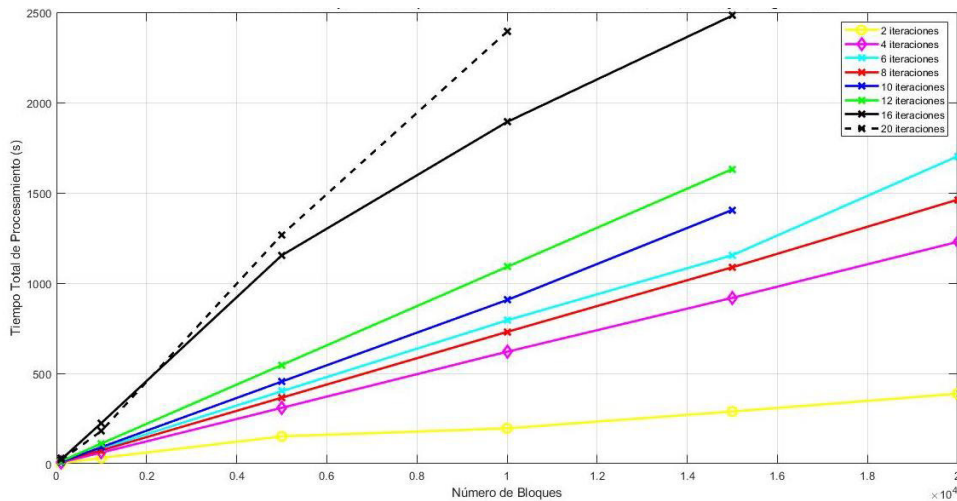


Figura 3. 3. Tiempo Total de Procesamiento Vs. Número de Bloques Transmitidos para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$, con una matriz grafo base BG1, $Z_c=24$, $j_i=1$ y longitud de palabra código aproximada de 1024

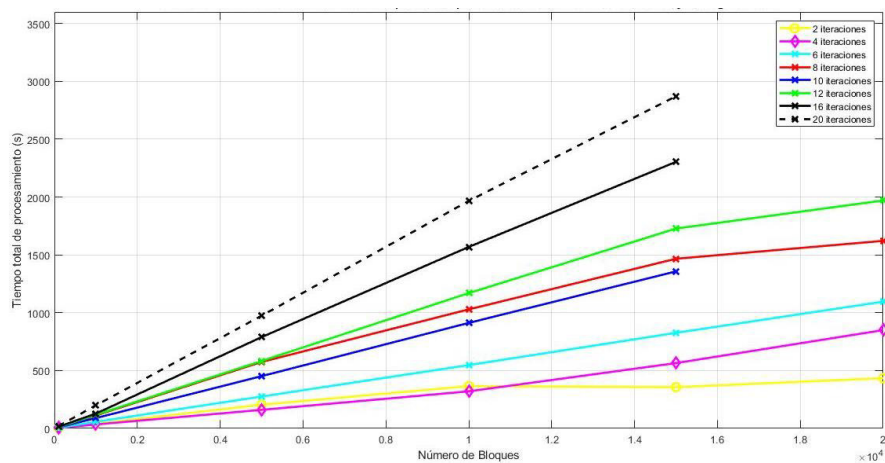


Figura 3. 4. Tiempo Total de Procesamiento Vs. Número de Bloques Transmitidos para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG2, $Z_c=24$, $j_i = 6$ y longitud de palabra código aproximada de 1024

Nótese que, a un mayor número de iteraciones y de bloques transmitidos, el tiempo de procesamiento aumentará con una naturaleza de crecimiento que tiende a ser lineal hasta las 20 iteraciones.

Ahora, considérese la Tabla 3.2 y 3.3, donde solo se toma en cuenta el tiempo de procesamiento promedio en la codificación y decodificación del mensaje a transmitir por un canal AWGN entre Número de Iteraciones de Codificación en segundos y Tiempo Promedio de Procesamiento.

Tabla 3.2. Tiempo Promedio de Codificación en función del Número de Iteraciones para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG1, $Z_c = 24$, $j_i = 1$ y longitud de palabra código aproximada de 1024

Iteraciones	Tiempo Promedio de Procesamiento en Codificación (ms)
2	2.5665
4	2.4322
6	2.9831
8	2.7109
10	2.4926
12	2.2966
16	2.9424
20	2.7083

Tabla 3.3. Tiempo Promedio de codificación en función del Número de Iteraciones para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG2, $Z_c = 52$, $j_i=6$ y longitud de palabra código aproximada de 1024

Iteraciones	Tiempo Promedio de Procesamiento de Codificación (ms)
2	2.7843
4	3.2782
6	2.5670
8	2.6453
10	1.8114
12	1.7978
16	1.7367

20	2.0072
----	--------

Nótese que el esquema de codificación NR-LDPC no emplea mucho tiempo de procesamiento y este se mantiene de casi de forma constante, independiente del número de bloques o iteraciones a usar. Esto porque si se saca un promedio, el tiempo con la matriz base o grafo BG1 es de 2.6 [ms] y con la matriz base o grafo es de 2.3 [ms].

Ahora considérese la Tabla 3.4 y 3.5, donde se muestran los resultados del tiempo de procesamiento en la decodificación del mensaje mediante un gráfico del Número de Iteraciones Vs. Tiempo Promedio de Procesamiento de Decodificación.

Tabla 3.4. Tiempo Promedio de Decodificación en función del Número de Iteraciones para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG1, $Z_c = 24$, $j_i = 1$ y longitud de palabra código aproximada de 1024

Iteraciones	Tiempo Promedio de Procesamiento en decodificación (ms)
2	1.8955
4	3.6004
6	6.1099
8	8.4167
10	8.9848
12	11.3686
16	14.1775
20	19.4932

Tabla 3.5. Tiempo Promedio de Decodificación en función del Número de Iteraciones para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG2, $Z_c = 52$, $j_i=6$ y longitud de palabra código aproximada de 1024

Iteraciones	Tiempo Promedio de Procesamiento de Decodificación (ms)
2	2.7551
4	4.2206
6	5.6565
8	8.5148
10	9.0112
12	11.2085

16	14.3176
20	19.4637

Nótese que el tiempo que se emplea en la decodificación es mucho mayor al que se utiliza en la codificación del mensaje; y contrario a este proceso, el tiempo de procesamiento de decodificación con respecto al número de bloques a transmitir tiende a incrementarse, con una naturaleza de crecimiento que tiende a ser del tipo lineal. Nótese que en ambos casos, ya sea con el manejo de la matriz o grafo base BG1 o BG2 se tiene una diferencia de casi 17[ms] en el manejo de 2 a 20 iteraciones. Pero la pregunta que aquí el lector se está planteando es ¿tiene una ventaja de que existan más iteraciones en decodificación?

Recuérdese que en la sección del marco teórico que hacía referencia a codificación NR-LDPC, se aseguró que el uso de un mayor número de iteraciones aseguraría una menor probabilidad de que el bit o los bits estimados de la decodificación sufran errores, disminuyendo así el BER. Para comprobar esta tesis considérese las figuras a continuación que hacen referencia a los resultados de BER Vs. E_b/N_0 , FER Vs. E_b/N_0 y BER Vs. Número de Bloques Transmitidos con el manejo de las matrices base BG1 y BG2. Así mismo, en la gráfica de E_b/N_0 Vs. BER se incluyen los resultados del BER teórico de un sistema modulado con BPSK, pero sin codificación ($R=1$).

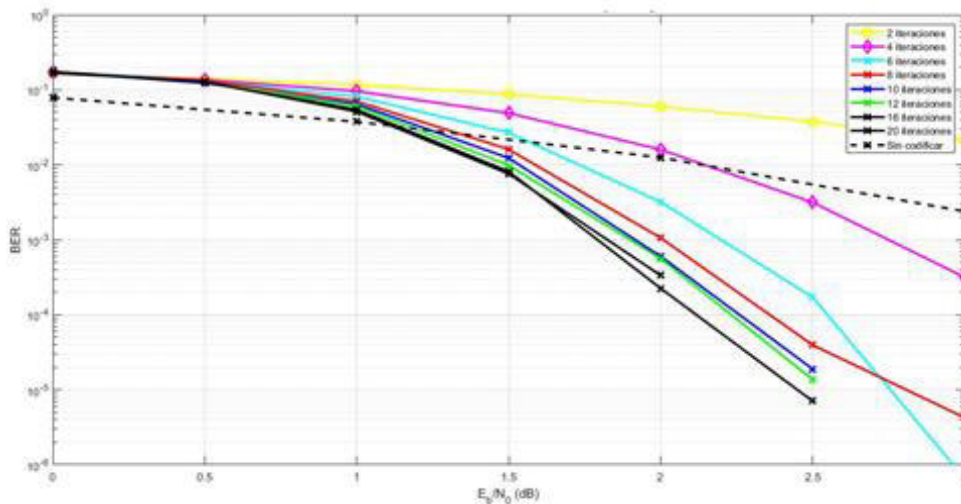


Figura 3.5. BER Vs. E_b/N_0 para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG1, $Z_c = 24$, $j_i=1$ y longitud de palabra código aproximada de 1024

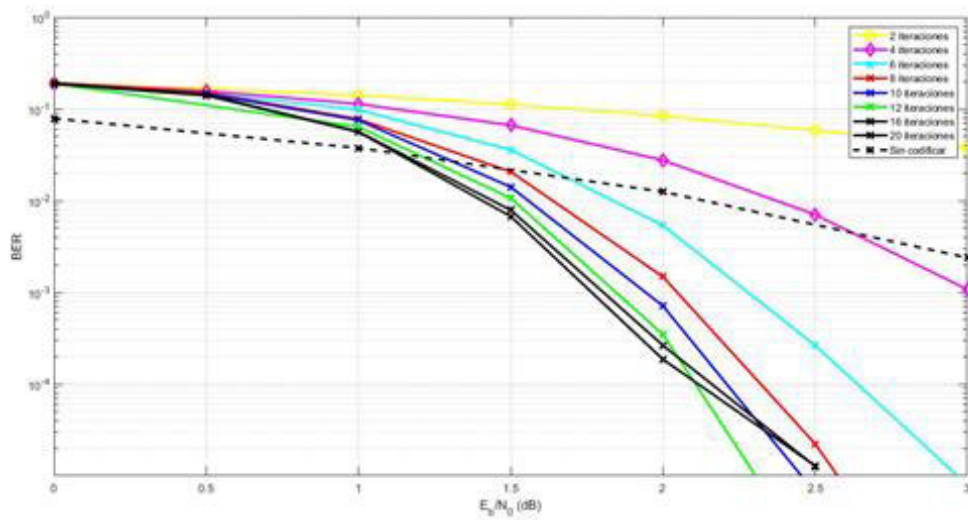


Figura 3.6. BER Vs. E_b/N_0 para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG2, $Z_c = 52$, $j_i = 6$ y longitud de palabra código aproximada de 1024

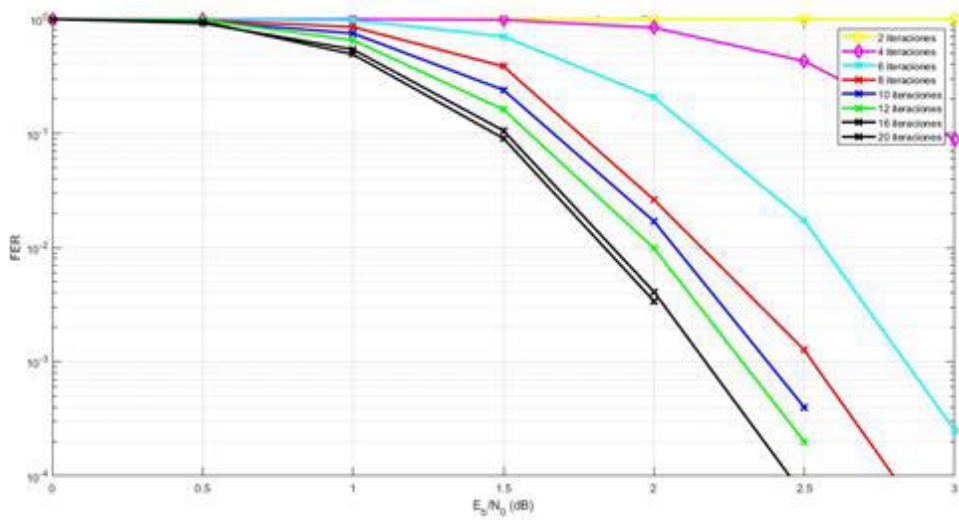


Figura 3.7. FER Vs. E_b/N_0 para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG1, $Z_c = 24$, $j_i=1$ y longitud de palabra código aproximada de 1024

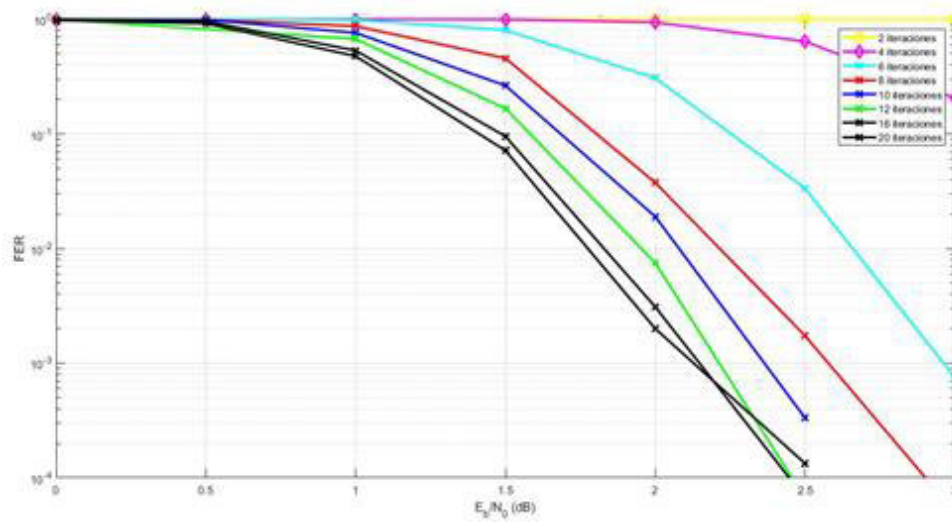


Figura 3.8. FER Vs. E_b/N_0 para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG2, $Z_c = 24$, $j_i=1$ y longitud de palabra código aproximada de 1024

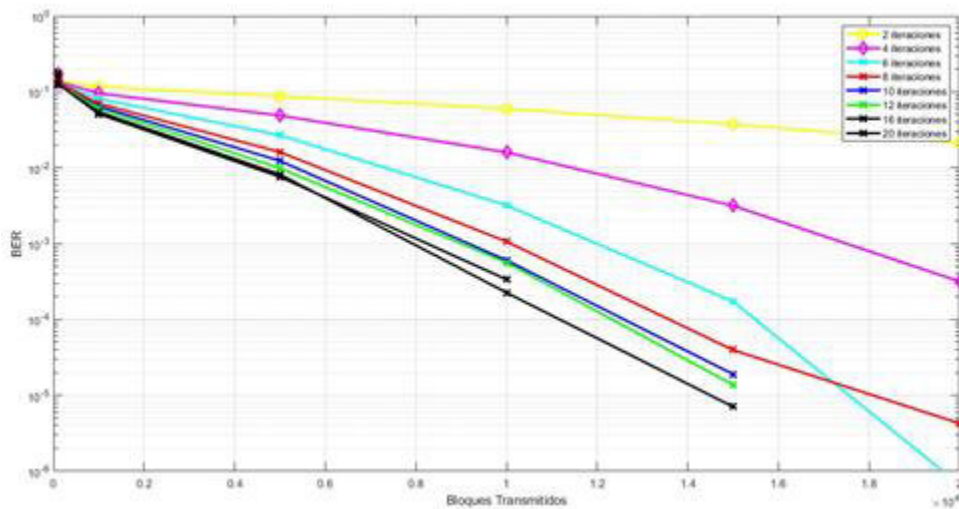


Figura 3.9. BER Vs. Número de Bloques Transmitidos para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG1, $Z_c = 24$, $j_i=1$ y longitud de palabra código aproximada de 1024

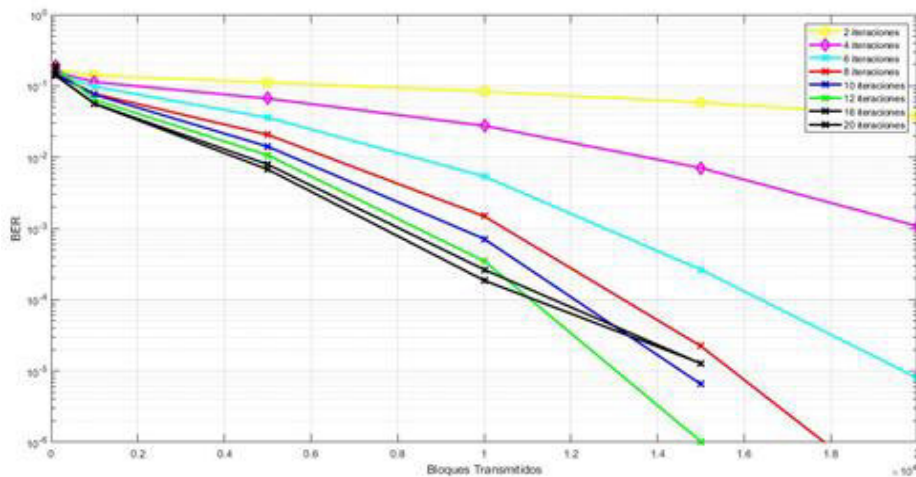


Figura 3.10. BER Vs. Número de Bloques Transmitidos para un sistema de comunicación que trabaja con codificación NR-LDPC y $R \approx \frac{1}{2}$ con una matriz grafo base BG2, $Z_c = 52$, $j_i = 6$ y longitud de palabra código aproximada de 1024

Nótese que la afirmación mencionada en la teoría es correcta, el BER y el FER disminuyen aumentando el número de iteraciones y el número de bloques. Así mismo, debe observarse que el uso de 4 iteraciones o menos no es práctico, ya que teóricamente, el BER o el FER a un mismo nivel de E_b/N_0 suele ser más elevado que en un sistema sin codificación. Bajo este análisis, el criterio técnico más rápido que se viene a la mente sería escoger una decodificación de un número de 20 iteraciones o más. Pero recuérdese por lo visto de las Figuras 3.3 a las 3.8, que a un mayor número de iteraciones y de bloques, el tiempo de procesamiento, sobretodo en la recepción, debido a que el proceso de decodificación es más complejo con respecto a la codificación, aumenta, algo que disminuye el rendimiento de la comunicación en términos de velocidad de transmisión, y consecuentemente, va en contra de uno de los objetivos de la implementación de los sistemas de comunicación con el estándar *New Radio 5G* descritos en la sección 1.3.8 que es aumentar la velocidad de transmisión en el plano de usuario [10]. Bajo esta premisa, este es el motivo por el cual diversos análisis y estudios que concluyen que el rango de iteraciones en decodificación está entre 5 a máximo 8, siendo este último el valor óptimo para casos prácticos [7]. De esta forma se asegura un BER y FER tolerable, pero a la vez se asegura una transmisión rápida de datos entre transmisor y receptor manteniendo la característica de una comunicación rápida y confiable de los sistemas de comunicación que manejan el estándar mencionado. Por este motivo se escoge el valor de 8 en cuanto al número de iteraciones en el proceso de decodificación SISO MINSUM cuando se vayan a comparar los esquemas de codificación implementados.

3.3.2 RENDIMIENTO DEL ESQUEMA DE CODIFICACIÓN NR-POLAR

Ahora, considérese los resultados obtenidos de la implementación del sistema de comunicación con el esquema de codificación NR-Polar y que cuenta con el bloque de decodificación SISO SSC y SSCL con distintos números de listas (2, 4, 8 y 16). Esto tiene como objetivo comparar el rendimiento de ambos esquemas de decodificación y el motivo por el cual el esquema de decodificación SISO SSC no fue considerado para la decodificación del esquema de codificación NR-Polar. Para esto, primero considérese las gráficas de BER Vs. E_b/N_o , FER Vs. E_b/N_o y BER Vs. Número de Bloques Transmitidos. También en el análisis se incluye el BER teórico obtenido para un sistema de comunicación sin esquema de codificación ($R=1$). Recuérdese, además, que, el análisis en NR-Polar se lo realiza solamente en *uplink*; como dato informativo se debe mencionar que si la decodificación se hiciera en *downlink*, el estudio sería similar porque se trabajaría con la misma longitud de la palabra código (1024), pero con la diferencia de que el polinomio generador CRC es de 24 bits. Por este motivo, en la literatura se considera que la tasa de codificación podría mantenerse aproximada a $R = 1/2$, por lo cual el análisis se mantendría igual al que se va a realizar [3], [84], [90].

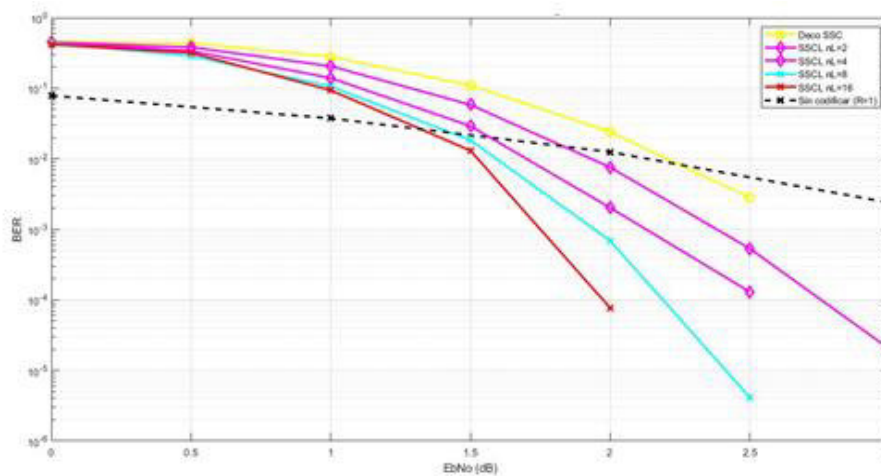


Figura 3.11. BER Vs. E_b/N_o (dB) para un sistema de comunicación que trabaja con codificación NR-Polar con los bloques de decodificación SISO SSC y SSCL a una tasa de codificación $R \approx \frac{1}{2}$ con distintos números de lista y longitud de palabra código aproximada de 1024

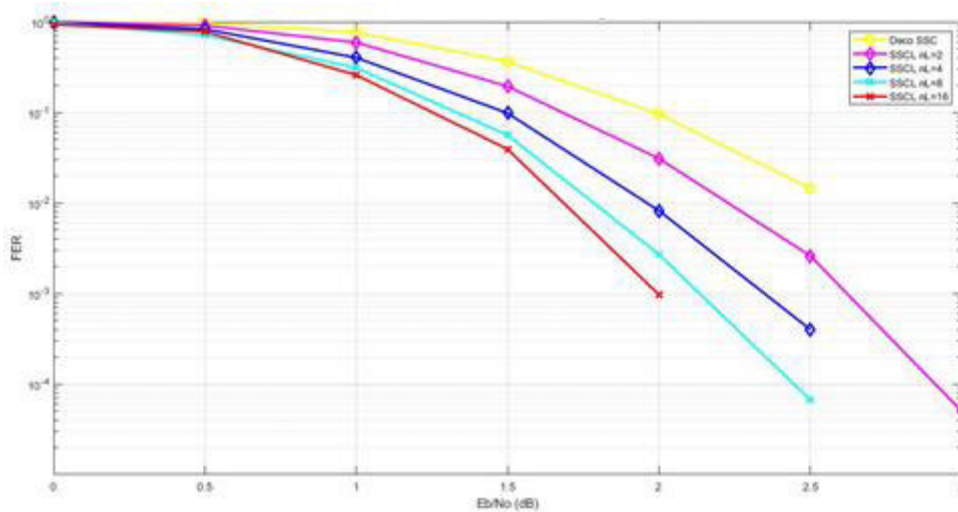


Figura 3.12. FER Vs. E_b/N_0 (dB) para un sistema de comunicación que trabaja con codificación NR-Polar con los bloques de decodificación SISO SSC y SSCL a una tasa de codificación $R \approx \frac{1}{2}$ con distintos números de lista y longitud de palabra código aproximada de 1024

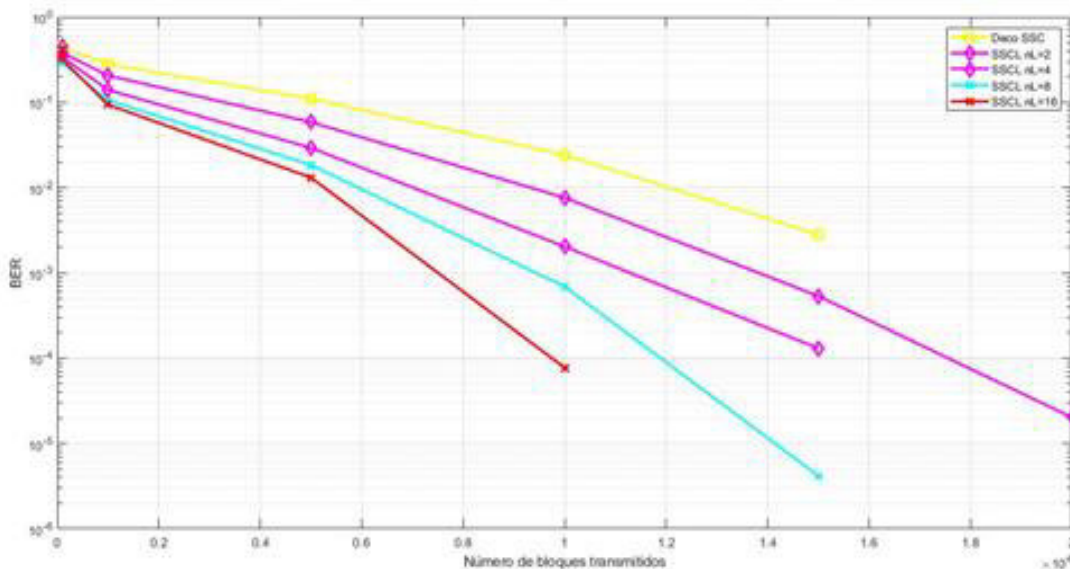


Figura 3.13. BER Vs. Número de bloques transmitidos para un sistema de comunicación que trabaja con codificación NR-Polar con los bloques de decodificación SISO SSC y SSCL a una tasa de codificación $R \approx \frac{1}{2}$ con distintos números de lista y longitud de palabra código aproximada de 1024

Obsérvese que sí existe una notoria diferencia por el uso de los esquemas de decodificación SSC y SSCL en cuanto a BER y FER se refiere. Incluso, en la Figura 3.15 se puede observar que la curva del BER de un sistema de comunicación sin codificar suele

ser mucho mejor solución en la teoría que implementar un sistema de codificación NR-Polar con el esquema SSC, si bien, tal como se mencionó, el sistema tiende a ser bueno cuando se aumenta el número de bits transmitidos a un orden mayor de 2^{20} bits [76]. Estos fueron los motivos por lo cual se llegó a afirmar que este tipo de codificación era de rendimiento “mediocre”, algo que se nota que no es así cuando se lo trabaja con un esquema de decodificación adecuado que es el SISO SSCL. Nótese que, igualmente, que para un mismo nivel de BER, el sistema puede mejorar de dos formas: aumentando el número de listas o el Eb/No. Pero el lector debe saber que en ambos casos se debe tomar en cuenta que ambos mecanismos introducen ciertas dificultades: aumentar el número de listas influye en el tiempo de procesamiento mientras que aumentar el Eb/No es aumentar la potencia de transmisión y con ello que los dispositivos consuman más energía.

Pero, desde el punto de vista de procesamiento, ¿cómo afecta realmente la introducción del uso de listas al rendimiento del sistema?

Las figuras 3.14 y Tabla 3.6 y 3.7 recopilan de forma gráfica los resultados obtenidos en cuanto a tiempo de procesamiento total, codificación por bloque y decodificación por bloque tanto para el sistema de codificación NR-Polar que es decodificado con SISO SSC como SISO SSCL.

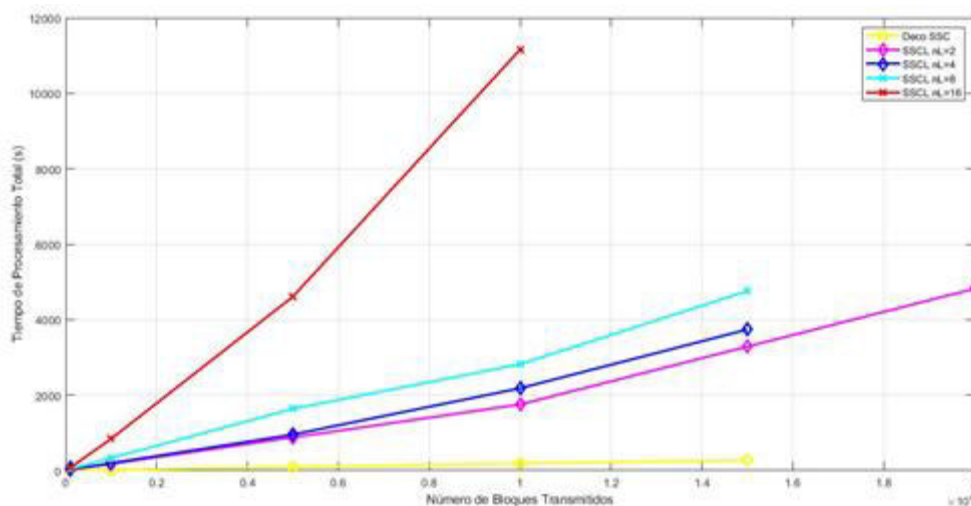


Figura 3.14. Tiempo Total de Procesamiento en [s] Vs. Número de bloques transmitidos para un sistema de comunicación que trabaja con codificación NR-Polar con los bloques de decodificación SISO SSC y SSCL a una tasa de codificación $R \approx \frac{1}{2}$ con distintos números de lista y longitud de palabra código aproximada de 1024

Tabla 3.6. Tiempo Promedio de Procesamiento de Codificación en función del número de listas usadas para un sistema de comunicación por bloque que trabaja con codificación NR-Polar con los bloques de decodificación SISO SSC y SSCL a una tasa de codificación $R \approx \frac{1}{2}$ con distintos números de lista y longitud de palabra código aproximada de 1024 bits

Listas	Tiempo Promedio de Procesamiento en Codificación (ms)
Sin listas (SSC)	2.8405
2	3.1794
4	3.5297
8	2.5842
16	2.4096

Tabla 3.7. Tiempo Promedio de Procesamiento de Decodificación en función del número de listas usadas para un sistema de comunicación por bloque que trabaja con codificación NR-Polar con los bloques de decodificación SISO SSC y SSCL a una tasa de codificación $R \approx \frac{1}{2}$ con distintos números de lista y longitud de palabra código aproximada de 1024 bits

Listas	Tiempo Promedio de Procesamiento en Decodificación (ms)
Sin listas (SSC)	18.5837
2	19.4636
4	23.0777
8	31.9947
16	86.9476

De las figuras expuestas, nótese qué en términos de tiempo de procesamiento, al igual de NR-LDPC, el sistema implementado con NR-Polar tienen comportamientos muy similares tanto en codificación (constante) como decodificación (lineal creciente). Si se observa la Figura 3.19, el tiempo de procesamiento de codificación promedio por bloque casi se mantiene igual con un promedio de 2.9 [ms], si bien se debe crear el código de redundancia cíclica o CRC. Mientras tanto en la recepción, la Figura 3.20 muestra que tal como pasó con el número de iteraciones en NR-LDPC, mientras más alto sea el número de listas, mayor será el tiempo de procesamiento en la decodificación, teniendo una diferencia

68.3639 [s] entre el esquema de decodificación SISO SSC y SISO SSCL con 16 listas, pero el rendimiento del sistema en términos de BER disminuirá.

En cuanto a desventajas, quizás, la única que presenta el esquema de decodificación SISO SSCL frente al SISO SSC es el tiempo de procesamiento mayor que presenta el primero frente a este último. Recuérdese que al igual que en NR-LDPC, el aumento de tiempo de procesamiento se va en contra de los objetivos del estándar *New Radio* 5G que es asegurar una transmisión de alta velocidad y menor latencia posible. Es por este motivo que, por medio de estudios técnicos, el uso de listas debe estar restringido en usos prácticos a 4 debido a las mismas causas expuestas en el literal anterior y mencionadas en este, mantener un BER tolerable y un tiempo de procesamiento que esté de acuerdo con las nuevas reglas que impone el estándar *New Radio* 5G con respecto a velocidad de transmisión entre transmisor y receptor [10], [3].

3.3.3 COMPARACIÓN DEL RENDIMIENTO DE LOS ESQUEMAS NR-LDPC Y NR-POLAR Y ASIGNACIÓN DEL PLANO DE USUARIO Y CONTROL

Ahora considérese el análisis entre los sistemas de codificación NR-LDPC y NR-Polar. En este caso se compararán los resultados de forma gráfica entre el sistema de comunicación con codificación NR-LDPC que contiene el bloque de decodificación SISO MINSUM con características de *layering*, *offset* y *rate matching* con 8 iteraciones y las matrices bases BG1 y BG2 con el sistema de comunicación que contiene el esquema de codificación NR-Polar que contiene el bloque de decodificación SISO SCCL y que trabaja con 4 listas. Este estudio intenta acercar el análisis de ambos esquemas a un nivel más práctico, ya que se usarán los valores de listas e iteraciones de los estándares técnicos vistos y los cuales ya han sido aplicados en sistemas implementados por las empresas desarrolladoras Qualcomm y Huawei a nivel mundial en el estándar *New Radio* 5G y que ya han sido aplicadas en redes en explotación por empresas de telefonía como Vodafone [87]. La Figura 3.21 muestra los resultados obtenidos en términos de BER Vs. E_b/N_o a una misma tasa de codificación ($R \approx \frac{1}{2}$). La presente gráfica adjunta los resultados del BER teórico obtenido en el mismo sistema de comunicación implementado con BPSK sin un sistema de codificación ($R = 1$).

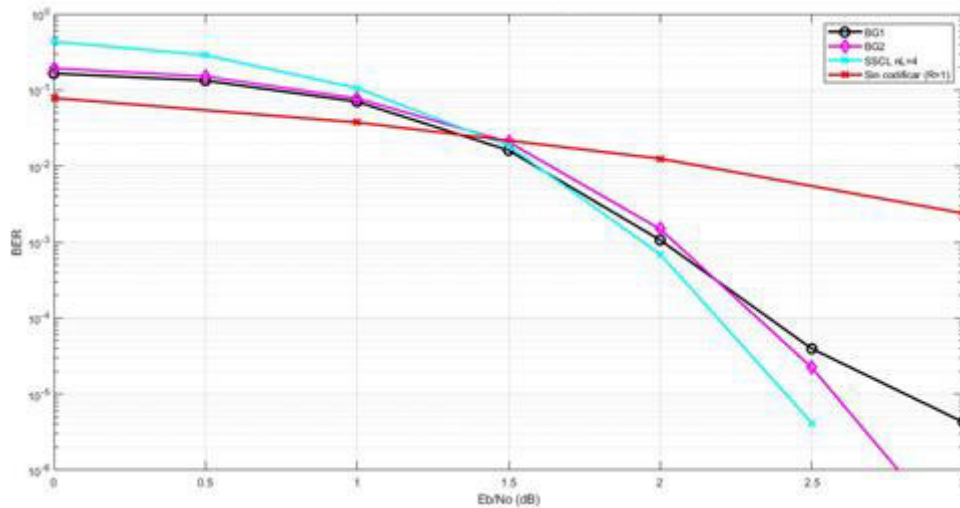


Figura 3.15. Comparación de BER Vs. E_b/N_o para sistemas de comunicación implementados con codificación NR-Polar y NR-LDPC a una tasa de codificación $R \approx \frac{1}{2}$, y longitud de palabra código aproximada de 1024

De la Figura 3.21 se puede concluir que el sistema con codificación NR-Polar presenta un mejor rendimiento en tasa de bits errados en comparación a NR-LDPC para valores más altos de E_b/N_o . Si bien esta diferencia no es tan significativa en el eje logarítmico denota la importancia del manejo de listas en decodificación. Nótese de aquí que el manejo de 8 iteraciones en decodificación NR-LDPC y de 4 listas en el caso de decodificación NR-Polar para casos prácticos no es algo deliberado. Véase que ambos esquemas tienen un rendimiento casi similar en cuanto a BER [10]. Nótese que este comportamiento es muy similar en el caso de las gráficas FER Vs. E_b/N_o y BER Vs. Número de Bloques Transmitidos dadas por la Figura 3.16 y 3.17.

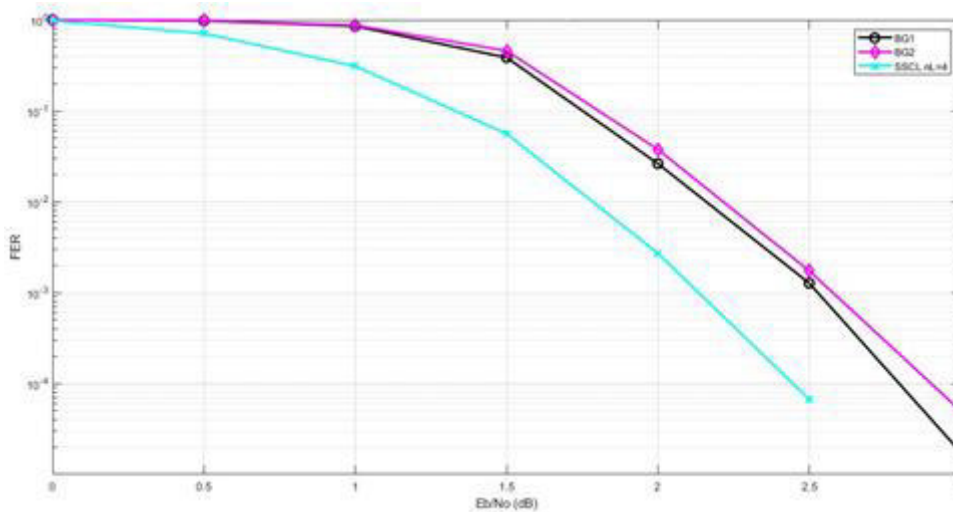


Figura 3.16. Comparación de FER Vs. E_b/N_o para dos sistemas de comunicación implementados con codificación NR-Polar y NR-LDPC, respectivamente, a una tasa de codificación $R \approx \frac{1}{2}$ y longitud de palabra código aproximada de 1024

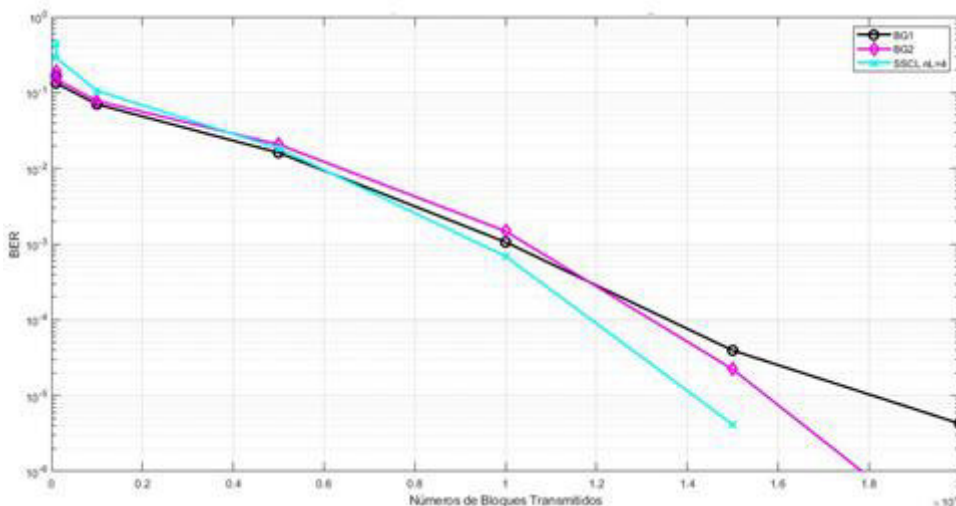


Figura 3.17. Comparación de BER Vs. Número de bloques transmitidos entre un sistema de comunicación que trabaja con codificación NR-Polar y otro con codificación NR-LDPC a una tasa de codificación $R \approx \frac{1}{2}$ y longitud de palabra código aproximada de 1024

Esto también está muy ligado a los tiempos de procesamiento. Si se aprecia la Tabla 3.8 y 3.9, puede notarse que ambos esquemas de codificación implementados tienen tiempos promedios similares en codificación por bloque, algo que no sucede en decodificación, tal como puede verse a continuación.

Tabla 3.8. Tiempo promedio de codificación por bloque de dos sistemas de comunicación implementados con codificación NR-Polar y NR-LDPC, respectivamente, a una tasa de codificación $R \approx \frac{1}{2}$ y longitud de palabra código aproximada de 1024

Esquema	Tiempo Promedio de Procesamiento en Codificación (ms)
NR-LDPC BG1 $Z_c = 24, j = 1$ <i>iteraciones</i> = 8	2.7
NR-LDPC BG2 $Z_c = 52, j = 6$ <i>iteraciones</i> = 8	2.6
NR-Polar Polar $nL = 8$ $CRC = 11$	3.5

Tabla 3.9. Tiempo promedio de decodificación por bloque de dos sistemas de comunicación implementados con codificación NR-Polar y NR-LDPC, respectivamente, a una tasa de codificación $R \approx \frac{1}{2}$ y longitud de palabra código aproximada de 1024

Esquema	Tiempo Promedio de Procesamiento en decodificación (ms)
NR-LDPC BG1 $Z_c = 24, j = 1$ <i>iteraciones</i> = 8	84.2
NR-LDPC BG2 $Z_c = 52, j = 6$ <i>iteraciones</i> = 8	85.1
NR-Polar Polar $nL = 8$ $CRC = 11$	230.8

En la Tabla 3.9 puede observarse que el tiempo promedio de decodificación con el esquema de codificación NR-Polar es mayor por 2.74 veces a la de NR-LDPC que usa la matriz base BG1 y 2.71 veces en comparación a la que utiliza la matriz base BG2. Esto es porque recuérdese que en el caso de NR-Polar, la naturaleza de decodificación secuencial

sumado al manejo de listas que daba al ya conocido esquema de decodificación SISO SCCL introduce mayor latencia frente a la usada en NR-LDPC. En consecuencia, el tiempo de procesamiento total es mucho mayor en el esquema de codificación NR-Polar en comparación al NR-LDPC. Esto es algo que queda muy evidente con la Figura 3.18 en donde se presentan los resultados de Bloques Transmitidos Vs. Tiempo de Procesamiento.

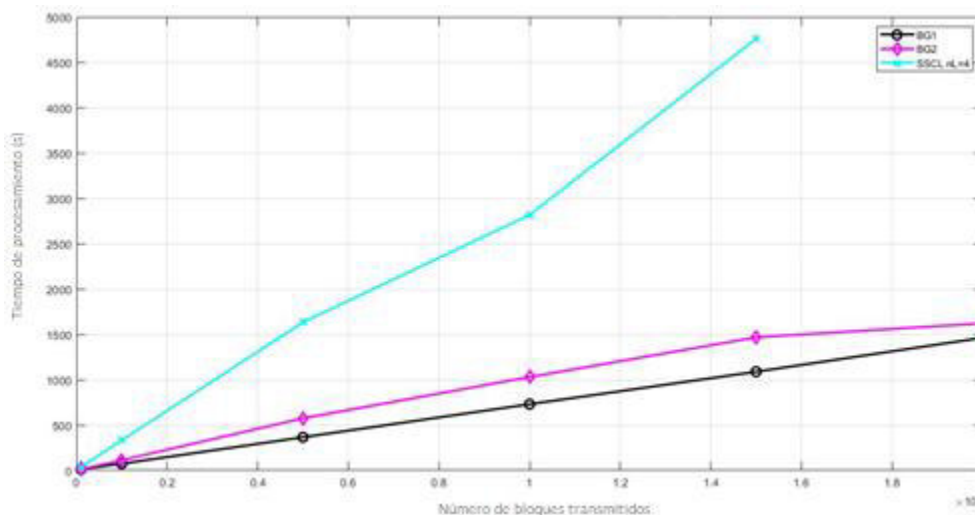


Figura 3. 18. Tiempo promedio de decodificación por bloque Vs. Comparación del Número de iteraciones/listas de dos sistemas de comunicación implementados con codificación NR-Polar y NR-LDPC, respectivamente, a una tasa de codificación $R \approx \frac{1}{2}$ y longitud de palabra código aproximada de 1024.

La Figura 3.18 es muy importante porque permite observar el tiempo de procesamiento en decodificación, lo que motiva a escoger la codificación NR-Polar para la codificación del plano de control en los escenarios de aplicación eMBB y URLLC de *New Radio* 5G y la codificación NR-LDPC para la codificación del plano de datos en el mencionado estándar en el escenario de aplicación eMBB. Quizás en este punto el lector pueda preguntarse por qué no solo escoger la codificación NR-LDPC en ambos planos para los escenarios de aplicación mencionados si se muestra que en sus tiempos de procesamiento es significativamente mejor. La respuesta es porque NR-Polar maneja listas por medio de códigos de redundancia cíclica, algo que introduce ya de forma implícita más confiabilidad e integridad a la transmisión de datos a nivel físico si bien como se ha visto se debe sacrificar rapidez en el tiempo de procesamiento, sobretodo en la decodificación o estimación del mensaje. Este es un motivo que quizás no es deliberado para que el estándar TS 38.212 dicte que NR-Polar debe utilizarse en *downlink* con un polinomio generador de longitud 24 en vez del que se ha trabajado de 11 en *uplink* [10], [1], [91]. Así mismo, tampoco es deliberado que el mismo estándar sugiera que NR-LDPC utilice

mecanismos extras que aseguren la confiabilidad e integridad de las palabras códigos transmitidas y estimadas en decodificación [1]. Estas técnicas pueden consistir en el mismo uso de códigos de redundancia cíclica de distinta longitud y dictadas en el estándar mencionado o el uso de *interleaving*¹¹ otra técnica que también se sugiere para complementar a NR-Polar, pero que consecuentemente introducirá latencia tanto en transmisores como receptores [45], [1], [24]. Lastimosamente, este estudio no se propuso realizar en este trabajo de titulación, por lo que se sugiere desarrollarse en futuras investigaciones, mismas que se detallan en la Sección 4 del presente trabajo de titulación. En general, el análisis de codificación *New Radio* 5G es muy extenso en su literatura, por lo cual, el presente trabajo se centró en rescatar las características principales de los esquemas involucrados en el estándar técnico TS 38.212 para introducirlos de la mejor y más sencilla manera al lector y para investigación de temas más avanzados con respecto a esta temática.

¹¹ El *interleaving* o entrelazado es una herramienta que se utiliza para mejorar los códigos de corrección de errores existentes para que también se puedan utilizar para realizar correcciones de errores en ráfagas. Durante el entrelazado, se dispone los símbolos de mensaje en múltiples bloques de código antes de enviarlos por los canales de red. Debido a esto, las secuencias de ruido de ráfagas largas se distribuyen entre múltiples bloques. Cuando el decodificador reordena los bloques, los errores aparecen como errores aleatorios independientes o errores de ráfaga con longitudes cortas. El decodificador puede corregir los errores utilizando el algoritmo de corrección de errores.

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan las conclusiones, recomendaciones y posibles trabajos futuros relacionados con el presente trabajo de titulación.

4.1 CONCLUSIONES

- Una de las características de las comunicaciones inalámbricas consiste en que los datos recibidos difieren de los transmitidos; esto se da por los fenómenos/anomalías presentes en el canal de comunicaciones, tales como el desvanecimiento, interferencia y el ruido. Para solucionar este inconveniente se han implementado códigos de corrección de errores.
- El principio de la codificación consiste en usar un bloque en el transmisor que sea capaz de formar una palabra código o símbolo C de n bits, que está compuesta del mensaje original de k bits y una redundancia de $(n-k)$ bits que permitirá detectar y corregir errores en el receptor, disminuyendo la tasa de bits errados del sistema de comunicación. A este tipo de codificación se denomina de bloque, en la que la tasa de codificación viene dada por la relación $R=k/n$.
- La decodificación permite la estimación de una palabra código. Tradicionalmente, los decodificadores se pueden clasificar en función de dos técnicas: *Hard* y *Soft*. El presente trabajo de titulación ha hecho uso de decodificación tipo *SISO (Soft Input Soft Output)*. Estos decodificadores aprovechan todos los niveles de voltaje de la señal proveniente del canal (en nuestro caso AWGN) mediante el manejo de índices de probabilidad logarítmicos en base a valores intrínsecos y extrínsecos. Por este motivo, este tipo de decodificadores tienen un bloque de cuantificación de señal previo y que le preceden al proceso de demodulación.
- Debido a su aumento de complejidad matemática, el proceso de decodificación *Soft* tiene mejor rendimiento que el *Hard* en términos de tasa de bits errados. Pese a eso, la gran desventaja como se ha visto en los resultados es que en los esquemas *Soft* la complejidad de implementación aumenta y el procesamiento será mayor a un esquema tipo *Hard*.
- El proceso de cuantificación arreglado por punto (*fixed-point quantization*) tiene como objetivos primero acondicionar la señal que proviene del canal inalámbrico (caracterizado en este caso por el canal AWGN) y proporcionar un valor más exacto de los bits contenidos en el símbolo para una mejor codificación de valores que se

encuentran cercanos a la regla de decisión en el demodulador a más de preparar la señal para la decodificación SISO. Este proceso reduce la carga computacional y ahorra energía, además que permite anteponer el decodificador al demodulador y así realizar el proceso de decodificación tipo *Soft*.

- El Estándar Técnico TS 38.212 define dos esquemas de codificación de canal para el estándar *New Radio* 5G: NR-LDPC y NR-Polar. El *mencionado* estándar define los procedimientos para la codificación del mensaje, pero no para la decodificación que se realizará en la recepción. Por este motivo, el presente proyecto ha adoptado el uso de los esquemas SISO (*Soft-Input Soft-Output*) MINSUM con funciones de Layering, Offset y Rate Matching para decodificación de palabras en NR-LDPC y SISO SSCL (*Simplified Successive Canceled With List*) para NR-Polar.
- El presente trabajo de titulación ha hecho el manejo de matrices base o grafos base BG1 (46x68) y BG2 (42x52) que están en función de índices j_i y factores de expansión Z_c definidos en el estándar técnico TS 38.212. En este caso, la matriz o grafo base BG1, cuyas dimensiones son 46x68, está diseñada para bloques de mensajes entre 300 a 8448 bits con tasas de codificación que están entre $\frac{1}{3}$ y aproximadamente $\frac{8}{9}$. Mientras tanto, la matriz o grafo base BG2, cuyas dimensiones son 42x52, está diseñada en cambio para bloques de mensajes entre 40 a 3840 bits con tasas de codificación que están entre $\frac{2}{3}$ y $\frac{1}{5}$. Recuérdese que la matriz o grafo base BG2 se diferencia de la matriz base BG1 porque el número de bits sistemáticos puede ser variable y puede tener valores tal que $k_b = \{6,8,9,10\}$.
- El proceso de decodificación SISO MINSUM que es utilizado para estimar palabras códigos transmitidas suele acompañarse en NR-LDPC con el proceso de *Layering* y el uso de Offset, a más del ya conocido *Rate Matching*. La técnica de decodificación por capas consiste en dividir la matriz iniciación L en capas y realizar el proceso iterativo de operación de filas y columnas en cada capa. Por otro lado, la técnica de corrección de variables u *Offset* permite que no existan valores muy cercanos a la regla de decisión, que en este caso es cero, mismos que se consideran valores “solitarios” y cuya determinación son difíciles debido a la gran incertidumbre que generan en el sistema probabilístico. Estos procesos permiten mejorar el rendimiento del sistema en términos de tasa de bits errados.

- La codificación polar es de naturaleza secuencial y se basa en el principio de transformación polar de un vector a partir de un núcleo o Kernel. Este núcleo se expande a vectores de orden superior en función de la n -ésima potencia de Kronecker que es un exponente de base 2.
- Los esquemas de codificación polar pueden ser representados de forma gráfica a través de los árboles de diagrama binario. Estos son esenciales para la decodificación porque permiten la técnica de cancelación sucesiva simplificada (SCC), misma que es el principio de implementación que se ha basado el presente trabajo de titulación para MATLAB.
- Al momento de hacer la transformación polar se suele expresar que el canal se polariza. Esto quiere decir que cada bit, por la transformación, puede verse como un canal independiente que contiene la información del resto de bits que se transmiten. Esto es algo que afecta al proceso de codificación y sobretodo decodificación porque en este caso, para cada canal, el error se puede propagar por la mala decodificación de un solo bit, algo que afecta la tasa de error de bits (BER). Esto se puede observar en los esquemas simples de decodificación como SISO SSC que suelen tener un rendimiento muy pobre en términos de BER. Por este motivo, el presente trabajo de titulación en su implementación ha hecho uso de la secuencia de confiabilidad establecida en el estándar técnico TS 38.212.
- La estimación de palabras codificadas con el esquema NR-POLAR utilizarán un decodificador tipo SISO de cancelación sucesiva simplificada con listas (SCCL). Se denomina “cancelación sucesiva” porque la decodificación es igualmente de naturaleza secuencial. Para la decodificación, se utilizarán códigos de redundancia cíclica (CRC) que permitirán generar distintas palabras código candidatas, las cuales posteriormente serán evaluadas en un bloque de verificación CRC. La palabra código candidata que tenga el CRC original será la seleccionada. En este caso de estudio que se centra en el enlace *Uplink*, el estándar especifica que el código debe incorporar en la parte del codificador un CRC de 11 bits y reducir el mensaje a transmitir solamente a 500 bits. El mencionado esquema tiene como desventaja que aumenta el tiempo de procesamiento en la estimación de la palabra código, pero tiene la gran ventaja que aumenta significativamente el rendimiento del sistema al disminuir la tasa de bits errados al usar el esquema de codificación NR-Polar.

- En cuanto a la implementación realizada, se puede decir que se han logrado implementar de forma exitosa en el software MATLAB R2020a dos sistemas de comunicación con modulación BPSK sobre un canal AWGN y un bloque de cuantificación de representación de punto-fijo como base para implementar los esquemas de codificación NR-LDPC y NR-Polar con sus respectivos bloques de decodificación. La implementación se ha hecho mediante lenguaje .m por medio de scripts y cuyo respaldo está en los anexos.
- Del análisis por separado y en conjunto de los esquemas NR-LDPC y NR-Polar con una misma tasa de codificación R (aprox. $1/2$) y usando distintos valores de E_b/N_0 en [dB], número de iteraciones, listas y número de bloques N a transmitir se comprueba que a un mayor número de bloques enviados y E_b/N_0 [dB] se obtendrán valores más bajos de BER y FER, pero se obtendrán tiempos de procesamiento mayores. En base a este parámetro se escoge 8 iteraciones para decodificación de palabras NR-LDPC y de 4 listas para decodificación en NR-Polar. Esto obedece a la búsqueda de un BER y FER tolerables, pero a la vez tiempos de procesamiento aceptables y que estén en alineamiento con los objetivos de rendimiento de los sistemas de comunicación que trabajan con el estándar *New Radio 5G*.
- Del análisis por separado se ha comprobado que el esquema NR-Polar con decodificación SISO SSC no es adecuado debido al bajo rendimiento en términos de la tasa de bits errados, pese a presentar un mejor rendimiento en tiempo de procesamiento. En este caso, se justifica el uso de códigos de redundancia cíclica (CRC) para la generación de listas porque con esto, el sistema adquiere una mejora significativa reduciendo el BER, pero se sacrifica latencia por el tiempo de procesamiento, sobretodo en la generación de M palabras código candidatas de la decodificación.
- Del análisis en conjunto con una misma tasa de codificación R (aprox. $1/2$) y usando distintos valores de E_b/N_0 en [dB], número de iteraciones, listas y número de bloques N a transmitir se ha comprobado que en cuanto a BER y FER, los esquemas de codificación tienen un comportamiento muy similar a mismos valores de E_b/N_0 y Números de Bloques Transmitidos, motivos por los cuales los valores escogidos no son deliberados. En cuanto a tiempo de procesamiento, la codificación por bloque tiene un comportamiento muy similar, pero en decodificación el esquema SISO SCCL presenta mayor latencia. Esto como se explicó en su momento es debido a

que la generación de listas aumenta el tiempo de procesamiento, pero adicionalmente tiene mayor confiabilidad e integridad de la palabra código estimada. De aquí que también se justifica el por qué ambos esquemas han sido seleccionados para el plano de control (NR-Polar) y de usuario (NR-LDPC), respectivamente.

- El reporte técnico TR 38.913 – V15.0.0 define tres escenarios de uso para la nueva versión del interfaz de radio *New Radio* 5G que son: Ancho de banda móvil avanzado (eMBB), Comunicaciones ultra confiables y de baja latencia (URLLC) y Comunicaciones masivas de tipo máquina (mMTC). En la actualidad, las normativas definen a través del mismo reporte que el esquema de codificación NR-Polar se lo usará para el plano de datos por su latencia baja y tiempo de procesamiento. Mientras tanto, el uso de codificación polar se ha definido por el momento para el plano de control de datos para los escenarios eMBB y URLLC, esto debido a la robustez del esquema de codificación, pero a la vez tomando en cuenta que la estimación de la palabra código en recepción suele tener un tiempo de procesamiento mayor a NR-LDPC.

4.2 RECOMENDACIONES

- Se recomienda utilizar una versión de MATLAB igual o superior a la usada R2020a para implementar los sistemas propuestos de tal manera de garantizar que todas las funciones estén disponibles. Se recomienda verificar todos los parámetros de entrada que están descritos en el capítulo 2 del presente documento, ya que todos estos deben estar correctamente establecidos en función de la tasa de codificación que se quiere trabajar. En este caso, poner especial atención a la matriz que se va a utilizar, la tasa de codificación a usar, el valor de E_b/N_0 que está en [dB], el número de iteraciones y el número de bloques a transmitir.
- Como se pudo observar en los resultados, el tiempo de simulación en Matlab puede durar entre pocos segundos, minutos, hasta horas o días (en el caso de la verificación de resultados). Esto está en función del número de iteraciones, número de listas y sobretodo el número de bloques que se vayan a considerar. Se recomienda usar un computador con buenos recursos de procesamiento y que mientras se realice la simulación no se haga ninguna otra tarea en el computador.
- Tal como se puede observar en el Anexo E que es la muestra de los resultados obtenidos en el presente trabajo, para obtener mejores resultados con altos valores de

Eb/No, iteraciones y número de listas se recomienda que se hagan las simulaciones con valores más grandes de números de bloques a transmitir.

4.3 TRABAJOS FUTUROS

A continuación, se enumeran los trabajos que se sugieren realizar en esta área:

- Análisis y cálculo de los grafos o matrices base BG1 y BG2 para NR-LDPC. Este proyecto tiene como objetivo el análisis de los fundamentos teóricos matemáticos para la realización de una matriz base BG1 o BG2, su cálculo y su posterior aplicación en la codificación de mensajes con el uso del esquema NR-LDPC.
- Análisis del parámetro de Bhattacharya. Este proyecto tiene como objetivo investigar el fundamento de la polarización de canal, sus características, ventajas y desventajas para su posterior aplicación en la codificación de mensajes con el uso del esquema NR-Polar.
- Comparación entre los esquemas de codificación NR-LDPC y NR-Polar con respecto a códigos de 3G y 4G (codificación tipo convolucional y los turbo-códigos). Este proyecto tiene como objetivo el análisis y comparación en términos de rendimiento en tasa de bits errados (BER), bloques errados (FER) y tiempo de procesamiento de los esquemas de codificación mencionados.
- Implementación de NR-LDPC y NR-Polar en FPGA (*Field Programmable Gate Array*). Este proyecto tiene como objetivo el acercar los esquemas de codificación propuestos en el presente documento hacia compuertas programables y la comparación con los resultados obtenidos en el presente proyecto, además que son un paso previo a la implementación de un canal físico real para el estándar *New Radio* 5G. En este caso, se le recomienda al lector revisar la documentación en línea propuesta por Intel, que ha realizado una línea exclusiva de este tipo de estudios, sobretodo en NR-Polar.
- Simulación en MATLAB de un sistema de comunicación inalámbrico con codificación NR-LDPC y NR-Polar con los diferentes tipos de modulación que debe brindar el canal físico de *New Radio* 5G, que están definidos en el estándar técnico TS 38.211. Este proyecto tiene como objetivo continuar con el análisis de las características del canal físico *New Radio* 5G, esta vez sumando, a más de los esquemas de codificación, los esquemas de modulación definidos para este estándar.
- Análisis e implementación del canal físico de control y de usuario para el estándar *New Radio* 5G según la normativa técnica descrita en el estándar técnico TS 38.213 y

38.214. Esto tiene como objetivo la continuación del análisis teórico y práctico del canal físico para el estándar *New Radio* 5G y se basa en el análisis e implementación de un canal físico de control y de usuario en función del estándar mencionado. Debido a que la documentación es extensa en este caso, se recomienda analizar la misma y dividir el tema en varios subtemas.

- En general, el documento presentado hace una introducción muy importante a lo que es la codificación NR-LDPC y NR-Polar en función del estándar técnico TS 38.212 de 5G *New Radio*, pero la misma no se puede considerar suficiente debido a la gran cantidad de temas que abarcan los estándares técnicos de la 3GPP sobre este tema. En este caso, se sugiere analizar los temas que se proponen en el estándar técnico que hablan al detalle sobre estos dos esquemas de codificación y sus características.

REFERENCIAS BIBLIOGRÁFICAS

- [1] 3GPP, TS 38.212: 5G New Radio Multiplexing and Channel Coding, 2018.
- [2] 3GPP, "LDPC Base Graph 1 & 2," de *TS 38.212: 5G NR Multiplexing and channel coding*, 25, 2018, p. 21.
- [3] 3GPP, "Polar Encoding," de *5G NR: Multiplexing and channel coding*, 2018, pp. 14-15.
- [4] M. Bolaños y D. Díaz, "Decodificación Hard-Decision," de *Análisis del Desempeño de la Codificación Reed-Muller en un Canal con Ruido Blanco Auditivo Gaussiano*, Popayán, 2015, p. 22.
- [5] M. Bolaños y D. Díaz, "Decodificación Soft-Decision," de *Análisis del Desempeño de la Codificación Reed-Muller en un Canal con Ruido Blanco Auditivo Gaussiano*, Popayán, 2015, pp. 25-26.
- [6] Zarubica, Radivoje & Hinton, Ryan & Wilson, S.G. & Hall, Eric., "Efficient quantization schemes for LDPC decoders," pp. 1-5, 2008.
- [7] Y. O. P. Liu y D. G. M. Mitchell, "On Generalized LDPC Codes for 5G Ultra Reliable Communication," *Y. Liu, P. M. Olmos and D. G. M. Mitchell, "On Generalized LDPC Codes for 5G Ultra Re2018 IEEE Information Theory Workshop (ITW)*, pp. 3-4, 2018.
- [8] 3GPP, "Rate matching for LDPC codes," de *5G New Radio Multiplexing and channel coding*, 2018, pp. 25-26.
- [9] A. Thangaraj, "Successive Cancellation List Decoding," 2019.
- [10] 3GPP, "TR 38.912: Study on New Radio (NR) Access Technology," 2018, p. 34.
- [11] C. Perez, "Canal de Radio," de *Sistemas de Telecomunicación*, Santander, 2007, p. 6.
- [12] D. Sánchez, "Canal AWGN," de *Análisis, Diseño e Implementación de un Modelado de Canal Basado en Cadenas de Markov y Series Temporales Para un Sistema Adaptativo*, Atizapán de Zaragoza, 2007, pp. 23-25.
- [13] C. Perez, "Canal de Radiofrecuencia," de *Sistemas de Telecomunicación*, Santander, 2007, pp. 6-7.
- [14] G. Andrade, "Alfabeto de modulación," de *Implementación de la Codificación y Decodificación del Código LDPC (Low Density Parity Check) en Matlab*, Quito, Pichincha, 2017, pp. 9-10.
- [15] *Lecture 07: Bit Error Rate (BER) of AWGN Channels*. [Película]. India: IIT Kanpur, 2015.

- [16] F. Del Pino, "Detección Por Envolvente," de *Estudio y Diseño de Moduladores y Demoduladores Integrados para Comunicaciones Inalámbricas en la Tecnología SiGe 0.35um de AMS*, Canaria, 2014, pp. 30-36.
- [17] G. Breed, Bit Error Rate, Oregon: Summit TechnicalMedia, LLC, 2003.
- [18] V. Meghdadi, BER Calculation, Islamabad, 2008.
- [19] C. Perez, "SNR," de *Sistemas de Telecomunicación*, Santander, 2007, p. 58.
- [20] D. A. Muqaibel, "Probability & Random Processes For Engineers," de *Random Processes Spectral Characteristics*, 2019, pp. 12-16.
- [21] A. Thangaraj, "Bit Error Rate (BER) and Signal to Noise Ratio (SNR)," de *NPTEL-NOC IITM*, 2019.
- [22] S. Malluri y V. K. Pamula, "Gaussian Q-function and Its Approximations," *International Conference on Communication Systems and Network Technologies*, pp. 74-77, 2013.
- [23] C. Cruz, "Cruz Fierro," 6 9 2020. [En línea]. Available: <http://tecno.cruzfierro.com/formularios/erf.pdf>. [Último acceso: 2 10 2020].
- [24] J. Bae, A. Abotabl, K.-B. S. Hsien-Ping L y J. Lee, "An overview of channel coding for 5G NR cellular communications," *Industrial Technology Advances*, 2019.
- [25] C. Perez, Codificación de Canal, 2015.
- [26] S. Shao, "Survey of Turbo, LDPC, and Polar Decoder ASIC Implementations," *IEEE Communications Surveys & Tutorials*, vol. 21, nº 3, pp. 2-5, 2019.
- [27] Andrade Salinas, G. G. (2017). Implementación de la codificación y decodificación del código LDPC (Low Density Parity Check) en MATLAB. 122 hojas. p. 5. Quito : EPN.
- [28] M. Bolaños y D. Díaz , "Apéndice B. Probabilidad de Error con Decodificación Soft Decision," de *Análisis del Desempeño de la Codificación Reed-Muller en un Canal con Ruido Blanco Auditivo Gaussiano*, Cauca, 2015, pp. 77-78.
- [29] C. Perez, "Relación Eb/No," de *Sistemas de Telecomunicación*, Santander, 2007, p. 58.
- [30] M. Madd, "BER Vs Eb/N0 BPSK Modulation over Different Types of Channel," *Australian Journal of Basic and Applied Sciences*, vol. 12, nº 5, pp. 31-38, 2018.
- [31] D. Chavez, Implementación de los esquemas de modulación BPSK y QPSK en Matlab/Simulink: estudio comparativo, Guayaquil, 2017.
- [32] Andrade Salinas, G. G. (2017). Implementación de la codificación y decodificación del código LDPC (Low Density Parity Check) en MATLAB. 122 hojas. p. 5. Quito : EPN. p. 14.

- [33] M. Tomlinson, C. Tjhai, M. Ambroze, M. Ahmed y M. Jibril, *Soft and Hard Decision Decoding Performance*, 2017.
- [34] M. García, L. Martínez y T. Ramírez, "Distancia de Hamming," de *Introducción a Teoría de Códigos*, Lejona, 2017.
- [35] D. Shafiullah, M. Islam, M. Faisal y I. Rahman, "Optimized Min-Sum decoding algorithm for Low Density PC codes," *Advanced Communication Technology (ICACT)*, pp. 475-480, 2012.
- [36] A. Thangaraj, "Implementation of $n = 3$ Repetition Code in MATLAB," de *NPTEL-NOC IITM*, 2019.
- [37] Rizk, M., Baghdadi, A., Jézéquel, M. *et al*, "Efficient quantization and fixed-point representation for MIMO turbo-detection and turbo-demapping," *EURASIP Journal on Embedded Systems*, 2017.
- [38] H. Carrasco, "Codificación Lineal de Bloques," de *Codificación de Canal*, Valparaíso, 2016, p. 1.
- [39] H. Carrasco, "Matriz Generadora," de *Codificación de Canal*, Valparaíso, 2016, pp. 3-8.
- [40] H. Carrasco, "Matriz de Verificación de Paridad," de *Codificación de Canal*, Valparaíso, 2016, p. 8.
- [41] Y. Hawwar, E. Farag, S. Vanakayala, R. Pauls, X. Yang, S. Subramanian, P. Sadhanala, L. Yang, B. Wang, Z. Li, H. Chen, Z. Lu, D. Clark, T. Fosket, P. Mallela, M. Shelton, D. Laurens, T. Salaun, L. Gougeon, N. Aubourg, H. Morvan y N. L., "3G UMTS Wireless System Physical Layer: Baseband Processing Hardware Implementation Perspective," *IEEE Communications Magazine*, vol. 44, nº 9, pp. 52-58, 2006.
- [42] Z. Shen, A. Papasakellariou, J. Montojo, D. Gerstenberger y F. Xu, "Overview of 3GPP LTE-Advanced Carrier Aggregation for 4G Wireless Communications," *IEEE Communications Magazine*, vol. 50, nº 2, pp. 122-130, 2012.
- [43] C. Berrou, A. Glavieux y P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes. 1," *IEEE International Conference*, vol. 2, pp. 1064-1070, 1993.
- [44] S. Tabbane, "5G Networks and 3GPP Release 15," de *ITU PITA Workshop on Mobile Network Network Planning & Security*, 2019.
- [45] 3GPP, "TR 38.913: Study on Scenarios and Requirements for Next Generation Access Technologies," 2018, p. 9.
- [46] K. Mallinson, "The Path to 5G: as uch evolution as revolution," 2016.
- [47] F. Luo y C. Zhang, "5G Standart Development: Technology & Roadmap," *Wiley IEEE Press*, pp. 616-622, 2016.

- [48] 3GPP, RP-161214 Study on New Radio Access Technology, 2016.
- [49] Samsung; Qualcomm; Incorporated; Nokia, ASB; KT Corporation; Intel Corporation, "3GPP RAN WG1 Meeting 86bis, R1-1610690, Way Forward on Observations for eMBB Data Channel Coding," Liboa, 2016.
- [50] ZTE Microelectronics, "3GPP TSG RAN WG1 87, R1-1611109, Evaluation on Channel coding," Reno, 2016.
- [51] ZTE Microelectronics, "3GPP TSG RAN WG1 87, R1-1613078, Evaluation on Channel Coding," Reno, 2016.
- [52] B. Zeynep, K. Egilmez, L. Xiang, R. Moulder y L. Hanzo, "The Development, Operation and Performance of 5G Polar Codes," *IEEE Communications Surveys & Tutorials*, vol. 22, n° 1, pp. 96-122, 2020.
- [53] R. Gallager, Low Density-Parity Check Codes, 1963.
- [54] D. MacKay y R. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, pp. 1645-1646, 1996.
- [55] O. Shao, "Survey of Turbo, LDPC, and Polar Decoder ASIC Implementations," *IEEE Communications Surveys & Tutorials*, vol. 21, n° 3, pp. 7-11, 2019.
- [56] Andrade Salinas, G. G. (2017). Implementación de la codificación y decodificación del código LDPC (Low Density Parity Check) en MATLAB. 122 hojas. Quito : EPN, p. 17.
- [57] 3GPP, "Low density parity check coding," de *TS 138.212: 5G New Radio Multiplexing and channel coding*, 2018, pp. 18-19.
- [58] Andrade Salinas, G. G. (2017). Implementación de la codificación y decodificación del código LDPC (Low Density Parity Check) en MATLAB. 122 hojas. Quito : EPN. p. 21.
- [59] Y. Fang, G. Bi, Y. Guan y M. Lau, "A Survey on Protograph LDPC Codes and Their Applications," *IEEE Communications Surveys Tutorials*, vol. 17, n° 4, pp. 1989-2016, 2015.
- [60] J. Thorpe, "Low-Density Parity-Check (LDPC) Codes Constructed from Protographs," *The Interplanetary Network Progress Report* , vol. 42, n° 154, p. 1–7, 2003.
- [61] Industrial Technology Advances, "Protograph codes," de *An overview of channel coding for 5G NR cellular communications*, 2019, p. 4.
- [62] A. Thangaraj, "LDPC Codes in 5G: protograph, base matrix, expansion," de *LDPC and Polar Codes in 5G Standard*, 2019.
- [63] F. Hamidi-Sepehr, A. Nimbalkar y G. Ermolaev, "Analysis of 5G LDPC Codes Rate-matching Design," *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pp. 1-5, 2018.

- [64] M. Lahmer, M. Belkasmi y A. Rhattoy, "Convergence Analysis Of Iterative Threshold Decoding Process," *Journal of Computer and Communications*, vol. 4, nº 14, pp. 98-107, 2016.
- [65] Andrade Salinas, G. G. (2017). Implementación de la codificación y decodificación del código LDPC (Low Density Parity Check) en MATLAB. 122 hojas. Quito : EPN, pp. 24-26.
- [66] P. Torres y S. Malhão, "Practical implementation of repetition codes," de *International Journal Of Mechatronics And Applied Mechanics*, 2018.
- [67] A. Gulliver, "Single parity check product codes," *IEEE Transactions on Communications*, vol. 49, nº 8, pp. 1354-1362, 2001.
- [68] A. Maher, C. D'Amours y A. Yongacoglu, "Simplified LLRs for the Decoding of Single Parity Check Turbo Product Codes Transmitted Using 16QAM," *Research Letters in Communications*, vol. 2007, 2007.
- [69] A. Thangaraj, "Soft Input and Soft Output (SISO) Decoder for the Single Parity Check(SPC) Code," de *LDPC and Polar Codes in 5G Standard*, 2019.
- [70] Andrade Salinas, G. G. (2017). Implementación de la codificación y decodificación del código LDPC (Low Density Parity Check) en MATLAB. 122 hojas. Quito : EPN, pp. 27-29.
- [71] E. Ram y Y. Cassuto, "LDPC Codes with Local and Global Decoding," 2018.
- [72] C. Marchand, L. Conde-Canencia y E. Boutillon, "Architecture and finite precision for Layered LDPC Decoders," *Signal Processing Systems (SIPS)*, pp. 350-355, 2010.
- [73] C. Chen, Y. Xu, H. Ju, D. He, Z. W y Y. Zhang, "Variable Correction for Min-Sum LDPC Decoding Applied in ATSC3.0," *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1-5, 2018.
- [74] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Transactions on Information Theory*, vol. 55, nº 7, pp. 3051-3073, 2009.
- [75] S. Mude y B. Naik, "Polar Code and Polarization using Bhattacharya Parameter," *IEEE 9th Annual Information Technology, Electronics and Mobile*, pp. 151-154, 2018.
- [76] W. Gross, "Are Polar Codes Practical?," de *Coding: From Practice to Theory*, Montreal, 2015.
- [77] R. Ruíz, "Árboles Binarios," de *Árboles Binarios*, Quito, 2012.
- [78] V. Bioglio, C. Condo y I. Land, "Design of Polar Codes in 5G New Radio," *IEEE Communications Surveys & Tutorials*, 2018.
- [79] 3GPP, "Polar sequence and its corresponding reliability," de *5G NR: Multiplexing and channel coding*, 2018, pp. 17-18.

- [80] F. Ercan, C. Condo, A. Hashemi y W. J. Gross, "On Error-Correction Performance and Implementation of Polar Code List Decoders for 5G," 2017.
- [81] A. Thangaraj, "Successive Cancellation(SC) Decoder for Polar Codes: Illustration of its Building Blocks with $N=2,4$," 2019.
- [82] L. Xiang, Y. Liu, Z. Kaykac Egilmez, R. Maunder, L.-L. Yang y L. Hanzo, "Soft List Decoding of Polar Codes," *IEEE Transactions on Vehicular Technology*, pp. 99-105, 2020.
- [83] A. Thangaraj, "Successive Cancellation(SC) Decoder for a General (N,K) Polar Code," 2019.
- [84] 3GPP, "CRC calculation," de *TS 38.212: 5G NR Multiplexing and channel coding* , 2018, p. 9.
- [85] Rizk, Mostafa & Baghdadi, Amer & Jézéquel, Michel & Mohanna, Yasser & Atat, Youssef, "Efficient quantization and fixed-point representation for MIMO turbo-detection and turbo-demapping," *EURASIP Journal on Embedded Systems*, 2017.
- [86] MATLAB, "tic: Start stopwatch timer," 2020. [En línea]. Available: <https://la.mathworks.com/help/matlab/ref/tic.html>.
- [87] P. Neuhaus, "5g-nr-ldpc: GitHub," 8 11 2018. [En línea]. Available: <https://github.com/vodafone-chair/5g-nr-ldpc>. [Último acceso: 2020 11 11].
- [88] M. Sreedharan, "NR-LDPC-BG: GitHub," 23 07 2020. [En línea]. Available: <https://github.com/manuts/NR-LDPC-BG>. [Último acceso: 2020 11 11].
- [89] B. Sokal, A. Matos, C. Silva, T. Maciel, Y. Silva, A. Almeida, D. Gaspar, L. Mendes, P. Neuhaus y W. Dias, "D3.1 Physical layer of the 5G-RANGE – Part I," 2018.
- [90] R. Maunder, "GitHub: polar-3gpp-matlab," 25 10 2019. [En línea]. Available: <https://github.com/robmaunder/polar-3gpp-matlab>. [Último acceso: 15 11 2020].
- [91] B. Yuan y K. Parhi, "Successive cancellation decoding of polar codes using stochastic computing," *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3040-3043, 2015.
- [92] C. Perez, "El Concepto del Canal de Comunicaciones," de *Sistemas de Telecomunicación*, Santander, 2007, p. 5.
- [93] C. Perez, "Receptor," de *Sistemas de Telecomunicación*, Santander, 2007, p. 6.
- [94] C. Perez, "Transmisor," de *Sistemas de Telecomunicación*, Santander, 2007, p. 5.
- [95] S. Malluri y V. K. Pamula, "Gaussian Q-function and Its Approximations," 2013.

ANEXOS

ANEXO A. Implementación en lenguaje .m por medio de un script de un sistema de comunicación con modulación BPSK, cuantificación de punto-fijo y codificación NR-LDPC con un bloque de decodificación SISO MINSUM con características de *Layered* y *Offset*.

ANEXO B. Implementación en lenguaje .m por medio de un script de un sistema de comunicación con modulación BPSK, cuantificación de punto-fijo y codificación NR-Polar con un bloque de decodificación SISO SSC (*Sucesive Simplified Canceled Decoder*)

ANEXO C. Implementación en lenguaje .m por medio de un script de un sistema de comunicación con modulación BPSK, cuantificación de punto-fijo y codificación NR-Polar con un bloque de decodificación SISO SSCL (*Sucesive Simplified Canceled Decoder with List*)

ANEXO D. Resultados obtenidos con las condiciones establecidas en el documento “D3.1 Physical layer of the 5G-RANGE – Part I” para verificar la correcta implementación de los sistemas de comunicaciones con los esquemas de codificación NR-LDPC y NR-Polar.

ANEXO E. Resultados obtenidos de las simulaciones realizadas con los distintos valores establecidos previamente de E_b/N_0 , número de bloques y número de iteraciones y listas con los esquemas de codificación NR-LDPC y NR-Polar, respectivamente.

ANEXO A

```
close all
clear all
clc

%Inicio contador de tiempo global
tinicio=tic;

%Condiciones iniciales
EbNodB=3;
%Establezco EbNo en dB
MaxIter=20;
%Establezco número de iteraciones para decodificación en función de la
recomendación del estándar (7-8)

%Condiciones para cuantización arreglada de punto
rmax=3;
%Límite máximo para cuantificar y verificar valor recibido, si r excede
rmax entonces se asume el valor rmax
maxqr=31;
%Valor de 5 bits para cuantizar la magnitud
maxqL=127;
%Valor de 7 bits para cuantizar la magnitud de L en el decodificador
offset=2;
%Valor de offset

load Matrices_Base/NR_2_6_52.txt;
%Cargo la matriz de expansión
B=NR_2_6_52;
%Defino matriz de expansión
[mb,nb]=size(B);
%Defino el número de filas y columnas
z=52;
%Defino el valor de expansión

%Para cada entrada necesito almacenar z valores. Se necesita declarar
el vector de almacenamiento.

Slen=sum(B(:)~= -1);
%Contador de valores distintos a -1 en B
treg=zeros(max(sum(B~= -1,2)),z);
%Creación del registro de almacenamiento temporal para procesamiento de
filas en función del peso máximo de las filas

kb=(nb-mb);
%Tamaño de la matriz de expansión destinado para información
k=kb*z;
%Número de bits de destinado a información
Rate=1/2;
%Defino tasa de codificación
nbRM=ceil(kb/Rate)+2;
%Defino el número de columnas para RATE MATCHING
n=nbRM*z;
%Número de bits totales considerando RATE MATCHING
mbRM=nbRM-kb;
%Número de filas para realizar RATE MATCHING
```

```

EbNo=10^(EbNodB/10);
%Defino EbNo adimensional
sigma = sqrt(1/(2*(k/(n-2*z))*EbNo));
%Defino sigma en función de la eficiencia - recordar que dos columnas
son usadas para RM y puntuación

%Se fijan variables BER y FER
Nbitrrs=0;
Nblkrrs=0;

%Número de bloques a transmitir
Nblocks=20000;

%Creo vector para almacenar tiempos de procesamiento
tcod=zeros(1,Nblocks);
tdeco=zeros(1,Nblocks);

for i=1:Nblocks

    %Genero mensaje aleatorio
    msg=randi([0 1],1,k);
    %Generación de mensaje de k-bit

    %Codificación
    tic

    cword=nrlldpc(B,z,msg);
    cword=cword(1:n);
    %RATE MATCHING solo transmite una parte del mensaje

    tcod(1,i)=toc;

    %Modulación BPSK
    s = 1 - 2*cword;
    %Modulación BPSK

    %Paso por el canal AWGN
    r = s + (sigma*randn(1,n));
    %Canal AWGN

    %Puntuación del mensaje
    r(1:2*z)=0;
    %Se aplica puntuación de Rate Matching

    %Cuantificación de punto arreglado (Fixed point quantification)
    rq=floor(r/rmax*maxqr);
    %Determino el valor cuantizado de la magnitud - Se convierte en
integral
    rq(rq>maxqr)=maxqr;
    %Si el valor excede maxqr=31 entonces se asigna el valor de 31 -
Clipping
    rq(rq<-(maxqr+1))=-(maxqr+1);
    %Si el valor excede maxqr=-31 entonces se asigna el valor de -31 -
Clipping

    tic

```

```

%Codificación por medio de paso de mensajes iterativos por capas -
SISO

msg_tilde=sisoiterative(B,offset,maxqr,maxqL,k,mBRM,nbRM,Slen,treg,rq,z
,MaxIter);
tdeco(1,i)=toc;

%Conteo de errores
Nrrs=sum(msg~=msg_tilde);

if Nrrs>0
    Nbitrrs = Nbitrrs + Nrrs;
    Nblkrrs=Nblkrrs+1;
end
end

%Calculo BER y FER
BER_sim=Nbitrrs/k/Nblocks;
FER_sim=Nblkrrs/Nblocks;

%Cuento tiempo de procesamiento codificador y decodificador por bloque
y global

tcodfinal=sum(tcod)/Nblocks;
tdecofinal=sum(tdeco)/Nblocks;
tfinal=toc(tinicio);

%Visualizo resultados
disp([EbNodB BER_sim FER_sim tcodfinal tdecofinal tfinal Nblocks
MaxIter])

%Fin del código

```

ANEXO B

```

clear all
close all
clc

tinicio=tic;

%Condiciones iniciales: valores de N y K establecidos para el estándar
5G
N=1024;
%Número de elementos de la matriz de confiabilidad
K=512;
%Secuencia de confiabilidad para N elementos
n=log2(N);
%Potencia elevada a la 2 de N

EbNodB=2.5;
Rate=K/N;
EbNo=10^(EbNodB/10);
sigma=sqrt(1/(2*Rate*EbNo));

%Establezco posiciones frías y para el mensaje

```

```

[Q1,F]=reability(N,K);

%Simulación
Nbiterrs = 0;
Nblkerrs = 0;
Nblocks = 15000;

tcod=zeros(1,Nblocks);
tdeco=zeros(1,Nblocks);

for blk=1:Nblocks
    %Creo mensaje de K bits
    msg=randi([0 1],1,K);

    %Creo codificación
    tic
    cword=nrpolar(N,n,K,Q1,msg);
    tcod(1,blk)=toc;

    %Modulación BPSK
    s=1-cword*2;

    %Envío a través del canal AWGN
    r=s+sigma*randn(1,N);

    %Decodificación SC
    tic
    msg_tilde=sisossc(n,N,K,Q1,F,r);
    tdeco(1,blk)=toc;

    %CSe procede a contar errores
    Nerrs = sum(msg ~= msg_tilde);
    if Nerrs > 0
        Nbiterrs = Nbiterrs + Nerrs;
        Nblkerrs = Nblkerrs + 1;
    end
end

BER_sim = Nbiterrs/K/Nblocks;
FER_sim = Nblkerrs/Nblocks;

%Cuento tiempo de procesamiento codificador y decodificador por bloque
y global

tcodfinal=sum(tcod)/Nblocks;
tdecofinal=sum(tdeco)/Nblocks;
tfinal=toc(tinicio);

%Visualización de resultados

disp([EbNodB BER_sim FER_sim tcodfinal tdecofinal tfinal Nblocks 1])

%Fin del Código

```

ANEXO C

```
close all
clear all
clc

tinicio=tic;

%Condiciones iniciales: valores de N y K establecidos para el estándar
5G
N=1024;
%Número de elementos de la matriz de confiabilidad
%Recuérdese que el estándar con listas exige mensaje + CRC con tamaño
de 11
A=500;
%Tamaño de la información a enviar
crc_long=11;
%Longitud del polinomio CRC
K=A+crc_long;
%Longitud total del mensaje

%Condiciones para listas - creación de CRC
crc_pol=fliplr([1 1 1 0 0 0 1 0 0 0 0 1]);
%Creo polinomio CRC
nL=16;
%Tamaño de la lista

%Condiciones para cuantización arreglada de punto
rmax=4;
%Límite máximo para la cuantización de un valor
maxqr=31;
%Valor cuantizable de 5 bits para la cuantización de la magnitud

%Condiciones de potencia y de eficiencia
n=log2(N);
%Potencia elevada a la 2 de N
EbNodB=2.5;
%EbNo en dB
Rate=A/N;
%Eficiencia del mensaje
EbNo=10^(EbNodB/10);
%Cálculo de EbNo
sigma=sqrt(1/(2*Rate*EbNo));
%Cálculo de sigma para BPSK

%Establezco posiciones frías y para el mensaje
[Q1,F]=reability(N,K);

%Errores
Nbiterrs = 0;
Nblkerrs = 0;

satx=@(x,th) min(max(x,-th),th);
%Función de cuantización
```



```

%Condiciones de simulación
Nblocks = 15<000;

tcod=zeros(1,Nblocks);
tdeco=zeros(1,Nblocks);

for blk=1:Nblocks

    %Creo el mensaje completo A+L
    msg=randi([0 1],1,A);
%Creo mensaje de A bits
    %Procedo a calcular el polinomio CRC
    [quot,rem]=gfdeconv([zeros(1,crc_long) fliplr(msg)],crc_pol);
    msg_crc=[msg fliplr([rem zeros(1,crc_long-length(rem))])];

    %Codificación Polar

    tic
    cword=nrpolar(N,n,K,Q1,msg_crc);
    tcod(1,blk)=toc;

    %Modulación BPSK
    s=1-cword*2;

    %Envío a través del canal AWGN
    r=s+(sigma*randn(1,N));

    %Cuantización
    r=satx(r,rmax);
    rq=round(r/rmax*maxqr);

    %Decodificación SSCL
    tic
    msg_tilde=sisosscl(n,N,K,Q1,F,rq,nL,maxqr,crc_pol,A);
    tdeco(1,blk)=toc;

    %Procedo a contar los errores
    Nerrs = sum(msg ~= msg_tilde);
    if Nerrs > 0
        Nbiterrs = Nbiterrs + Nerrs;
        Nblkerrs = Nblkerrs + 1;
    end
end

BER_sim = Nbiterrs/K/Nblocks;
FER_sim = Nblkerrs/Nblocks;

%Cuento tiempo de procesamiento codificador y decodificador por bloque
y global

tcodfinal=sum(tcod)/Nblocks;
tdecofinal=sum(tdeco)/Nblocks;
tfinal=toc(tinicio);

%Visualización de resultados

```

```
disp([EbNodB BER_sim FER_sim tcodfinal tdecofinal tfinal Nblocks nL])
%Fin del Código
```

ANEXO D

```
%Pruebas de funcionamiento NR-LDPC: BG1, j=1, z=24, maxIter=20

NR_LDPC=[0 0.17033 0.9975 0.0029658 0.36048 7271.5 100000 20;
1 0.003486 0.16925 0.0024109 0.19081 3866.5 100000 20;
1.5 0.0075834 0.0276 0.002623 0.20906 4235.7 100000 20;
2 0.00031468 0.000891 0.0029664 0.23823 4826.9 100000 20;
2.5 8.6174e-06 0.0000554 0.0026836 0.30941 6244 100000 20;
3 1.8939e-07 1.6798e-05 0.0027626 0.25537 5165.1 100000 20];

%Pruebas de funcionamiento NR-Polar: nL=8

nr_polar_8=[0 0.43245 0.96885 0.0024512 0.32405 6552.9 100000; 8;
1 0.10462 0.30095 0.1138 0.34628 9224.1 100000 8;
1.5 0.014777 0.04805 0.0025999 2.7402 54877 100000
8;
2 0.00099276 0.00245 0.0024283 0.33071 6683.1 100000 8;
2.5 3.8748e-05 0.0002 0.0023966 0.39572 7982.3 100000 8;
3 2.4521e-07 5e-05 0.0027626 0.25537 5165.1 100000 8];
```

ANEXO E

```
%% RESULTADOS SIN CODIFICAR
UNCODED=[0,0.07865; 1, 0.037506;2,0.012501; 3,0.0023883;4, 0.00019091];

%% ALMACENAMIENTO DE RESULTADOS NRLDPC BG1 - j=1 - Zc=24

NR_LDPC_BG1_2=[0 0.1678 1 0.0029907 0.021124
2.5485 100 2;
0.5 0.14127 1 0.0029946 0.022319 2.5611
100 2;
1 0.11668 1 0.0028741 0.0215 32.096
1000 2;
1.5 0.087533 1 0.002348 0.017438 151.17
5000 2;
2 0.059835 1 0.0023112 0.016914 195.6
10000 2;
2.5 0.037439 0.99993 0.0022218 0.016687 288.78
15000 2;
3 0.021345 0.9989 0.0022253 0.016707 387.4
20000 2];

NR_LDPC_BG1_4=[0 0.17186 1 0.0026929 0.037846
7.587 100 4;
0.5 0.13661 1 0.0026295 0.039177 7.0764
100 4;
1 0.095875 1 0.002682 0.038407 61.68
1000 4;
1.5 0.048841 0.9852 0.0023312 0.034419 309.19
5000 4;
2 0.015965 0.8423 0.0022555 0.034045 619.83
10000 4;
2.5 0.0031563 0.42853 0.0022265 0.03417 918.36
15000 4];
```

```

3 0.00031714 0.08845 0.002208 0.033964 1228.3
20000 4];

NR_LDPC_BG1_6=[0 0.17388 1 0.0033041 0.063499
6.4435 100 6;
0.5 0.13282 0.99 0.0039329 0.064367 10.897
100 6;
1 0.081206 0.981 0.0034898 0.064626 83.971
1000 6;
1.5 0.02684 0.7052 0.002987 0.065057 400.97
5000 6;
2 0.003196 0.2065 0.0028109 0.062643 793.83
10000 6;
2.5 0.00017247 0.0172 0.0022101 0.054632 1154.29
15000 6;
3 6.6288e-07 0.00025 0.0021473 0.05287 1702.2
20000 6];

NR_LDPC_BG1_8=[0 0.1654 1.0000 0.0037 0.1122 11.6236 100
8.0000;
0.5000 0.1333 0.9800 0.0042 0.1231 12.7715 100
8.0000;
1 0.070205 0.853 0.0023101 0.071355 74.039
1000 8;
1.5 0.016079 0.3856 0.0021959 0.070801 365.45
5000 8;
2 0.0010621 0.0262 0.0021763 0.0707 729.66
10000 8;
2.5 3.952e-05 0.0012667 0.0021861 0.070221 1087.5
15000 8;
3 4.2614e-06 1.756e-05 0.0022084 0.070797 1462
20000 8];

NR_LDPC_BG1_10=[0 0.17479 1 0.0026634 0.090899
9.3851 100 10;
0.5 0.12263 0.96 0.0033336 0.090574 9.4625
100 10;
1 0.064413 0.75 0.0022978 0.089356 91.827
1000 10;
1.5 0.012317 0.2398 0.0021943 0.088624 454.61
5000 10;
2 0.00060265 0.017 0.0022 0.088326 906.18
10000 10;
2.5 1.8939e-05 0.0004 0.0022665 0.091313 1405.2
15000 10];

NR_LDPC_BG1_12=[0 0.17189 1 0.0025535 0.13446
17.398 100 12;
0.5 0.12648 0.98 0.0022001 0.11834 13.104
100 12;
1 0.060555 0.653 0.0023523 0.10927 111.89
1000 12;
1.5 0.0098462 0.1614 0.0022367 0.10696 546.49
5000 12;
2 0.00056307 0.0099 0.0022243 0.10666 1089.8
10000 12;
2.5 1.3773e-05 0.0002 0.0022132 0.10643 1631.4
15000 12];

```

```

NR_LDPC_BG1_16=[0      0.17508      1      0.0030645      0.1463
28.735      100      16;
0.5      0.12612      0.92      0.0026711      0.1473      22.169
100      16;
1      0.054483      0.542      0.0025007      0.1222      225.95
1000      16;
1.5      0.0081739      0.1056      0.0035303      0.12706      1153.7
5000      16;
2      0.00022557      0.0041      0.0034901      0.16521      1892.5
10000      16;
2.5      7.0707e-06      6.6667e-05      0.0023977      0.14258      2481.5
15000      16];

```

```

NR_LDPC_BG1_20=[0      0.16956      1      0.0029676      0.18244
19.148      100      20;
0.5      0.12847      0.95      0.0031313      0.18682      19.024
100      20;
1      0.051019      0.492      0.0024432      0.17918      181.76
1000      20;
1.5      0.0075769      0.0904      0.0026068      0.25029      1267.6
5000      20;
2      0.00033598      0.0034      0.0023928      0.17593      2392.54
10000      20];

```

%% ALMACENAMIENTO DE RESULTADOS NR-LDPC, BG2, j=6, Zc=52

```

NR_LDPC_BG2_2=[0      0.19138      1      0.0029078      0.02382
3.1869      100      2;
0.5      0.16721      1      0.0031165      0.020765
2.4733      100      2;
1      0.14063      1      0.0033266      0.036418      40.036
1000      2
1.5      0.11194      1      0.0033088      0.03746
205.13      5000      2;
2      0.084015      1      0.002968      0.033415      365.6
10000      2;
2.5      0.05846      1      0.002049      0.021318
356.77      15000      2;
3      0.037804      1      0.0018135      0.019661      434.31
20000      2];

```

```

NR_LDPC_BG2_4=[0      0.19156      1      0.011006      0.057227
7.3269      100      4;
0.5      0.15712      1      0.0029828      0.039331
4.3297      100      4;
1      0.1134      1      0.0015862      0.032477      34.367
1000      4;
1.5      0.066482      0.9976      0.0013957      0.030593
160.41      5000      4;
2      0.027614      0.9396      0.0013821      0.030599      320.66
10000      4;
2.5      0.0070519      0.64087      0.0016159      0.035933
564.66      15000      4;
3      0.0010771      0.20465      0.0029792      0.069287      849.9
20000      4
];

```

```

NR_LDPC_BG2_6=[0      0.1875      1.0000      0.0074      0.0700      10.3994
100.0000      6.0000;

```

```

0.5      0.14971      1      0.0023597      0.057976      6.0658
100      6;
1      0.097865      0.988      0.0017019      0.055086      56.926
1000      6;
1.5      0.035924      0.8042      0.001618      0.053455      275.86
5000      6;
2      0.0053781      0.307      0.0016301      0.053029      547.55
10000      6;
2.5      0.0002641      0.033467      0.0016303      0.053341      826.01
15000      6;
3      7.9808e-06      0.00075      0.0016292      0.053068      1095.8
20000      6;
];

NR_LDPC_BG2_8=[0      0.19242      1      0.002997      0.07433
18.933      100      8;
0.5      0.15081      0.99      0.004207      0.07188      11.69
100      8;
1      0.077525      0.871      0.0024772      0.0897      112.38
1000      8;
1.5      0.020693      0.457      0.0025054      0.09217      574.48
5000      8;
2      0.0014804      0.0375      0.0022788      0.10035      1028.9
10000      8;
2.5      2.2308e-05      0.0017333      0.0022506      0.08857      1466.5
15000      8;
3      9.6154e-08      5e-05      0.0018012      0.079039      1621.2
20000      8];

NR_LDPC_BG2_10=[0      0.18623      1      0.0022778      0.090998
9.3983      100      10;
0.5      0.14563      0.99      0.0020853      0.095092
9.7531      100      10;
1      0.075481      0.761      0.0016223      0.087701      89.444
1000      10;
1.5      0.014104      0.2638      0.0016366      0.088634      451.88
5000      10;
2      0.00071192      0.019      0.0016303      0.089471      912.02
10000      10;
2.5      6.5385e-06      0.00033333      0.0016162      0.088774      1357.3
15000      10];

NR_LDPC_BG2_12=[0      0.18885      1      0.0022421      0.11697
11.985      100      12;
1      0.063769      0.672      0.0018595      0.11541      117.44
1000      12;
1.5      0.010684      0.1666      0.0017529      0.11472
582.9      5000      12;
2      0.00034865      0.0075      0.001756      0.11511      1169.7
10000      12;
2.5      1.0256e-06      6.6667e-05      0.0017119      0.11335
1727.4      15000      12;
3      0      0      0.0014647      0.096954      1970.2
20000      12];

NR_LDPC_BG2_16=[0      0.19138      1      0.0019978      0.15502
15.733      100      16;
0.5      0.14287      0.96      0.0018837      0.14833
15.05      100      16;

```

```

1      0.0566      0.533      0.0014367      0.12662      128.16
1000      16;
1.5    0.0079004    0.0952    0.0019862    0.15526
789.11      5000      16;
2      0.00026423    0.0031    0.0017681    0.15501      1568.9
10000      16;
2.5    1.2692e-05    6.6667e-05    0.0013482    0.11882
2304.1      15000      16];

NR_LDPC_BG2_20=[0      0.18744      0.98      0.0022036      0.19461
19.755      100      20;
0.5      0.14      0.92      0.0026596      0.19831      20.173
100      20;
1      0.056052      0.477      0.0018772      0.19758      199.6
1000      20;
1.5    0.0067554      0.0716      0.0017741      0.19298      974.35
5000      20;
2      0.00018596      0.002      0.001799      0.19475      1966.5
10000      20;
2.5    1.2821e-05    0.00013333      0.00173      0.18959      2871.4
15000      20;
];

```

ORDEN DE EMPASTADO