



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – TELECOMUNICACIONES

CODIFICACIÓN DESIGUAL PARA EL CONTROL DE ERRORES EN
IMÁGENES DIGITALES EN UN AMBIENTE DE RADIOS DEFINIDOS POR
SOFTWARE.

TESIS
QUE PARA OPTAR EL GRADO DE:
MAESTRA EN INGENIERÍA

PRESENTA:
ING. ANA MARÍA LUISA RODRÍGUEZ ARMAS

TUTOR PRINCIPAL:
DR. FRANCISCO JAVIER GARCÍA UGALDE

CIUDAD DE MÉXICO, FEBRERO 2017

JURADO ASIGNADO:

Presidente: Dr. Psenicka Bohumil

Secretario: Dr. Gómez Castellanos Javier

1^{er}. Vocal: Dr. García Ugalde Francisco

2^{do}. Vocal: Dr. Matías Maruri José María

3^{er}. Vocal: Dr. Rivera Rivera Carlos

Lugar donde se realizó la tesis:
Laboratorio de codificación y seguridad de los sistemas de información.
Posgrado de Ingeniería.
Universidad Nacional Autónoma de México (UNAM).
Ciudad Universitaria, Ciudad de México.

TUTOR DE TESIS:

Dr. Francisco Javier García Ugalde

FIRMA

Agradecimientos

A Dios, por darme fuerza y fe para superar la idea de lo que me parecía imposible terminar.

A mi madre, María de los Ángeles Armas González por nunca soltarme durante esta travesía, y darme su apoyo incondicional, te amo mamá.

A mis hermanos y sobrinos, por impulsarme a ser mejor cada día.

Al Dr. Francisco Javier García Ugalde por darme la oportunidad de realizar este trabajo, por su confianza, y ayuda cuando más lo necesité, gracias.

A Daniel Guerrero que, durante tanto tiempo siempre ha creído en mí, por su amor y paciencia, muchas gracias, te adoro.

A mis compañeros de la maestría, por su apoyo, en especial a Miguel Ángel Alvarado, que durante este tiempo me tendió su mano para nunca dejarme caer. Gracias por los consejos, exigencias, confianza, y el cariño que me has brindado, te quiero mucho.

A mis amistades que desde que empecé este trabajo, me motivaron a seguir adelante, pese a las circunstancias que la vida me dejó.

Esta investigación fue realizada en parte gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IN-106816 “Algoritmos robustos y aplicaciones de seguridad con marca de agua codificada para imágenes digitales”. Por lo cual agradezco a la DGAPA-UNAM la beca recibida.

Resumen

El presente trabajo, consistió en proponer una metodología de diseño para implementar un esquema de protección desigual contra errores (UEP, Unequal error protection por sus siglas en inglés). En el desarrollo de la experimentación, la técnica UEP fue aplicada en imágenes digitales, ordenando la información en capas binarias, conforme a su nivel de significancia. Se utilizan dos familias de códigos correctores de errores propuesta: códigos de bloque del tipo BCH, y códigos convolucionales.

La metodología propuesta consiste en proteger las capas binarias de la imagen antes de su transmisión, de acuerdo a su importancia relativa. Durante la experimentación, para poder llevar a cabo una comparación, se analizaron esquemas de protección igual contra errores (EEP, por sus siglas en inglés), y desigual UEP. La metodología se validó considerando la imagen en la recepción del sistema de comunicaciones, mediante la medición y comparación de los parámetros de calidad, como son la relación señal pico a ruido, (PSNR, por sus siglas en inglés), tasa de bit en error (BER, por sus siglas en inglés), y el índice de semejanza estructural (SSIM, por sus siglas en inglés). El mejor esquema UEP, se escogió mediante la comparación de los diversos esquemas propuestos, con respecto al esquema sin protección contra errores, y protección igual EEP, fijando la razón señal a ruido E_b/N_0 en un cierto nivel de operación.

Para las simulaciones y experimentación física, el diseño e implementación del sistema de comunicaciones propuesto, se desarrolló en Matlab y GNU Radio respectivamente, empleando el dispositivo SDR USRP2. De esta manera, la validación de la metodología se hizo mediante las pruebas realizadas en el sistema de comunicaciones simulado en Matlab, y posteriormente, se aplicó en los USRP2, mediante dos etapas: 1) la etapa de la codificación/decodificación de la imagen realizada en Matlab, y 2) la etapa de la transmisión/recepción de la imagen implementada en GNU Radio.

De acuerdo a la validación de la metodología UEP en los escenarios de comunicación simulado, y físico real, con respecto a los resultados obtenidos con el esquema EEP, se muestra que el sistema UEP basado en códigos convolucionales, con el uso del concepto de RCPC (Rate-Compatible Punctured Convolutional Codes), logró disminuir la redundancia del orden de un 50%, un ahorro en el tiempo de ejecución del 44%, y disminución del tiempo de transmisión de 20%, sin deteriorar la calidad en términos de BER, PSNR y SSIM de la imagen recibida. Por otro lado, también con respecto al esquema EEP, con el

esquema UEP basado en códigos BCH, se obtuvo un ahorro en el tiempo de ejecución de casi el 25%, como se observa, menor al logrado con los códigos convolucionales. En conclusión, la metodología UEP funcionó correctamente para ambos sistemas de protección contra errores, y escenarios de comunicación, resaltando que para BCH, los parámetros de calidad en el esquema UEP fueron mayores con respecto al sistema basado en códigos convolucionales, pero sin lograr disminuir tanto la redundancia, como en el caso de los códigos convolucionales. Esto debido a que, si se buscaba reducir esta aún más, existía el riesgo de perder la calidad visual de la imagen recibida.

Abstract

This thesis work was focused on proposing a design methodology to implement the scheme of Unequal Error Protection (UEP). UEP technique was applied to digital images, whose information was organized into binary layers, according to their relative significance level. Using two great families of very important error-correcting codes in actual current applications, the protection systems analyzed in the proposed methodology, were based on block codes of type BCH, and convolutional codes.

The proposed methodology consists to protect binary layers of a digital image prior to its transmission in a communication channel, organized per their relative weight. During experimentation and to do a comparison, Equal Error Protection schemes (EEP), and unequal UEP were analyzed. The methodology was validated by measuring and comparing the quality parameters such as the Peak Signal-to-Noise Ratio (PSNR), bit error rate (BER), and Structural Similarity Index (SSIM) in the reception side of the communication system. The best UEP scheme has been chosen by comparing several schemes proposed with respect to the unprotected scheme, and equal protection EEP, for a certain signal to noise ratio E_b/N_0 level.

For simulations and physical experimentation, the design and implementation of the proposed communications system was developed in Matlab and GNU Radio respectively, using the SDR device USRP2. In this way, the validation of the methodology was the result of several tests performed in the simulated communications system in Matlab, and later, it was applied in the USRP2, through two stages: 1) the image coding/decoding performed in Matlab; 2) the transmission/reception of the image implemented in GNU Radio.

Concerning the validation of the UEP methodology in the simulated and physical real communication scenarios, with respect to the results given from the EEP scheme, it is shown that the UEP system based on convolutional codes, with the use of the RCPC concept, could reduce the redundancy to almost 50%, having a saving in execution time of 44%, and in transmission time of 20%, without diminishing the quality in terms of BER, PSNR and SSIM of the received image. On the other hand, also with respect to the EEP scheme, the UEP scheme based on BCH codes resulted in a saving in execution time of almost 25%, as it is shown, less than the obtained by convolutional codes. In conclusion,

UEP worked correctly for both error protection systems, and communication scenarios, highlighting that for BCH, the quality parameters in the UEP scheme were greater than those obtained in the system based on convolutional codes, but without achieving the reduction in its redundancy as in the case of the convolutional codes. This is because trying to reduce it much more, has the risk of losing visual quality of the received image.

Contenido

Resumen.....	I
Abstract	III
Contenido.....	V
Lista de acrónimos.....	VIII
Lista de símbolos.....	IX
Lista de figuras.....	X
Lista de tablas.....	XIII
Introducción	1
Introducción	1
Justificación.....	2
Planteamiento del problema	3
Alcance.....	3
Objetivos	4
Metodología.....	4
1. Estado del arte	6
1.1 Estudios centrados en el uso de la Protección Desigual contra Errores (UEP) en imágenes.....	6
1.2 Aplicaciones actuales basadas en la tecnología de Radio Definido por Software (SDR).....	10
2. Marco Teórico.....	13

2.1	Antecedentes.	13
2.1.1	Orígenes de la Protección Desigual contra Errores.....	13
2.1.2	Orígenes de la aplicación de Radio Definido por software.....	14
2.2	Modulación y modelo de canal inalámbrico.....	14
2.2.1	Modulación BPSK.....	14
2.2.2	Modulación GFSK.....	17
2.2.3	Canal AWGN.....	19
2.3	Codificación de canal.....	19
2.3.1	Códigos BCH.....	21
2.3.2	Códigos Convolucionales.....	27
2.4	Esquemas de Protección contra errores.....	37
2.5	Plataforma de Radio Definido por Software.....	37
2.5.1	Radio Definido por Software (SDR).....	38
2.5.2	GNU Radio	43
3.	Metodología propuesta para el diseño de un sistema de control de errores desigual UEP, en imágenes	48
3.1	Descripción del sistema de comunicaciones propuesto	48
3.2	Descripción de la metodología propuesta.	50
3.2.1	Métodos de medición de calidad de imagen.	50
3.2.2	Diseño UEP para códigos correctores binarios: de bloque BCH y convolucional.....	52
4.	Implementación y análisis de resultados en el entorno de simulación por computadora	55
4.1	Selección y procesamiento de la imagen.....	56
4.2	Pruebas y resultados del esquema UEP realizadas en el sistema simulado en computadora.....	61
4.2.1	Simulación sin codificación, y con codificación para códigos BCH, empleando los esquemas UEP- EEP	61

4.2.2	Simulación sin codificación y con codificación con códigos convolucionales empleando los esquemas UEP- EEP	76
5.	Implementación y análisis de resultados en el entorno físico con USRP2.....	103
5.1	Pruebas físicas de la estrategia UEP en la plataforma de radios definidos por software USRP2.....	103
5.1.1	Desarrollo y resultado experimental para comparación de los casos: sin codificación, y con códigos BCH, empleando EEP- UEP	108
5.1.2	Desarrollo y resultado experimental para comparación de los casos: sin codificación, y con códigos convolucionales, empleando EEP- UEP	114
5.2	Análisis de resultados finales.....	120
6.	Conclusiones y trabajo a futuro.....	123
6.1	Conclusiones	123
6.2	Trabajo a futuro	124
	Referencias	125

Lista de acrónimos

A/D	Analogic/Digital
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
BER	Bit Error Rate
BPSK	Binary Phase-shift keying
D/A	Digital/Analogic
DAS	Distributed Antenna System
DDC	Digital Down Converter
DUC	Digital Up Converter
EEP	Equal Error Protection
FPGA	Field Programmable Gate Array
GF	Galois Field
GFSK	Gaussian frequency-shift keying
LDCP	Low-Density Parity-Check
LFSR	Linear Feedback Shift Register
MFT	Multi-Fold Turbo
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
PCC	Punctured Convolutional Code
PER	Pixel Error Rate
PSNR	Peak Signal to Noise Ratio
RCPC	Rate-Compatible Punctured Convolutional Codes
ROI	Region Of Interest
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
SSIM	Structural Similarity Index
UEP	Unequal Error Protection
UHD	Universal Hardware Driver
USRP	Universal Software Radio Peripheral
VCO	Voltage-Controlled Oscillator

Lista de símbolos

$s_i(t)$	Señal modulada PSK.
E	Energía del símbolo.
T	Tiempo de duración del símbolo.
t	Tiempo.
M	Modulador M ario.
m	Bits por símbolo.
P_b	Probabilidad de bit en error.
Δ_f	Desviación máxima de frecuencia.
$V_m(t)$	Voltaje máximo de la señal moduladora binaria.
k_l	Sensibilidad a la desviación.
f_s	Frecuencia de espacio.
f_m	Frecuencia de marco.
$\frac{E_b}{N_0}$	Relación señal a ruido en términos de la energía por bit y densidad espectral.
k	Bit de información.
n	Longitud del bloque.
R	Tasa del código.
t	Capacidad de corrección de errores.
d	Distancia mínima entre palabras.
$g(x)$	Polinomio generador.
$m_i(x)$	Polinomio mínimo.
α	Raíz del polinomio generador.
$c(x)$	Palabra de código.
$i(x)$	Bits de información
$r(x)$	Residuo.
$\bar{e}(x)$	Polinomio del error.
$p(x)$	Polinomio recibido.
S_j	Síndromes.
$\sigma(x)$	Polinomio localizador de errores.
d_i	Discrepancia en la iteración i .
$e(x)$	Polinomio de error (búsqueda de Chien)
K	Longitud de restricción.
P	Matriz de perforación.
d_f	Distancia libre.
Γ	Métrica acumulada.

Lista de figuras

<i>Figura 1.1: Diagrama de la comunicación estenográfica usando Turbo codificación UEP.</i>	7
<i>Figura 1.2: Diagrama del procedimiento de incorporación de marca de agua con UEP.</i>	8
<i>Figura 1.3: Diagrama de bloques del esquema UEP sobre ROI para imágenes y videos.</i>	9
<i>Figura 1.4: Configuración del banco de pruebas en el escenario de Hogar digital.</i>	11
<i>Figura 1.5: Diagrama de bloques del transmisor basado en el sistema OFDM usando codificación de canal.</i>	12
<i>Figura 1.6: Diagrama de bloques del receptor basado en el sistema OFDM usando codificación de canal.</i>	12
<i>Figura 2.1: Diagrama de constelación de la modulación BPSK.</i>	15
<i>Figura 2.2: Diagrama general de un transmisor BPSK.</i>	15
<i>Figura 2.3: Forma de onda de la salida modulada en BPSK en función del tiempo.</i>	16
<i>Figura 2.4: Diagrama general de un receptor BPSK.</i>	16
<i>Figura 2.5: Esquema general de un transmisor GFSK y su forma de onda a la salida.</i>	18
<i>Figura 2.6: Esquema general de un receptor GFSK.</i>	18
<i>Figura 2.7: Arquitectura general del codificador BCH.</i>	23
<i>Figura 2.8: Palabra codificada resultante de la codificación sistemática BCH.</i>	23
<i>Figura 2.9: Arquitectura de un decodificador BCH con aritmética GF (2^m).</i>	24
<i>Figura 2.10: Codificador convolucional con longitud de restricción K y tasa $R=b/n$.</i>	28
<i>Figura 2.11: Codificador convolucional con longitud de restricción $K=3$ y tasa $R=1/2$.</i>	29
<i>Figura 2.12: Codificador convolucional con longitud de restricción $K=3$ y tasa $R=1/2$.</i>	30
<i>Figura 2.13: Diagrama de trellis con longitud de restricción $K=3$ y tasa $R=1/2$ para una entrada 1010.</i>	31
<i>Figura 2.14: Diagrama del codificador convolucional de tasa de codificación de $1/n$ para códigos perforados.</i>	32
<i>Figura 2.15: Representación del algoritmo de Viterbi.</i>	34
<i>Figura 2.16: Selección de ramas supervivientes para decodificación convolucional.</i>	36
<i>Figura 2.17: Palabra resultante de la decodificación, bajo el criterio de máxima verosimilitud.</i>	36
<i>Figura 2.18: Diagrama de bloques general del funcionamiento de un sistema de radio.</i>	38
<i>Figura 2.19: Diagrama de bloques general de la tecnología SDR.</i>	39
<i>Figura 2.20: Imagen del SDR USRP N210.</i>	43
<i>Figura 2.21: Arquitectura GNU Radio.</i>	44
<i>Figura 2.22: Ejemplo de la transmisión y modulación de una señal en GNU Radio Companion.</i>	45
<i>Figura 2.23: Señal Rx del ejemplo de la Figura 2.22.</i>	46

<i>Figura 3.1: Sistema de comunicaciones digital inalámbrico empleado.</i>	49
<i>Figura 4.1: Imagen original “Tux”.</i>	56
<i>Figura 4.2: Imagen “Tux” dividida en 8 planos ordenados por su significancia: a) capa menos significativa. h) capa más significativa.</i>	57
<i>Figura 4.3: Sistema de comunicaciones sin codificación de canal en Simulink.</i>	58
<i>Figura 4.4: Imagen recibida con una BER $\approx 10^3$. i) Capa 1 con una $E_b/N_0 = 4$ [dB]. ii) Errores en “Tux”.</i>	58
<i>Figura 4.5: Imagen recibida con una BER $\approx 10^3$. iii) Capa 2 con una $E_b/N_0 = 4$ [dB]. iv) Errores en “Tux”.</i>	58
<i>Figura 4.6: Imagen recibida con una BER $\approx 10^3$. v) Capa 3 con una $E_b/N_0 = 4$ [dB]. vi) Errores en “Tux”.</i>	59
<i>Figura 4.7: Imagen recibida con una BER $\approx 10^3$. vii) Capa 4 con una $E_b/N_0 = 4$ [dB]. viii) Errores en “Tux”.</i>	59
<i>Figura 4.8: Imagen recibida con una BER $\approx 10^3$. ix) Capa 5 con una $E_b/N_0 = 4$ [dB]. x) Errores en “Tux”.</i>	59
<i>Figura 4.9: Imagen recibida con una BER $\approx 10^3$. xi) Capa 6 con una $E_b/N_0 = 4$ [dB]. xii) Errores en “Tux”.</i>	60
<i>Figura 4.10: Imagen recibida con una BER $\approx 10^3$. xiii) Capa 7 con una $E_b/N_0 = 4$ [dB]. xiv) Errores en “Tux”.</i>	60
<i>Figura 4.11: Imagen recibida con una BER $\approx 10^3$. xv) Capa 8 con una $E_b/N_0 = 4$ [dB]. xvi) Errores en “Tux”.</i>	60
<i>Figura 4.12: Diagrama de flujo adaptado al software Matlab de la metodología para códigos BCH presentada en el capítulo 3.</i>	62
<i>Figura 4.13: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 2, obtenidos para la imagen “Tux”.</i>	65
<i>Figura 4.14: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 3, obtenidos para la imagen “Tux”.</i>	67
<i>Figura 4.15: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 4, obtenidos para la imagen “Tux”.</i>	69
<i>Figura 4.16: Imágenes “Tux” resultantes y de error de las pruebas 2, 3 y 4.</i>	71
<i>Figura 4.17: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 4, obtenidos para la imagen “Lena”.</i>	74
<i>Figura 4.18: “Lena” resultante de la prueba 3, con códigos BCH. 1) Imagen recibida. 2) Errores de la imagen recibida.</i>	75
<i>Figura 4.19: Diagrama de flujo adaptado de la metodología propuesta en el capítulo 3 con códigos convolucionales.</i>	77
<i>Figura 4.20: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional de la prueba 2, obtenidos para la imagen “Tux”.</i>	81

<i>Figura 4.21: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional de la prueba 3, obtenidos para la imagen “Tux”.</i>	83
<i>Figura 4.22: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional de la prueba 4, obtenidos para la imagen “Tux”.</i>	86
<i>Figura 4.23: Imágenes “Tux” resultantes y de error de las pruebas 2, 3 y 4.</i>	88
<i>Figura 4.24: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 5, obtenidos para la imagen “Tux”.</i>	91
<i>Figura 4.25: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 6, obtenidos para la imagen “Tux”.</i>	93
<i>Figura 4.26: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 7, obtenidos para la imagen “Tux”.</i>	95
<i>Figura 4.27: Imágenes “Tux” resultantes y de error de las pruebas 5, 6 y 7.</i>	97
<i>Figura 4.28: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 7, obtenidos para la imagen “Lena”.</i>	100
<i>Figura 4.29: “Lena” resultante de la prueba 7 convolucional, con RCPC. 1) Imagen recibida. 2) Errores de la imagen recibida.</i>	101
<i>Figura 5.1: Diseño del Transmisor para USRP2.</i>	105
<i>Figura 5.2: Espectro transmitido en banda base.</i>	106
<i>Figura 5.3: Espectro recibido en banda base.</i>	107
<i>Figura 5.4: Diseño del Receptor para USRP2.</i>	107
<i>Figura 5.5: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación en BCH obtenida de “Tux”, empleando el USRP2.</i>	110
<i>Figura 5.6: “Tux” resultante con 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación BCH con UEP. 4) Errores residuales con codificación BCH, con UEP.</i>	111
<i>Figura 5.7: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación en BCH obtenida de “Lena”, empleando el USRP2.</i>	112
<i>Figura 5.8: “Lena” resultante de 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación BCH con UEP. 4) Errores con codificación BCH con UEP.</i>	113
<i>Figura 5.9: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación convolucional RCPC obtenidas de “Tux”, con el uso de USRP2.</i>	116
<i>Figura 5.10: “Tux” resultante con 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación convolucional y RCPC para UEP. 4) Errores residuales con codificación convolucional y RCPC para UEP.</i>	117
<i>Figura 5.11: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación convolucional RCPC obtenidas de “Lena”, con el uso de USRP2.</i>	118
<i>Figura 5.12: “Lena” resultante con 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación convolucional y RCPC, con UEP. 4) Errores residuales con codificación convolucional y RCPC, con UEP.</i>	119

Lista de tablas

<i>Tabla 2.1: Ventajas y desventajas del SDR.</i>	41
<i>Tabla 2.2: Características de USRP por modelo.</i>	42
<i>Tabla 3.1: Esquemas de modulación en términos de Probabilidad de bit en error (P_b), y Tasa de bit de error (BER).</i>	51
<i>Tabla 4.1: Código BCH escogido para la prueba 1 empleando el esquema EEP.</i>	63
<i>Tabla 4.2: Valores de calidad resultantes de la prueba 1 con un código BCH para diferentes E_b/N_0 para la imagen “Tux”.</i>	63
<i>Tabla 4.3: Códigos BCH para la prueba 4 empleando el esquema UEP.</i>	64
<i>Tabla 4.4: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 2.</i>	66
<i>Tabla 4.5: Códigos BCH para la prueba 3 empleando el esquema UEP.</i>	66
<i>Tabla 4.6: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 3.</i>	68
<i>Tabla 4.7: Códigos BCH para la prueba 4 empleando el esquema UEP.</i>	68
<i>Tabla 4.8: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 4.</i>	70
<i>Tabla 4.9: Concentrado de resultados con códigos BCH, empleando el esquema UEP.</i>	70
<i>Tabla 4.10: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 3, obtenidos de “Lena”.</i>	75
<i>Tabla 4.11: Comparación de los resultados de las imágenes “Tux” y “Lena” empleando códigos BCH con UEP.</i>	75
<i>Tabla 4.12: Código convolucional escogido para la prueba 1 empleando el esquema EEP.</i>	78
<i>Tabla 4.13: Valores de calidad resultantes de la prueba 1 con un código convolucional para diferentes E_b/N_0 obtenidos de con la imagen “Tux”.</i>	78
<i>Tabla 4.14: Código convolucional escogido para la prueba 2 empleando el esquema UEP.</i>	79
<i>Tabla 4.15: Tiempos de ejecución para distintos esquemas con codificación convolucional de la prueba 2.</i>	82
<i>Tabla 4.16: Código convolucional escogido para la prueba 3 empleando el esquema UEP.</i>	82
<i>Tabla 4.17: Tiempos de ejecución para distintos esquemas con codificación convolucional de la prueba 3.</i>	84
<i>Tabla 4.18: Código convolucional escogido para la prueba 6 empleando el esquema UEP.</i>	85
<i>Tabla 4.19: Tiempos de ejecución para distintos esquemas con codificación convolucional de la prueba 4.</i>	85
<i>Tabla 4.20: Concentrado de resultados con códigos convolucionales empleando el esquema UEP.</i>	87

<i>Tabla 4.21: Código convolucional con estrategia RCPC, establecido para la prueba 5 y con UEP...</i>	90
<i>Tabla 4.22: Tiempos de ejecución para distintos esquemas con codificación convolucional, con RCPC de la prueba 5.</i>	92
<i>Tabla 4.23: Código convolucional con RCPC, escogido para la prueba 6 con UEP.</i>	92
<i>Tabla 4.24: Tiempos de ejecución para distintos esquemas con codificación convolucional con RCPC, de la prueba 6.</i>	94
<i>Tabla 4.25: Código convolucional con RCPC, escogido para la prueba 7 con UEP.</i>	94
<i>Tabla 4.26: Tiempos de ejecución para distintos esquemas con codificación convolucional con RCPC, de la prueba 7.</i>	96
<i>Tabla 4.27: Concentrado de resultados con códigos convolucionales con RCPC, empleando UEP. ..</i>	96
<i>Tabla 4.28: Tiempos de ejecución para distintos esquemas con codificación convolucional con RCPC, de la prueba 7 obtenidos con “Lena”.</i>	101
<i>Tabla 4.29: Comparación de los resultados de las imágenes “Tux” y “Lena” para códigos convolucionales, con RCPC con UEP.</i>	102
<i>Tabla 5.1: Código BCH escogido para el desarrollo experimental con EEP en la plataforma USRP2.</i>	108
<i>Tabla 5.2: Códigos BCH usados para el desarrollo experimental con UEP en la plataforma USRP2.</i>	108
<i>Tabla 5.3: Tiempos de tx/rx para diferentes esquemas con codificación BCH mediante el USRP2, obtenidos de “Tux” y “Lena”.</i>	113
<i>Tabla 5.4: Comparación de resultados de las imágenes “Tux” y “Lena” empleando códigos BCH, con USRP2.</i>	113
<i>Tabla 5.5: Código convolucional escogido para el desarrollo experimental con EEP en el USRP2.</i>	114
<i>Tabla 5.6: Códigos convolucionales con RCPC usados en el desarrollo experimental con UEP, en el USRP2.</i>	115
<i>Tabla 5.7: Tiempos de tx/rx para diferentes esquemas con codificación convolucional mediante el USRP2, obtenidos de “Tux” y “Lena”.</i>	119
<i>Tabla 5.8: Comparación de resultados de las imágenes “Tux” y “Lena” empleando códigos convolucionales con UEP, mediante la plataforma USRP2.</i>	119
<i>Tabla 5.9: Concentrado de resultados para diferentes escenarios y códigos, obtenidos con “Tux”. ..</i>	121

Introducción

Introducción

En los sistemas de comunicaciones digitales inalámbricos existen diversas causas que provocan la alteración de la señal recibida, tales como el ruido generado por el propio canal de comunicaciones, en que las propiedades físicas del medio y el receptor tienen un papel importante en la propagación de la señal [1]. El ruido del canal produce errores de transmisión, que hacen que la señal reconstruida por el receptor no sea la misma que la señal transmitida; por ese motivo, es indispensable crear un diseño óptimo del sistema de comunicaciones para que el control de errores en la recepción de los datos sea confiable.

En 1948, Shannon [2] demostró que, mediante la codificación adecuada de la información, los errores inducidos por un canal ruidoso se pueden reducir a cualquier nivel sin sacrificar la tasa de transmisión. Entonces, para controlar dichos errores de transmisión, es necesario introducir algún mecanismo que permita detectar los errores y además corregirlos.

Con el empleo de la codificación de canal, se debe buscar la fidelidad de la transmisión la cual se mide en términos de la probabilidad de bit en error, requiriendo una estrategia al usar los códigos correctores de errores para obtener la probabilidad mínima [3]. Dentro del contexto anterior, existen diferentes escenarios de protección en redes inalámbricas, donde aplicando uniformemente una buena protección contra errores, es decir, usando una misma codificación para toda la información, se puede convertir en un exceso, debido a que requiere mayor uso de ancho de banda, y a su vez, podría resultar ser un desperdicio proteger por igual toda la información por ejemplo, en una red inalámbrica en donde la información de la capa física debe estar mejor protegida que la carga útil para asegurarse de que la información es entregada satisfactoriamente, a pesar que el objetivo principal es entregar los datos de carga útil [4].

Para evitar procesar la información menos significativa por igual que la más significativa, se recurre al método de Protección Desigual contra Errores (UEP), el cual consiste en la

asignación de la redundancia propia de la codificación en función de la importancia de los bits de información [4]. Las técnicas de UEP han sido diseñadas para aplicaciones dirigidas a transmisión de imágenes y almacenamiento, las cuales se basan en el uso de códigos correctores de errores.

El presente trabajo conduce a la investigación en la implementación de un sistema de comunicaciones digital inalámbrico basado en la tecnología flexible conocida como Radio Definido por Software (SDR por sus siglas en inglés), teniendo la ventaja de que se trata de un equipo de propósito general en el ámbito de las comunicaciones inalámbricas, que se puede emplear para la transmisión de información, logrando obtener la solución al problema sobre los errores que resulten en los datos obtenidos producidos por el canal de comunicaciones, realizando la detección y corrección de los errores en el receptor; además con el propósito de mejorar el desempeño del sistema, es posible introducir la técnica de UEP comparándolo con la Protección Igual contra Errores (EEP por sus siglas en inglés), empleando diferentes codificadores junto con métodos de control de errores de tasa compatible (RCPC por sus siglas en inglés), permitiendo una adaptación a los parámetros de calidad relativa del tipo de información que se esté recibiendo.

Justificación

- Generar aportaciones a la plataforma SDR al realizar un sistema de comunicaciones digitales, considerando los modelos de la teoría de códigos y la implementación de la codificación desigual de canal.
- Con base en la teoría de la información, obtener la experiencia en el diseño de la metodología al emplear los algoritmos óptimos referentes a la detección y corrección de errores, para la construcción del sistema de comunicaciones empleando UEP.
- Demostrar ventajas significativas del mecanismo UEP sobre EEP en el sistema de comunicaciones propuesto.

Planteamiento del problema

En los sistemas de comunicaciones digitales, existe la necesidad de implementar mayor protección a cierta información comparada con el resto de otra información, debido a que en aplicaciones multimedia se requieren grandes cantidades de datos para transmitir a través de canales inalámbricos, donde la información es susceptible de ser alterada por ruido que a su vez, suelen ser sensibles en las diferentes capas de la red [4]; en consecuencia, se requiere usar recursos abundantes de energía y ancho de banda que a la larga afectan económicamente. Por lo tanto, es necesario proponer un sistema de comunicaciones viable que soporte la Protección Desigual contra Errores (UEP) para la transmisión multimedia, manteniendo una complejidad computacional baja, disminuyendo el ancho de banda y comprobando el beneficio de su uso con respecto a la Protección Igual contra Errores (EEP), para lograr que la información se reciba con medidas resultantes de parámetros de calidad cuantitativa y cualitativa aceptables.

Alcance

Este trabajo investiga el desempeño del esquema de la Protección Desigual contra Errores (UEP) sobre el esquema convencional de Protección Igual contra Errores (EEP) en una imagen digital monocromática en formato binario (bmp), empleando el uso de codificación de bloque y convolucional junto con métodos de control de errores de tasa compatible (RCPC). Experimentalmente, esta investigación consiste en dos pruebas: la primera por simulación, en donde se aplica el modelo de canal de comunicación con Ruido Auditivo Blanco Gaussiano (AWGN por sus siglas en inglés) para una fuente de datos proveniente de una imagen monocromática de $m \times n$ y 8 bits por pixel, en un cierto rango de valores de relación señal a ruido (SNR por sus siglas en inglés); la segunda por experimentación, bajo el esquema de la tecnología SDR, se realizaron las pruebas en un canal de comunicación real, en el cual influyen fenómenos naturales tales como el ruido e interferencias electromagnéticas, empleando los mismos conceptos de codificación, esquemas de protección y fuentes de información que se realizaron en la simulación.

Objetivos

- Desarrollar un sistema de transmisión en un canal inalámbrico en simulación y con el uso de la plataforma SDR, mediante codificación de control de errores efectuando los principios de la protección desigual contra errores (UEP) y la protección igual contra errores (EEP).
- Implementar y probar el desempeño de la codificación de canal (bloque y convolucional) para la transmisión de datos.
- Evaluar el desempeño de los dos esquemas de control de errores (UEP y EEP) bajo el sistema de comunicaciones empleado.

Metodología

Con el propósito de cumplir con los objetivos propuestos en este trabajo de tesis, se presenta la siguiente metodología:

- Realizar la revisión documental (artículos, libros, tesis, etc.) acerca del estado del arte, respecto al desarrollo de los SDR para corroborar la viabilidad de este trabajo; investigación de lo más relevante sobre la aplicación actual del concepto de Protección Desigual contra Errores (UEP) empleando códigos correctores y detectores de errores para diferentes usos.
- Estudiar los algoritmos de los códigos correctores de errores para seleccionar y ajustar los parámetros indicados de acuerdo a la necesidad de este trabajo.
- Seleccionar adecuadamente los códigos correctores de errores, realizando diversas pruebas de simulación en el ambiente de Matlab, para estudiar y comprobar el comportamiento del funcionamiento de la codificación desigual (UEP) y la codificación igual (EEP) sobre la fuente de información propuesta en el sistema de comunicaciones empleado.
- Analizar la plataforma SDR en el software y el hardware empleado. Para el caso de este proyecto, se usarán dos radios USRP N210 de la compañía ETTUS Research y

el ambiente de desarrollo de software Matlab 2014b, para establecer la comunicación entre ambos, con el fin de comprobar la viabilidad de las pruebas.

- Aplicar los códigos correctores de errores analizados junto con los conceptos EEP y UEP sobre el sistema de comunicaciones digitales propuesto en los radios, dentro de ambientes de desarrollo de Matlab y GNU radio.
- Medir los resultados sobre las imágenes digitales recibidas en los radios a partir de los parámetros de calidad que a continuación se mencionan: la Relación Señal Pico a Ruido (por sus siglas en inglés PSNR), Tasa de Error Binario (BER por sus siglas en inglés), e Índice de Similitud Estructural (SSIM por sus siglas en inglés), con el propósito de comprobar la correcta detección y corrección de errores, comparación entre códigos, la calidad y fidelidad de la recepción de la imagen.
- Analizar los resultados para posteriormente validar el funcionamiento apropiado del sistema de comunicaciones empleado y del esquema de codificación definido.

Capítulo 1

Estado del arte

1.1 Estudios centrados en el uso de la Protección Desigual contra Errores (UEP) en imágenes.

La Protección Desigual contra Errores (UEP) es una técnica usada en aplicaciones para transmisiones digitales como video, imágenes, voz y datos, para la cual se necesita proteger parte del contenido de la información que se considere más relevante cuando es transmitido a través de un canal inalámbrico; para propósito de esta investigación, se centrara en la aplicación del UEP en imágenes.

Existen diversos trabajos anteriores que han abordado el problema de la transmisión multimedia a través de canales de comunicaciones ruidosos.

El término de codificación UEP fue abordado por primera vez por Masnick y Wolf en [5], donde a cada palabra de código se le asignó un nivel de protección \bar{f} ; si f errores se producían en la recepción de la palabra de código, entonces todos los dígitos que tuvieran protección \bar{f} mayor o igual a f , podrían ser decodificados correctamente, aunque la palabra de código no pudiera ser decodificada exitosamente. En [6] la técnica UEP se lleva a cabo en primer lugar, optimizando el número de encabezados asignados a las 5 sub-bandas de una imagen JPEG para evitar la propagación de errores y, en segundo lugar, mediante la optimización del número de retransmisiones con el propósito de reducir al mínimo la distorsión del canal. Además, se compararon los esquemas UEP y EEP sobre la imagen

JPEG, utilizando Turbo códigos y códigos convolucionales, donde el esquema UEP resulto tener un mejor rendimiento de procesamiento que el esquema EEP.

En [7] los autores propusieron un esquema de codificación de errores para la transmisión de imágenes con técnica de estenografía, como se muestra en la Figura 1.1, el cual consistió en aplicar dos Turbo códigos con la misma tasa de codificación y a la salida, empleó un doble mecanismo de RCPC, para incrementar la tasa de codificación. Lograron corroborar mediante simulación, que el esquema de Turbo código en UEP que propusieron comparada con el de EEP, provee mejor desempeño de BER y la tasa de codificación fue constante cuando el mensaje secreto junto con la imagen, fueron transmitidos en un canal ruidoso.

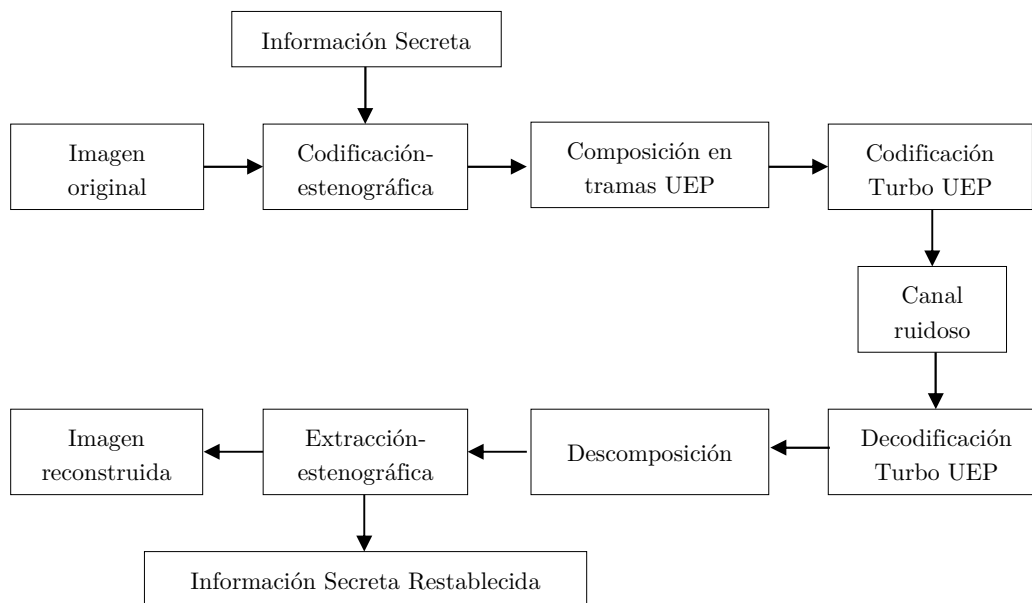


Figura 1.1: Diagrama de la comunicación estenográfica usando Turbo codificación UEP.

Por otra parte, en [8] usando UEP en imágenes con marca de agua, se propuso dividir la imagen en tres regiones: mayor protección, menor protección y sin protección; la codificación empleando UEP en la imagen, se usó en las regiones de mayor y menor protección mientras que, en la región sin protección, los bits que resultaron al salir de la codificación UEP, se incrustaron en esta región, debido a que carecía de importancia visual y sufría menos modificación.

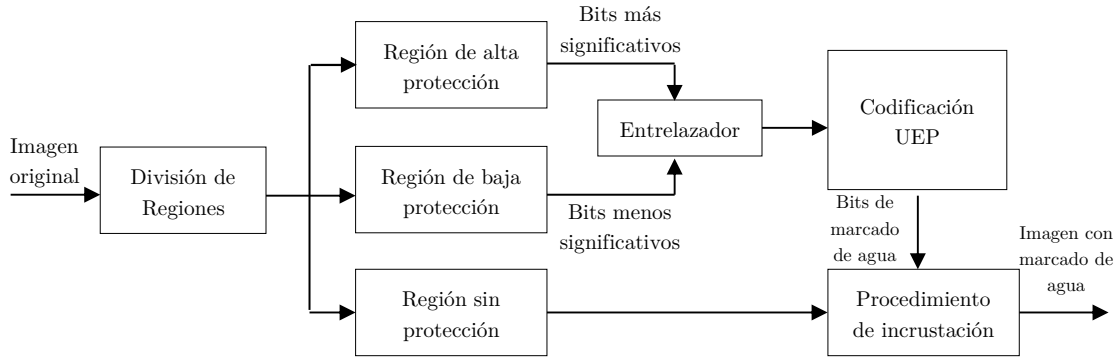


Figura 1.2: Diagrama del procedimiento de incorporación de marca de agua con UEP.

Mejoras significativas se obtuvieron en [9], en donde los autores propusieron un nuevo esquema de codificación UEP basado en códigos Doble Protograph - Control de Paridad de Baja Densidad (PD-LDPC por sus siglas en inglés) para la transmisión de imágenes médicas, el cual redujo la complejidad del sistema; se obtuvo un valor mayor de PSNR de la imagen recibida sobre un canal AWGN con baja SNR.

Un esquema de UEP fue empleado con Múltiples Turbo códigos (MFT por sus siglas en inglés) en un Canal de Línea Eléctrica (PLC por sus siglas en inglés); al proteger la información más significativa, sin disminuir el nivel de protección de los componentes menos significativos, los autores comprobaron que el rendimiento global del sistema también mejora con respecto a la codificación EEP y al uso de Turbo códigos convencionales, aplicando como parámetro la Tasa de Error de Pixel (PER por sus siglas en inglés). Como trabajo a futuro, consideraron perforar los componentes menos significativos para aumentar la tasa del código; este trabajo fue propuesto en [10].

Otra técnica de UEP es presentada en [11], donde los autores propusieron un esquema UEP de imágenes y videos sobre un canal inalámbrico; sobre una imagen en formato JPEG2000, se dividió el codestream en celdas de Región De Interés (ROI por sus siglas en inglés) y de Fondo (BG por sus siglas en inglés), conteniendo cada uno un cierto número de paquetes de paridad (véase en la Figura 1.3), utilizando codificación Reed-Solomon. El propósito fue comparar el rendimiento de su propuesta UEP contra el UEP óptimo [12], sin procesar ROI y EEP, en términos de una métrica de calidad perceptual SSIM. El enfoque propuesto UEP logró un rendimiento óptimo similar al UEP óptimo con respecto al EEP y redujo la complejidad computacional.

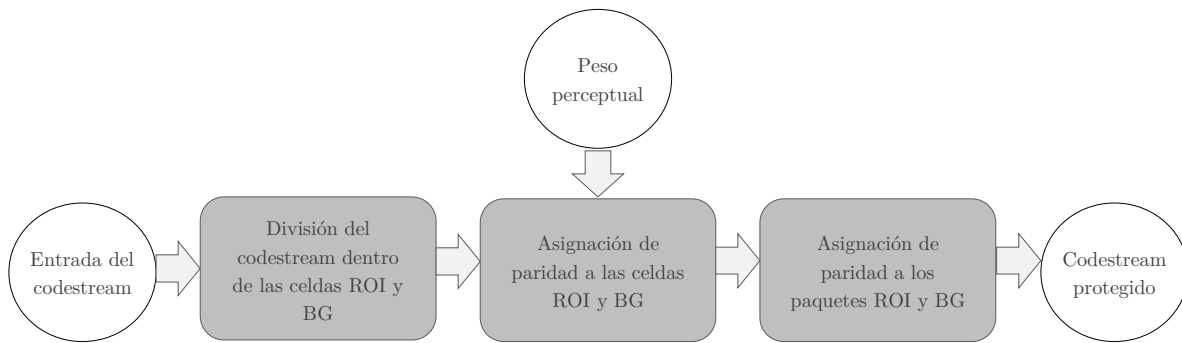


Figura 1.3: Diagrama de bloques del esquema UEP sobre ROI para imágenes y videos.

Desde un enfoque práctico, UEP se ha estudiado de manera efectiva para transmisiones inalámbricas como se ha visto en las investigaciones mencionadas anteriormente, donde se puede tolerar cierto grado de fiabilidad en la recepción. En la investigación [13] se estudió la transmisión de imágenes en canales móviles aplicando UEP con codificación convolucional perforada (PCC por sus siglas en inglés), probando con seis esquemas (dos sin codificación y 4 con codificación), dividiendo en dos los datos de la imagen, ordenándolos y aplicando UEP dependiendo de su importancia. Recientemente, se realizó un estudio [14] sobre el diseño eficaz empleando UEP con decodificadores LDPC en memorias; el método se dividió en cuatro niveles con un grado de protección diferente ajustable, cumpliendo con baja complejidad computacional y energía.

En resumen, diversos tipos de codificación se han empleado, desde códigos Reed-Solomon, convolucionales y RCPC, hasta códigos más complejos como Turbo códigos, MFT, LDPC y combinación de Turbo códigos con RCPC. Los canales de transmisión considerados para el desarrollo de las técnicas de UEP fueron representados por modelos como AWGN, PLC y Rayleigh-fading, en un ambiente simulado. Un número de parámetros de calidad perceptual y cuantitativa fueron propuestos en años recientes, por ejemplo, PER, PSNR, BER y SSIM, este último permite obtener un mejor diseño en el esquema de protección contra errores, más confiable y eficiente, considerando la percepción por el ojo humano. En las investigaciones realizadas, se comprobó que empleando la técnica UEP en la transmisión de imágenes, resultó mejor que la usada con EEP en cuanto a desempeño computacional, rendimiento en la codificación y reducción de ancho de banda.

1.2 Aplicaciones actuales basadas en la tecnología de Radio Definido por Software (SDR).

De lo antes mencionado, los trabajos relacionados con la técnica UEP se han realizado en su mayoría por medio de simulaciones; para propósito de este trabajo, se desea adecuar la estrategia de UEP sobre una imagen en un canal inalámbrico real, basándose en la tecnología SDR que permite mediante hardware y software transmitir los datos en dicho canal.

Dentro de la literatura de las comunicaciones inalámbricas, existen diversos estudios realizados sobre la tecnología SDR, con la finalidad de explorar y experimentar los distintos usos que posee para obtener resultados en los prototipos de diseño, para mejoras en este campo de las comunicaciones, los cuales se presentan en este apartado.

En cursos impartidos en la Universidad de Southampton [15], usaron la tecnología SDR mediante USRP en la plataforma de ambiente de diseño LabVIEW, con el propósito de construir un sistema básico de comunicaciones inalámbricas que les permitiera a los estudiantes comprender el reto de diseñar en un entorno real con ayuda de sus conocimientos teóricos. El USRP N210 ha sido empleado como infraestructura de la tecnología GSM en conexión con una Estación Base Transceptora Abierta (OpenBTS por sus siglas en inglés) [16]. El trabajo tuvo como objetivo diseñar una metodología que consistía en medir la calidad de voz en una infraestructura general de GSM, la cual resulto positiva porque fue de bajo costo, contemplando los componentes de hardware y software libre empleados.

En [17] los investigadores emplearon el USRP como banco de pruebas para evaluar diferentes soluciones sobre la administración del espectro radioeléctrico usando el concepto de Radio Cognitiva (CR por sus siglas en inglés), en la Figura 1.4 se observa el uso del USRP en los nodos dentro del banco de pruebas. El propósito del artículo fue proporcionar una plataforma precisa y realista en la que muestra una solución innovadora ante el problema de la administración del espectro para diversos escenarios; por ejemplo, se realizó una transmisión de video en un contexto de Hogar Digital que permitió demostrar la eficacia del banco de pruebas en tiempo real evaluando la factibilidad de la recepción en términos de calidad perceptual por el usuario en un ambiente real.

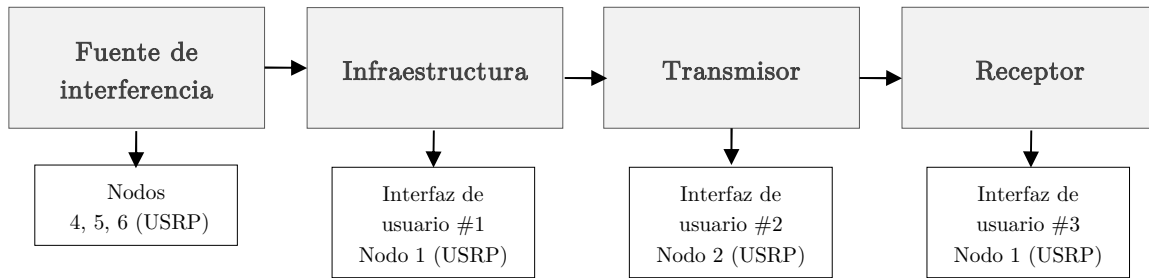


Figura 1.4: Configuración del banco de pruebas en el escenario de Hogar digital.

Otra de las aplicaciones usadas en USRP, han sido para emular un Sistema de Antenas Distribuidas (por sus siglas en inglés DAS) en [18], en el cual se usó la plataforma de LabVIEW y tres USRP 2920, dos como transmisores interconectados por cable MIMO, para comunicarse con uno solo como receptor. El objetivo principal que se presentó en este trabajo fue investigar el efecto que ocasionan los retardos sobre la red de alimentación entre las señales transmitidas desde diferentes antenas (DAS) utilizando Multiplexación por División de Frecuencias ortogonales (OFDM por sus siglas en inglés), en el que los resultados revelaron a partir de distintas configuraciones que, cuando se tienen distintas distancias en la propagación, y estas son de 30 m de separación entre las antenas (DAS) a través de la red de alimentación, pueden tener un impacto significativo en el rendimiento de la Tasa de Error de Bit (BER por sus siglas en inglés). El USRP se ha implementado también como instrumento de medición como en [19], donde los autores comparan el resultado de las mediciones con un analizador de redes vectorial, en el que aseguran que el USRP proporciona una alternativa económica para ciertas mediciones con algunas limitaciones como crosstalk, y calibración, entre otras.

Por otro lado, en [20] se realizó un trabajo que consiste en la implementación de un sistema basado en OFDM para la transmisión inalámbrica en tiempo real con codificación de canal convolucional empleando la plataforma de software libre GNU Radio y USRP N210, tal como se muestra en las Figuras 1.5 y 1.6; el objetivo que los autores plantearon fue estudiar las mejoras en los procesos de transmisión y recepción de señales entre dos tarjetas USRP, del cual obtuvieron en su análisis que hubo una mejora en la Relación de Errores de Paquete (PER por sus siglas en inglés) al introducir la codificación de canal, y al aumentar el orden de la FFT en el sistema OFDM.

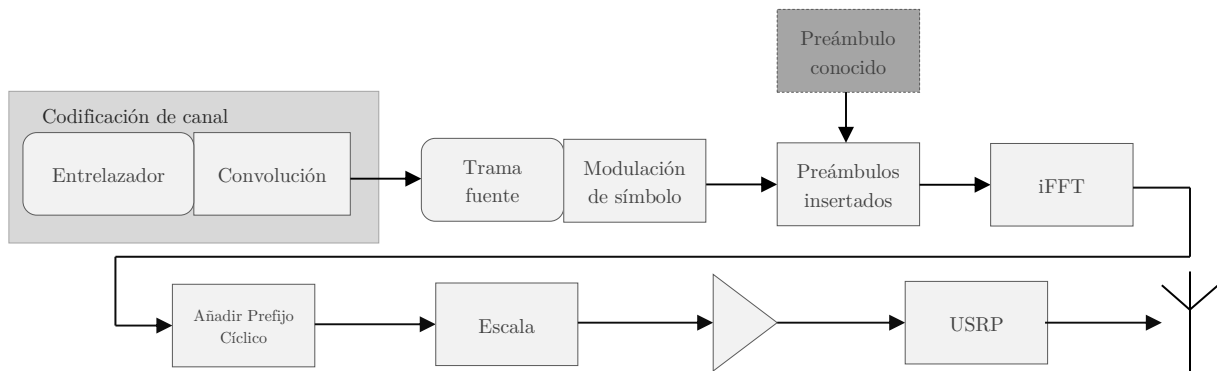


Figura 1.5: Diagrama de bloques del transmisor basado en el sistema OFDM usando codificación de canal.

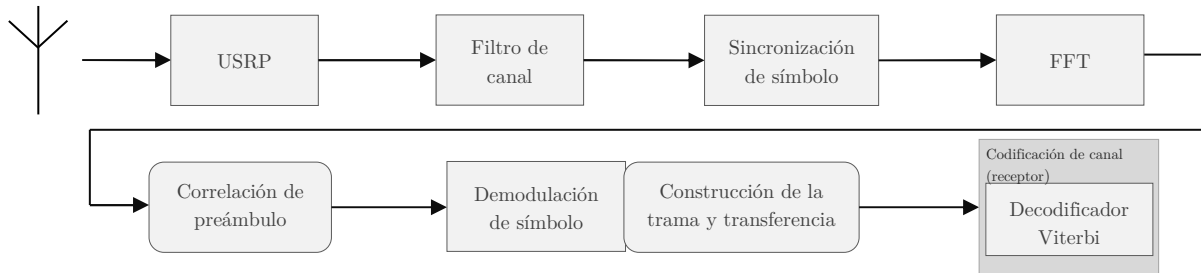


Figura 1.6: Diagrama de bloques del receptor basado en el sistema OFDM usando codificación de canal.

Recapitulando las diversas investigaciones y aplicaciones abordadas en este apartado, el uso de la tecnología SDR específicamente empleando el USRP, ha dado resultados favorables en el análisis de la búsqueda de soluciones ante diversas prácticas como por ejemplo, para propósito académico en el que gracias a la flexibilidad y alcance que poseen los equipos USRP, se pueden desprender distintos experimentos que se describen en [15], o en [16] y [19], donde fue implementado como instrumento de medición; mientras que en [17], fue una herramienta para aplicar un algoritmo general que permitiera emplear el concepto de Radio Cognitiva emulando un escenario real, y finalmente en [18] y [20], implementaron sistemas existentes puestos en marcha en la actualidad en el que los autores pretendieron probar y encontrar la causa a los obstáculos que se presentan al propagar la información en un ambiente real.

Por lo tanto, de acuerdo a las investigaciones sobre el uso de UEP y SDR en transmisiones inalámbricas, se considera en este trabajo el uso de UEP en imágenes transmitidas primeramente en un canal AWGN, posteriormente sobre un canal real con el uso de la tecnología SDR, utilizando las técnicas de detección y corrección de errores tal como la codificación de canal de bloque y convolucional combinada con RCPC para ambos canales. Contribuyendo con la metodología empleada para encontrar los resultados óptimos con respecto a los parámetros de calidad perceptual y la mejora en la complejidad computacional utilizada, haciendo un análisis del desempeño del UEP con respecto a EEP dentro del canal simulado y el canal real.

Capítulo 2

Marco Teórico

2.1 Antecedentes.

2.1.1 Orígenes de la Protección Desigual contra Errores.

El concepto UEP comienza desde el uso de la codificación de canal para el control de errores en la transmisión y recepción de información, mismo que se ha desarrollado desde hace más de medio siglo. El trabajo de Shannon en 1948 [2] denominado “Una teoría matemática de la comunicación”, postuló el segundo teorema de la codificación en un canal con ruido, que establece que los datos codificados se pueden transmitir en un canal discreto sin memoria a velocidades cerca de la capacidad del canal con una pequeña probabilidad de error. Este formalismo ha generado mucha investigación en la búsqueda de los métodos más eficientes de transmisión de datos en un entorno ruidoso, como se presentó en el capítulo anterior.

Partiendo de este principio, se propuso que los datos debían de ser protegidos todos por igual (EEP), sin embargo, en algunas aplicaciones, parte de la información debe ser mejor protegida contra errores que otra y, además, el aplicar el concepto de EEP provoca redundancia excesiva, por lo que no siempre es la mejor opción para un sistema de comunicaciones. Esto llevó al desarrollo de una solución, y un reto, en la construcción de una protección que podría introducir un mejor control de errores de manera selectiva, conocida como UEP. La primera codificación UEP fue propuesta por primera vez por Masnick y Wolf en [5].

2.1.2 Orígenes de la aplicación de Radio Definido por software.

Por otro lado, la primera implementación importante del término SDR fue en un programa militar, que requirió un radio que tuviese sus componentes de capa física dentro de un software, llamado Speakeasy de DARPA. Su objetivo principal fue tener un solo radio que pudiera proporcionar diez protocolos diferentes de radios militares y que pudieran operar entre 2 MHz y 2 GHz [21]. En 1993, Joe Mitola [22] fue el primero en publicar el término de “software radio”, asegurando que los componentes físicos de un sistema de radio pueden ser adaptados mediante software en una computadora de propósito general. Más tarde, en 1999 [23], Mitola utilizó el término “radio cognitiva”, refiriéndose a aquellos radios que poseen inteligencia necesaria para adaptarse al medio.

2.2 Modulación y modelo de canal inalámbrico.

El concepto de modulación digital consiste en hacer variar la amplitud, la frecuencia, o la fase de una portadora, dependiendo de los valores que vaya tomando la señal digital de información. La entrada de la modulación digital está formada de datos M-arios codificados. Existen tres formatos de modulación digital básicos: Modulación por Desplazamiento de Amplitud (ASK), Modulación por Desplazamiento de Frecuencia (FSK) y Modulación por Desplazamiento de Fase (PSK); estas técnicas de señalización (pasa banda), consisten en la modulación de una forma de onda portadora con una señal digital [24].

2.2.1 Modulación BPSK.

Para el esquema simulado en cómputo, en el presente trabajo se utiliza la Modulación por Desplazamiento de Fase Binario (BPSK por sus siglas en inglés), cuya función consiste en obtener dos fases de salida para una sola frecuencia portadora; una fase de salida representa un 1 lógico, y la otra un 0 lógico, entonces, cuando la señal de entrada digital cambia de estado, la fase de la portadora de salida varía entre 0 y Π radianes como se muestra en la Figura 2.1.

La expresión analítica para la señal modulada PSK es la siguiente

$$s_i(t) = \sqrt{\frac{2E}{T}} \cos\left[\omega_0 t + \frac{2\pi i}{M}\right] \quad (2.1)$$

para $i=1, \dots, M$, donde E representa la energía del símbolo, T es el tiempo de duración del símbolo y $0 \leq t \leq T$. Para la modulación binaria BPSK, los datos de la señal modulada cambian la fase de la forma de onda $s_i(t)$ con dos estados [25].

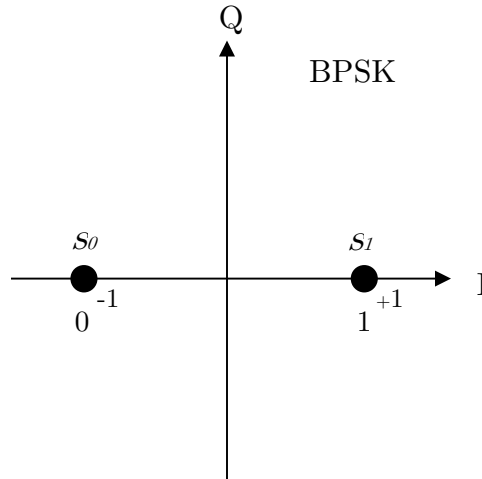


Figura 2.1: Diagrama de constelación de la modulación BPSK.

El esquema de un transmisor BPSK mostrado en la Figura 2.2, funciona como un conmutador de inversión de fase, debido a que, dependiendo de la condición lógica de la entrada digital, la potadora se transfiere a la salida ya sea en fase, o desfasada, con respecto al oscilador de la portadora [26].

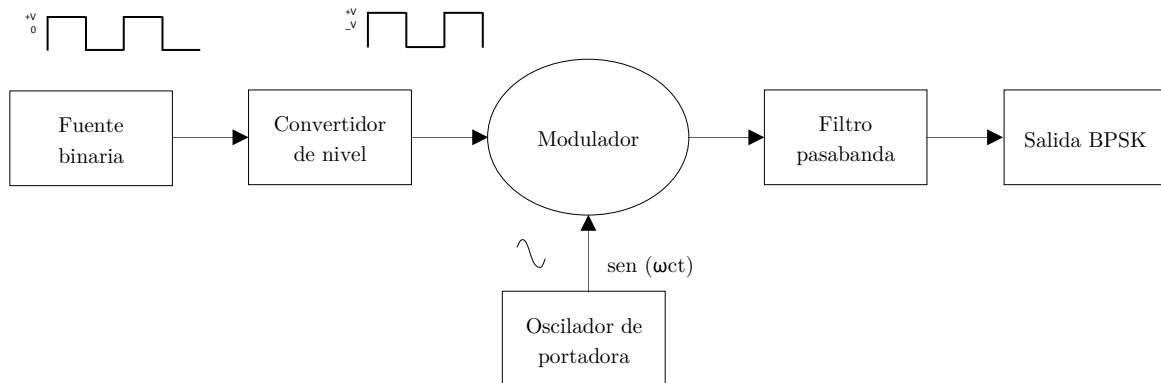


Figura 2.2: Diagrama general de un transmisor BPSK.

En la Figura 2.3 se presenta el diagrama de la forma de onda de una señal BPSK en función del tiempo, nótese los cambios de fase de la portadora cuando cambia el símbolo transmitido. La salida del modulador BPSK es una señal de doble banda lateral con portadora suprimida, en donde las frecuencias laterales superiores e inferiores están separadas de la frecuencia portadora por un valor igual a la mitad de la tasa de bits, por lo que el ancho de banda mínimo necesario para permitir el paso de la señal de salida de BPSK es igual a la frecuencia, o tasa, de los bits de entrada.

En el caso del receptor BPSK, la detección es coherente. La demodulación con detección coherente es aquella en la que se conoce la fase de la señal portadora recibida y se utiliza en la demodulación. En la Figura 2.4, se muestran los elementos básicos que conforman el receptor BPSK, del cual se desprenden los siguientes: un circuito recuperador que detecta la portadora y genera una señal portadora coherente en frecuencia y fase con la señal portadora original de transmisión. El modulador balanceado es un detector de producto entre la entrada BPSK y la portadora recuperada. El filtro pasa bajas separa los datos binarios recuperados de la señal demodulada compleja [26].

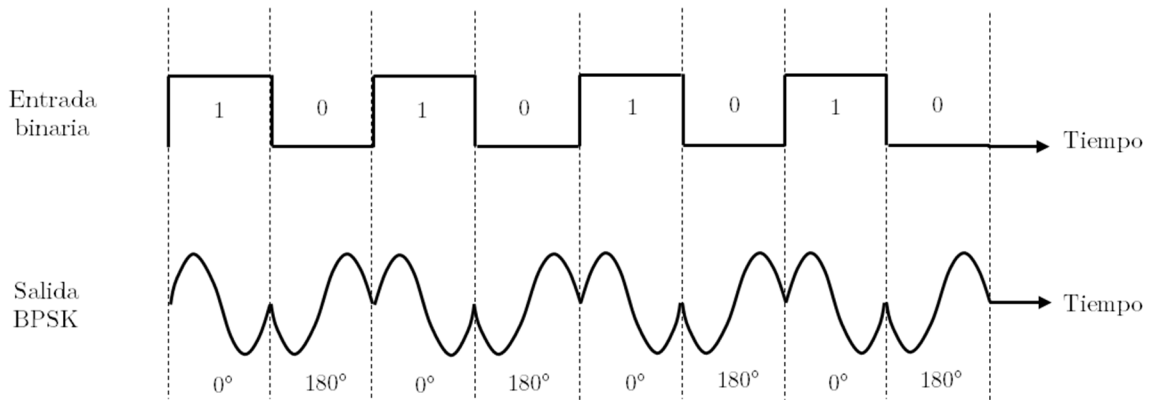


Figura 2.3: Forma de onda de la salida modulada en BPSK en función del tiempo.

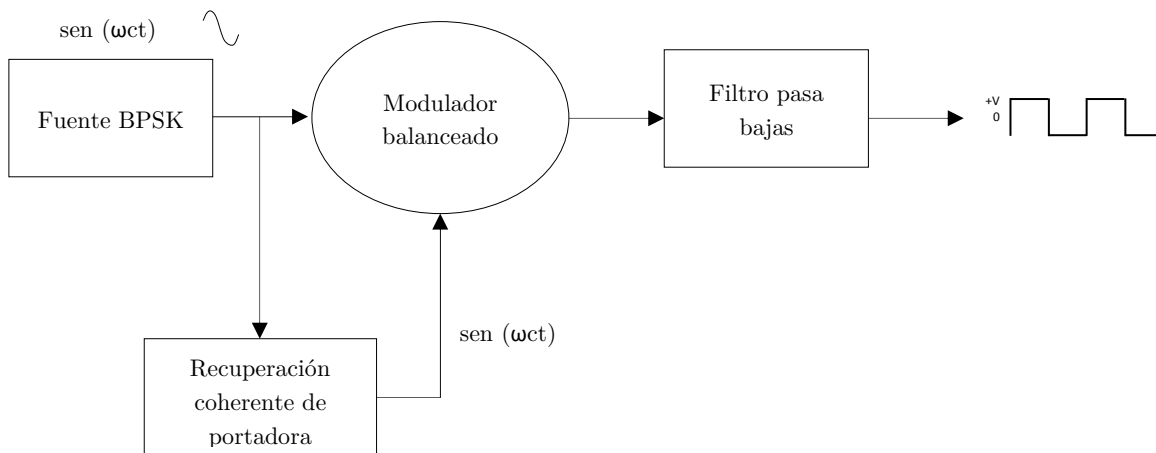


Figura 2.4: Diagrama general de un receptor BPSK.

Las ventajas que posee la modulación binaria BPSK es que es el sistema más simple. En el modulador M-ario de fase, donde $M=2^m$, y m es el número de bits por símbolo, el mayor número de fases es directamente proporcional a la cantidad de información que puede ser enviada, utilizando el mismo ancho de banda, y con eficiencia energética, debido a que la energía en todos los símbolos es la misma. Mientras que su desventaja es que la

probabilidad de bit en error P_b aumenta, a medida que se incrementa el tamaño M del alfabeto, excepto en el caso QPSK, o 4PSK, que por la manera de obtenerse, tiene la misma P_b que el BPSK.

2.2.2 Modulación GFSK.

La modulación de Desplazamiento en Frecuencia Gaussiana (GFSK por sus siglas en inglés) es un método de modulación para comunicación digital usado en estándares como Bluetooth. Este método es una variante de la modulación FSK, cuya función es transmitir información por medio de cambios de frecuencia en la onda portadora. Al aplicar un filtrado con envolvente gaussiana a la señal de onda cuadrada, se suaviza, y con ella se cambia la frecuencia de la onda portadora de manera suave, dando como resultado la modulación GFSK, con una reducción de potencia en las bandas laterales [27].

El esquema general de un transmisor GFSK es mostrado en la Figura 2.5 [26] [28], así como sus formas de onda de entrada y de salida. Se puede observar que antes de aplicar la modulación FSK, se usa el filtro gaussiano que hace que las transiciones sean más suaves y, se puedan enviar los datos transmitidos con una mejor eficiencia en el ancho de banda. Después del filtro gaussiano, se encuentra el modulador FSK, mismo que se forma con un Oscilador Controlado por Voltaje (VCO por sus siglas en inglés), cuya función es desviar la frecuencia de la portadora con respecto a la fuente binaria, esto ocurre cuando existen los cambios lógicos entre 0 (llamada frecuencia de espacio f_s) y 1 (llamada frecuencia de marca f_m).

La frecuencia de la portadora se escoge de tal modo que esté a la mitad entre las frecuencias de marca y de espacio. La desviación máxima de frecuencia de la portadora Δ_f es igual a la diferencia entre la frecuencia de la portadora y la frecuencia de marca o de espacio, puede funcionar como modulación por barrido, en el que la desviación máxima de la frecuencia es el producto del voltaje binario de entrada por la sensibilidad del VCO, como se muestra en la siguiente expresión:

$$\Delta_f = V_m(t)k_l \quad (2.2)$$

donde Δ_f = desviación máxima de frecuencia (Hertz)
 $v_m(t)$ = voltaje máximo de señal moduladora binaria (volts)
 k_l = sensibilidad a la desviación (Hertz por volt)

Para el caso del receptor GFSK, este consta de cuatro bloques como se muestra en la Figura 2.6 [29]. El mezclador se compone de multiplicadores con funciones en cuadratura: seno y coseno. El filtro pasa bajas separa los datos binarios recuperados de la señal demodulada compleja. El bloque del retardador y multiplicador utiliza la demodulación FM;

idealmente, se da salida a cuatro muestras por cada bit recibido, positivas para un “uno”, y negativas para un “cero”. Finalmente, el bloque recortador permite la salida de los bits mediante la suma de cuatro muestras de entrada, y la comprobación del signo de la suma.

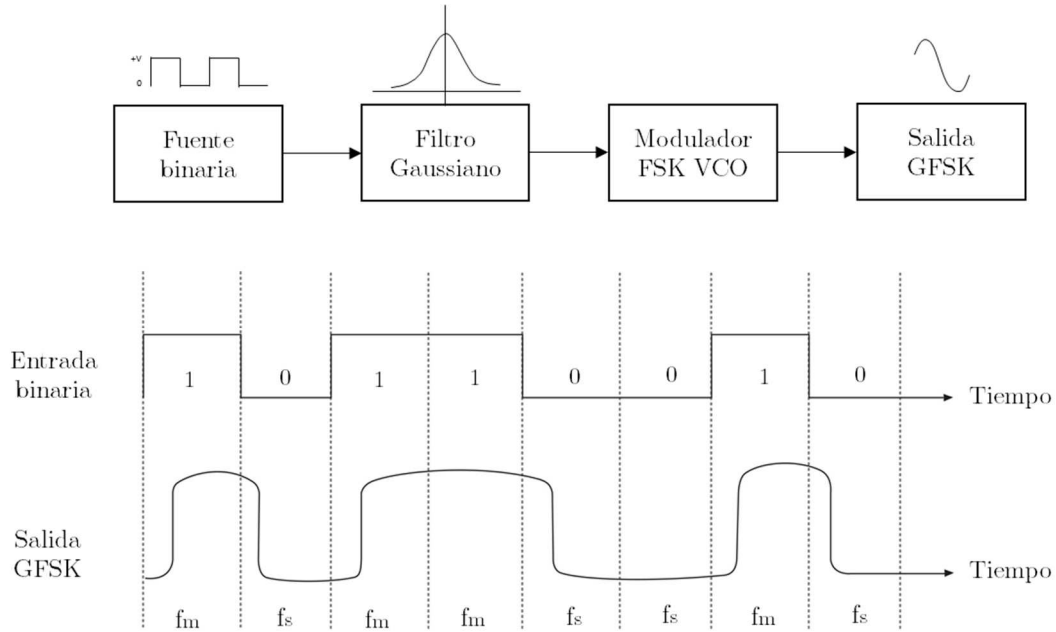


Figura 2.5: Esquema general de un transmisor GFSK y su forma de onda a la salida.

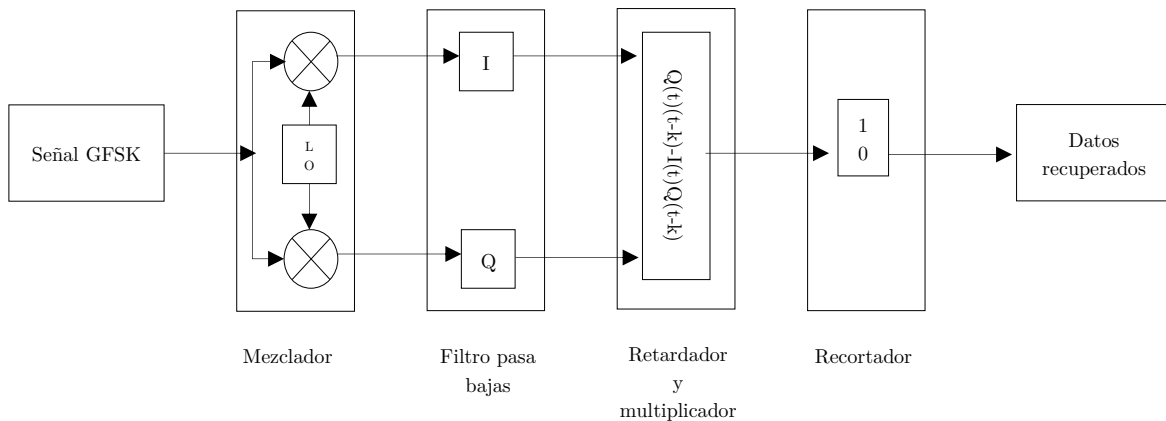


Figura 2.6: Esquema general de un receptor GFSK.

Dado que la información no se representa con cambios de amplitud, los moduladores de frecuencia son más robustos respecto al ruido. Por esta razón, en esta tesis se empleará la técnica de modulación GFSK para el sistema de comunicaciones digitales real propuesto.

2.2.3 Canal AWGN.

Para introducir el modelo de canal de comunicaciones con Ruido Aditivo Blanco, Gaussiano (AWGN por sus siglas en inglés), ampliamente utilizado, es necesario abordar primero el concepto de ruido, el cual se refiere a señales eléctricas no deseadas que siempre están presentes en los sistemas. Sin embargo, hay una fuente natural de ruido llamado ruido térmico, el cual no puede ser eliminado y, además es causado por el movimiento térmico de los electrones en todos los componentes, tales como resistores, cable, etc., por lo que los mismos electrones que permiten la conducción eléctrica, también generan el ruido térmico [25].

El ruido térmico se puede describir como un proceso aleatorio Gaussiano con media cero. La distribución Gaussiana es usada en sistemas ruidosos debido al “Teorema del límite central”, el cual considera que independientemente de la procedencia de diversas variables aleatorias, realizando la suma de las mismas, se puede obtener una variable aleatoria con una distribución Gaussiana. Otra característica del ruido térmico es que la densidad espectral de potencia es la misma para todas las frecuencias, esta propiedad del ruido se le conoce como ruido blanco. Entonces, retomando que el ruido térmico es un proceso Gaussiano, y que las muestras no están correlacionadas, las muestras del ruido también son independientes. Por lo tanto, el efecto en la detección de señales con ruido en un canal AWGN es que el ruido afecta cada símbolo transmitido independientemente; por lo que el término aditivo significa que el ruido es añadido a la señal.

Debido a que el ruido térmico está presente en todos los sistemas de comunicaciones, sus características (aditivo, Gaussiano y blanco) son las más usadas para modelar el ruido en estos sistemas [25].

En este trabajo, el canal AWGN es usado para simular por computadora el canal de ruido en el sistema de comunicaciones propuesto. El modelo del canal AWGN se crea mediante un objeto en Matlab, el cual involucra a la potencia relativa, misma que se describe por cantidades como E_b/N_0 , que es la relación entre la energía de bit y la densidad espectral de potencia.

2.3 Codificación de canal.

El objetivo de un sistema de comunicaciones es la transmisión de información sin tener pérdidas, sin embargo, en un sistema físico real las pérdidas son inevitables, disminuyendo la razón señal a ruido E_b/N_0 [dB] en el receptor. De acuerdo a un sistema general de comunicaciones digitales, se debe proceder a utilizar mecanismos de detección y protección de datos para evitar que existan altos niveles de errores en la recepción, y aplicar

parámetros de calidad sobre los datos recibidos decodificados para asegurar la fiabilidad de la información.

El mecanismo que se usará en este trabajo es llamado codificación de canal, cuyos antecedentes fueron presentados en el apartado 2.1.1 de este capítulo. La codificación de canal es un área de investigación en la que la fiabilidad del canal se incrementa al llevar a cabo la corrección de errores; se puede lograr mediante la adición de redundancia a la información que se transmite [30], lo que conduce a la generación de un vector de símbolos ya codificados. El receptor debe ser capaz de detectar y corregir los bits dañados que resultaron de haber atravesado el canal ruidoso. Como lo establece esta área de trabajo, por la forma de operar, la codificación de canal utilizada es de dos tipos: de bloque y convolucional.

Los códigos de bloque, como su nombre lo indica, dividen la información en bloques de mensajes de k bits de información. Teniendo un total de 2^k mensajes diferentes. El codificador transforma cada mensaje en un bloque codificado de n símbolos discretos, usualmente binarios, llamado palabra de código. El arreglo se conoce como código de bloque (n,k) , donde $k < n$. La relación $R=k/n$ es llamada tasa del código y se puede interpretar como el número de k bits de información que entran al codificador por cada bloque codificado transmitido de longitud n . De esta forma, $n-k$ bits de redundancia son agregados a cada mensaje de k bits para formar una palabra de código [31]. La idea fundamental es crear una distancia entre las palabras del código, la distancia mínima d entre palabras, permite calcular la capacidad t de corrección del código, donde $d \geq 2t+1$. Una primicia de la teoría de códigos, es construir un código que tenga el mayor número posible de palabras de código, separadas de la mayor distancia posible. Es decir, con la mayor capacidad de corrección posible.

En el caso de los códigos convolucionales, se opera con secuencias semi-infinitas, generalmente binarias a la entrada y a la salida del codificador. Analizando estas secuencias en el tiempo, se considera una ventana de longitud k bits a la entrada del codificador, y consecuentemente, una ventana de longitud n bits a la salida. El conjunto de secuencias codificadas producidas, utilizando un circuito codificador de memoria de orden m : (n,k,m) , es llamado código convolucional. La relación $R=k/n$, al igual que en el código de bloque, se le llama tasa de codificación, donde $k < n$.

El reto de los métodos descritos anteriormente es calcular los bits de redundancia que logren la transmisión más confiable a través del canal ruidoso, a tasas de bit hasta un valor máximo, menor que la capacidad del canal [2]. Este es el mayor problema para el diseño del codificador, no importando de qué tipo sea.

Este capítulo se trata uno de los temas principales de la tesis, la codificación de bloque BCH y la codificación convolucional. Por esta razón, es importante tener claro los conceptos de teoría de códigos, previo a la aplicación en el sistema de comunicaciones empleado.

2.3.1 Códigos BCH.

En los sistemas de comunicaciones digitales actuales, la familia de códigos correctores de errores Bose-Chaudhuri-Hocquenghem (BCH) es de las más utilizadas, y consiste en una generalización de los primeros códigos definidos, llamados de Hamming. Permiten combatir t múltiples errores en el sistema, y fueron descubiertos por Hocquenghem en 1959, e independientemente por Bose y Chaudhuri en 1960. Esta clase de códigos de bloque, tienen la propiedad de ser cíclicos, es decir, al hacer corrimientos cíclicos en una palabra de código, se generan otras palabras del mismo código. Se definen por un polinomio generador $g(x)$ de grado $n-k$, y constituyen una gran clase de códigos de fácil construcción, con múltiples longitudes de bloques n y tasas de código $R=k/n$. Para los códigos BCH binarios, n es siempre igual a una potencia de dos menos uno ($n=2^m-1$). Cuanto mayor sea la longitud del bloque n , existen mayores combinaciones de códigos (n,k) . Sin embargo, las tasas de código óptimas para códigos BCH son entre $1/3$ y $3/4$, otras tasas podrían mejorar poco el rendimiento, o incluso disminuirlo [32].

2.3.1.1 Codificador BCH.

Para cualquier m entera positiva ($m \geq 3$) y t ($t < 2^{m-1}$), existe un código binario BCH con los siguientes parámetros:

Longitud del bloque: $n = 2^m - 1$
 Grado de $g(x)$: grado $g(x) = n - k$
 Distancia mínima: $d_{\min} \geq 2t + 1$

El polinomio generador de este código es definido en términos de sus raíces en un campo finito de Galois de orden 2^m , $GF(2^m)$, siendo α un elemento primitivo en $GF(2^m)$. El polinomio generador $g(x)$, corrector de t errores del código BCH de longitud $2^m - 1$, es el polinomio de menor grado sobre $GF(2^m)$, con:

$$\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t} \quad (2.3)$$

como sus raíces.

El cálculo del polinomio generador es el más pequeño común múltiplo de los polinomios mínimos $m_i(x)$, tiene la siguiente definición:

$$g(x) = \text{MPCM}\{m_i(x)\}, \quad (2.4)$$

donde para un código BCH binario, i toma valores impares: $i = \{1, 3, (2t-1)\}$, y $m_i(x)$ es el polinomio mínimo de α^i , es decir, tiene por raíces α^i y todos los conjugados de α^i , en GF (2^m). Para los códigos BCH binarios, los conjugados de α^i se obtienen a partir de $\alpha^{i \cdot 2^j}$, dando los valores al exponente $j = \{0, 1, 2, \dots\}$, hasta que se obtiene la igualdad:

$$\alpha^i = \alpha^{i \cdot 2^j} \quad (2.5)$$

Esta igualdad se alcanza sobre campos finitos, dado que el número de conjugados de α^i es también finito. Se tiene entonces

$$m_i(x) = \prod_j (x - \alpha^{i \cdot 2^j}) \quad (2.6)$$

Considerando una de las propiedades de los códigos cíclicos, que establece que toda palabra de código $c(x)$ es un múltiplo del polinomio generador: $c(x) = K(x) \cdot g(x) = x^{n-k}i(x) + r(x)$, el codificador de un código BCH se construye utilizando un Registro de Desplazamiento con Retroalimentación Lineal (LFSR por sus siglas en inglés), para llevar a cabo una división polinomial, que permite calcular la redundancia $r(x)$ de la palabra del código, dada por:

$$c(x) = x^{n-k}i(x) + r(x) \quad (2.7)$$

La palabra de código es: $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$
 Los bits de información: $i(x) = i_0 + i_1x + \dots + i_{k-1}x^{k-1}$
 Residuo de $x^{n-k}i(x)/g(x)$: $r(x) = r_0 + r_1x + \dots + r_{m-1}x^{m-1}$

Para un código BCH binario, los coeficientes, c_i, i_j, r_j son binarios. El circuito de la Figura 2.7 permite el cálculo de la redundancia $r(x)$, como el residuo de la división $x^{n-k}i(x)/g(x)$, es decir, calcula la operación $x^{n-k}i(x)$ módulo $g(x)$.

La operación del codificador LFSR BCH de la Figura 2.7 es la siguiente:

1. Para el ciclo de reloj de 1 a k , los bits del mensaje original son transmitidos sin cambio, es decir, los bits de información formarán parte de la palabra codificada (codificación sistemática). Durante esta operación, el switch S2 está en la posición 2, y el registro de desplazamiento con retroalimentación lineal calcula los bits de redundancia (el switch S1 está ahora activado).

2. Para el ciclo de $k+1$ a n , los bits de redundancia en el LFSR son transmitidos (el interruptor S_2 está en la posición 1) y la retroalimentación en el LFSR está con el switch S_1 desactivado.

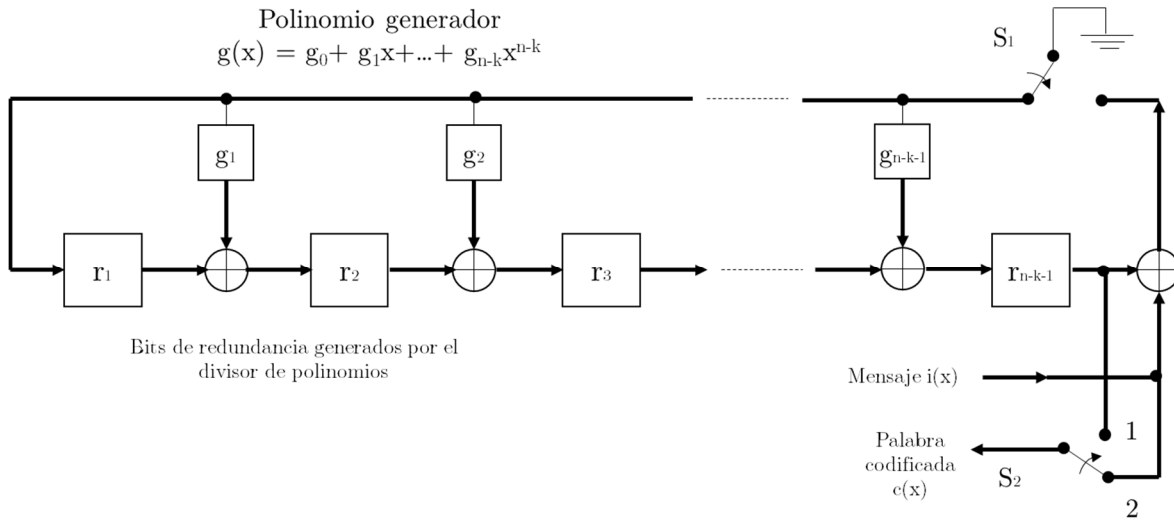


Figura 2.7: Arquitectura general del codificador BCH.

En resumen, la codificación BCH se basa en el siguiente algoritmo:

- a) Seleccionar un polinomio primitivo de grado m .
- b) Construir el campo finito de Galois $GF(2^m)$.
- c) Calcular los polinomios mínimos $m_i(x)$ a partir de α^i , para $i = \{1, 3, (2t-1)\}$.
- d) Obtener el polinomio generador $g(x)$ como el más pequeño común múltiplo de $m_i(x)$.
- e) Encontrar los bits de redundancia mediante el residuo de la división $r(x) = x^{n-k}i(x)/g(x)$, donde $x^{n-k}i(x)$ son los bits de información, ocupando los bits más significativos de la palabra del código $c(x)$, y $g(x)$ es el polinomio generador, quedando la palabra codificada como se observa en la Figura 2.8.

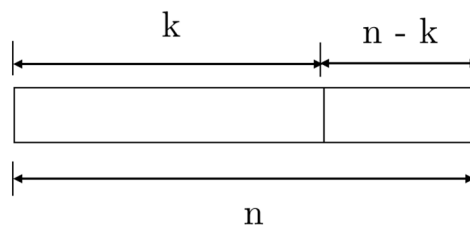


Figura 2.8: Palabra codificada resultante de la codificación sistemática BCH.

2.3.1.2 Decodificador BCH.

De manera esquemática, en la Figura 2.9 [30] se muestra el diagrama de bloques del decodificador BCH, el cual consiste de circuitos digitales que realizan las siguientes etapas:

- Con base a la propiedad de los códigos cíclicos, de que toda palabra de código es un múltiplo del polinomio generador, por tanto, las raíces de $g(x)$ son también raíces de $c(x)$. Calcular los síndromes S_j con la evaluación en las raíces del polinomio $g(x)$, del polinomio recibido $R(x) = c(x) + e(x)$.
- Utilizando el algoritmo de Berlekamp, calcular los coeficientes del polinomio localizador de errores $\sigma(x)$.
- Con la búsqueda de Chien, calcular las raíces inversas de $\sigma(x)$, para localizar los errores $\alpha^{j_1}, \dots, \alpha^{j_v}$.
- Corregir la palabra recibida en posiciones de error calculadas.

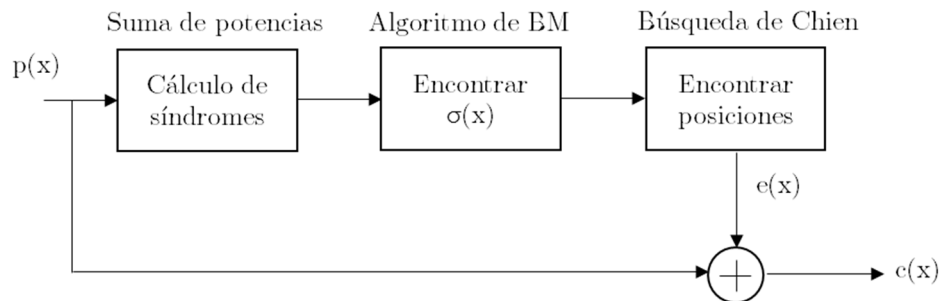


Figura 2.9: Arquitectura de un decodificador BCH con aritmética $GF(2^m)$.

2.3.1.2.1 Cálculo de síndromes [30] [33]

Haciendo operaciones en el campo finito de Galois $GF(2^m)$, las posiciones de los errores pueden ser encontrados resolviendo una serie de ecuaciones. Estas ecuaciones son obtenidas a partir del polinomio de error $\bar{e}(x)$ y las raíces del polinomio generador α^j , para $b \leq j \leq b+2t-1$, donde $b = \{1, 3, (2t-1)\}$, con el siguiente desarrollo:

Se define el polinomio recibido $p(x)=c(x)+\bar{e}(x)$, donde $c(x)$ que representa el polinomio asociado a la palabra de código, y el polinomio de error es:

$$\bar{e}(x) = e_{j_1}x^{j_1} + e_{j_2}x^{j_2} + \dots + e_{j_v}x^{j_v} \quad (2.8)$$

donde $v \leq t$ es el número de errores. Los conjuntos $\{e_{j_1}, e_{j_2}, \dots, e_{j_v}\}$ y $\{\alpha^{j_1}, \alpha^{j_2}, \dots, \alpha^{j_v}\}$ son definidos como el valor del error y la posición del error respectivamente, donde $e_j \in \{0,1\}$ para códigos BCH binarios, y $\alpha^{j_v} \in GF(2^m)$.

Recordando que toda palabra de código es un múltiplo del polinomio generador $g(x)$, los síndromes S_j , se obtienen a partir de la evaluación del polinomio recibido $p(x)$ en cada una de las raíces del polinomio $g(x)$.

$$\begin{aligned} S_1 &= p(\alpha^b) = e_{j1}\alpha^{bj1} + e_{j2}\alpha^{bj2} + \dots + e_{jv}\alpha^{bjv} \\ S_3 &= p(\alpha^{b+1}) = e_{j1}\alpha^{(b+1)j1} + e_{j2}\alpha^{(b+1)j2} + \dots + e_{jv}\alpha^{(b+1)jv} \\ &\vdots \\ S_j &= p(\alpha^{b+2t-1}) = e_{j1}\alpha^{(b+2t-1)j1} + e_{j2}\alpha^{(b+2t-1)j2} + \dots + e_{jv}\alpha^{(b+2t-1)jv} \end{aligned}$$

Para reducir el número de cálculos computacionales en los códigos binarios BCH, se tiene que:

$$(S_j)^2 = S_{2j} \quad (2.9)$$

2.3.1.2.2 Cálculo del polinomio localizador de errores. [30] [33]

El polinomio localizador de errores está definido como

$$\sigma(x) \triangleq \prod_{l=1}^v (1 + \alpha^{jl}x) = 1 + \sigma_1x + \sigma_2x^2 + \dots + \sigma_vx^v \quad (2.10)$$

Con raíces iguales a las inversas de las posiciones de los errores. A continuación, se tiene el siguiente sistema de ecuaciones, entre los coeficientes de $\sigma(x)$ y los síndromes

$$\begin{pmatrix} S_{v+1} \\ S_{v+2} \\ \vdots \\ S_{2v} \end{pmatrix} = \begin{pmatrix} S_1 & S_2 & \dots & S_v \\ S_2 & S_3 & \dots & S_{v+1} \\ \vdots & \vdots & \ddots & \vdots \\ S_v & S_{v+1} & S_{2v-1} & \end{pmatrix} \begin{pmatrix} \sigma_v \\ \sigma_{v-1} \\ \vdots \\ \sigma_1 \end{pmatrix} \quad (2.11)$$

Resolviendo este sistema (2.11), que resulta ser la operación más demandante computacionalmente de la decodificación BCH, se obtienen las posiciones de los errores. En este apartado, se describirá el algoritmo para resolver este sistema (2.11) utilizando el algoritmo de Berlekamp-Massey en la plataforma de Matlab¹.

¹ <http://www.mathworks.com/help/comm/ref/bchdec.html#fp1448240>

Algoritmo de Berlekamp-Massey.

Este algoritmo fue propuesto por Berlekamp y Massey en 1969. Es un método iterativo, computacionalmente eficiente en términos del número de operaciones sobre GF (2^m).

La meta del algoritmo es encontrar una solución polinomial $\sigma^{(i+1)}(x)$ de grado mínimo que satisfaga el siguiente sistema, derivado de la ecuación (2.11)

$$\sum_{j=0}^{l_i} S_k - j\sigma_j^{(i+1)} = 0, \quad l_i < k < i + 1 \quad (2.12)$$

En el algoritmo, se define la discrepancia en la iteración i como

$$d_i = S_{i+1} + S_{i+1}\sigma_1^{(i)} + \dots + S_{i-l_i+1}\sigma_{l_i}^{(i)} \quad (2.13)$$

esta discrepancia sirve para determinar el valor del polinomio localizador de errores $\sigma(x)$ de la siguiente iteración. En esta etapa, se tienen dos casos a considerar:

- a) Si la discrepancia $d_i = 0$, entonces la ecuación (2.12) se cumple para:

$$\sigma^{(i+1)}(x) = \sigma^i(x), \quad l_{i+1} = l_i \quad (2.14)$$

- b) Si la discrepancia $d_i \neq 0$, hacemos que $\sigma^{(m)}(x)$ sea la solución de la iteración m , tal que $-1 \leq m < i$, $d_m \neq 0$, y $(m - l_m)$ sea máxima. Entonces

$$\begin{aligned} \sigma^{(i+1)}(x) &= \sigma^i(x) + d_i d_m^{-1} x^{(i-m)} \sigma^{(m)}(x) \\ l_{i+1} &= \max\{l_i, l_m + i - m\} \end{aligned} \quad (2.15)$$

Con valores iniciales dados y el contador de iteraciones $i=0$, el cálculo de $\sigma^{(i+1)}(x)$ continua hasta cumplir con las condiciones $i \geq l_{i+1} + t - 1$, o $i = 2t - 1$.

Los valores iniciales de este algoritmo están dados por:

$$\begin{aligned} \sigma^{(-1)}(x) &= 1, \quad l_{-1} = 0, \quad d_{-1} = 1, \\ \sigma^{(0)}(x) &= 1, \quad l_0 = 0, \quad d_0 = S_1 \end{aligned} \quad (2.16)$$

2.3.1.2.3 Búsqueda de Chien y corrección de errores. [33]

Una vez obtenido el polinomio localizador de errores con el algoritmo de Berlekamp-Massey, para encontrar las posiciones de los errores en la palabra recibida, es utilizado el método de la búsqueda de Chien, el cual consiste en evaluar el polinomio localizador de errores $\sigma(x)$,

para $x = \alpha^{-i}$, donde $i=0, 1, \dots, n-1$. Cuando $\sigma(\alpha^{-i})=0$, entonces i representa la posición del error en la palabra recibida de longitud n , por lo que se obtiene el polinomio de error $e(x)$.

2.3.1.2.4 Corrección de la palabra recibida. [33]

Dado que el código es binario, el decodificador suma el patrón del error y la palabra recibida, obteniendo la palabra corregida $c(x)$ es decir:

$$c(x) = p(x) + e(x) \quad (2.17)$$

Recordando que $p(x)=c(x)+\bar{e}(x)$, sustituyendo $p(x)$ en la ecuación (2.17), se tiene que $c(x)=c(x)+\bar{e}(x)+e(x) = c(x)$, en aritmética módulo 2.

2.3.2 Códigos Convolucionales.

La otra gran familia de códigos correctores de errores que se utiliza en la práctica, son los códigos convoluciones, definidos por Elias en 1955 como una alternativa para los códigos de bloque. En 1967, Viterbi propuso un esquema de decodificación de máxima verosimilitud que fue relativamente fácil de implementar para códigos con bajos órdenes de memoria; el algoritmo asociado es llamado decodificación de Viterbi [31].

A diferencia de los códigos de bloque, los códigos convolucionales trabajan con secuencias generalmente binarias de longitudes de algunos miles de bits. Las tasas de codificación que se utilizan son del orden de $R=1/2$, o $R=1/3$. Estas tasas definen lo que se conoce con el nombre de “código madre”, a partir del cual por un proceso de perforación (punctured) se pueden obtener otras tasas, por ejemplo: $R=5/6$, $R=7/8$, etc. Como se verá más adelante, el circuito del codificador es una máquina de estados finitos, por lo que el código se puede estudiar a lo largo del tiempo como un conjunto de transiciones entre estados, definiendo lo que se conoce como estructura de enrejado, o “trellis”, asociada al código. Una secuencia codificada será entonces una trayectoria a lo largo del enrejado. Cuando hay ruido en la comunicación, esa trayectoria se modifica en algunas partes, generando la trayectoria recibida. El algoritmo de decodificación de Viterbi, al ser óptimo, calcula la trayectoria correcta a partir de la trayectoria recibida, utilizando un criterio de máxima verosimilitud (maximum likelihood).

2.3.2.1 Codificador convolucional.

Hemos dicho que un codificador convolucional es una máquina de estados finitos, su realización requiere de un subconjunto de registros de corrimiento y de sumadores módulo 2

(compuertas or-exclusivas XOR), que permiten definir un conjunto de polinomios generadores del código, tantos como la tasa del código requiera. Por ejemplo, un codificador de tasa $R=1/2$, necesita dos polinomios, un codificador de tasa $R=1/3$, necesita 3 polinomios, etc. La secuencia de información a la entrada, se desplaza a través de los registros de corrimiento, y con los sumadores módulo 2 se calcula la secuencia codificada de salida. Con esta estructura, el circuito codificador tiene memoria, por lo que la salida codificada no solo depende de la entrada instantánea, sino también de los valores anteriores [30]. Así, existen tres parámetros que definen un código convolucional: i) el número de bits de entrada b , en cada ciclo de codificación, ii) el número de bits de salida n , en cada ciclo de codificación y, iii) la longitud de restricción (K), que tiene que ver con el número de etapas en el registro de corrimiento del codificador. Con estos parámetros, un código convolucional se representa como (K, R) donde $R= b/n$ es la tasa del código. Para ilustrar, en la Figura 2.10 se muestra la estructura general de un codificador convolucional, conformado con bK -etapas en el registro de desplazamiento, y n sumadores módulo 2, siendo K la longitud de restricción [25].

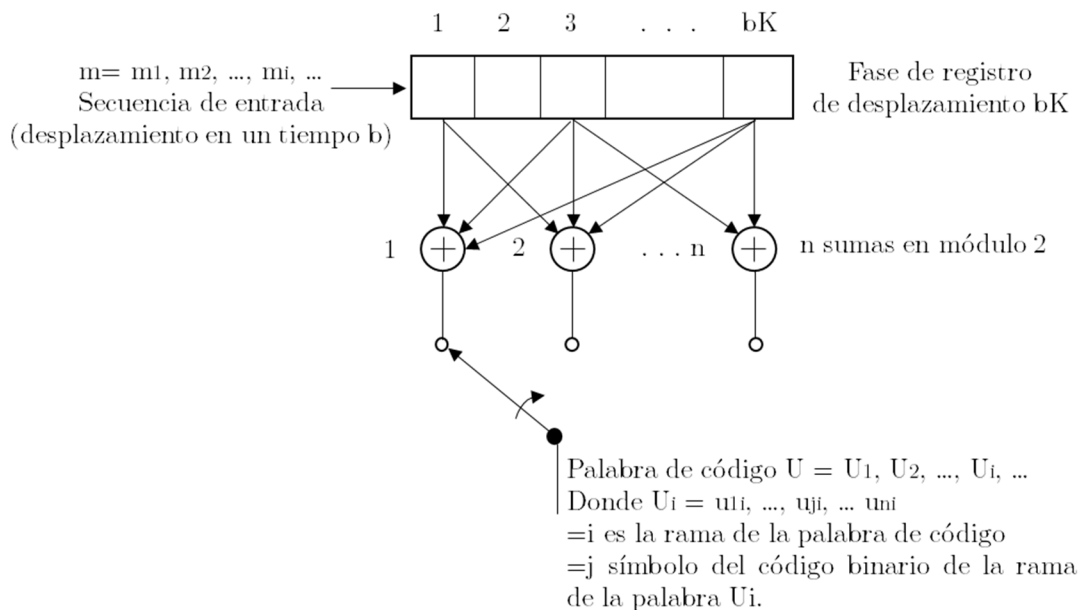


Figura 2.10: Codificador convolucional con longitud de restricción K y tasa $R=b/n$.

La longitud de restricción representa el número de b -bits desplazados en el registro de corrimiento, que pueden influir a la salida del codificador. En cada unidad de tiempo, los b -bits son desplazados dentro de las primeras posiciones del registro; todos los bits en el registro, son desplazados en b posiciones hacia la derecha, y las salidas de las n sumas son secuencialmente muestreadas para obtener la secuencia binaria codificada [25]. Las n conexiones del codificador convolucional pueden ser definidas por polinomios generadores $g_i(x)$, $i \in [1, n]$ dando un conjunto de n polinomios generadores, uno para cada una de las n sumas de módulo 2. A la suma, cada polinomio es de grado $K-1$, y determina la conexión

del registro de desplazamiento del codificador [25]. Los códigos convolucionales no sistemáticos, es decir, aquellos donde la secuencia de entrada no forma parte directa de la salida, son los más utilizados, porque tienen un mejor desempeño. Sin embargo, pueden tener la propiedad de ser catastróficos, misma que se debe evitar, porque genera más errores de los que se producen en la comunicación. Una condición necesaria y suficiente para evitar un código catastrófico, es que los polinomios $g_i(x)$ no tengan factores, de grado al menos uno, en común, haciendo la factorización en aritmética módulo 2 [25]. En la Figura 2.11 se muestra un ejemplo simple, pero de uso frecuente, de un codificador convolucional con parámetros $K=3$, $R=1/2$ y polinomios $g_1(x) = 1 + x + x^2$, $g_2(x) = 1 + x^2$, en el que hay K registros de desplazamiento etiquetados como D. Dado que el código es de tasa $R=1/2$, la secuencia codificada se obtiene como resultado sucesivo de los dos sumadores módulo 2, por cada bit de entrada.

Un codificador convolucional también puede describirse como: una máquina de estados finitos, un diagrama de árbol, o un diagrama de enrejado “trellis”. Dado que el codificador convolucional tiene una memoria de tamaño $b(K-1)$, su comportamiento a lo largo del tiempo está descrito por una máquina de estados, donde se tienen $2^{b(K-1)}$ estados en el diagrama, dependiente de la longitud de restricción K , y del número de bits de entrada b , en cada ciclo de codificación.

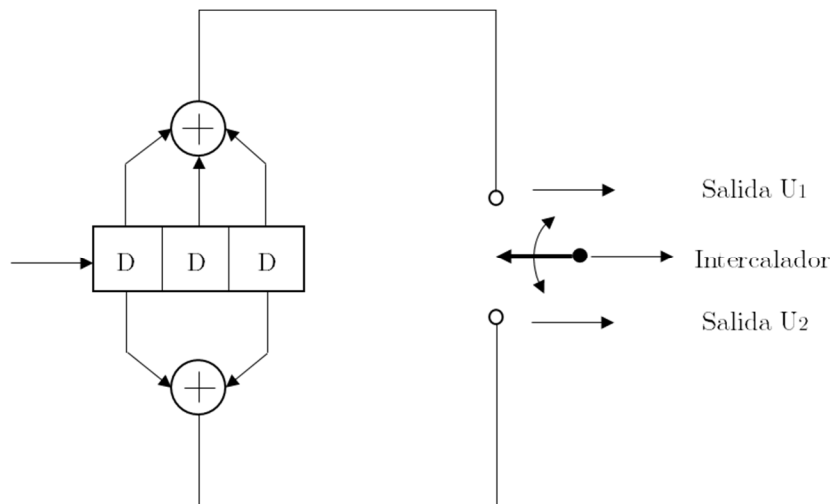


Figura 2.11: Codificador convolucional con longitud de restricción $K=3$ y tasa $R=1/2$.

El codificador de la Figura 2.11, tiene el **diagrama de estados**, dado en la Figura 2.12, en el que hay cuatro estados, cada estado está representado por el contenido de los $(K-1)$ bits más recientes del registro de desplazamiento, y cada transición de estados es designada por los bits de entrada y de salida del codificador, respectivamente. Obsérvese que no es posible que en una sola transición se pueda mover de un estado dado, a otro, de manera arbitraria, ya que para el código de tasa $R=1/2$, solo es posible desplazar un bit a la vez, habiendo solamente dos posibles transiciones por estado, por cada unidad de bit. Por ejemplo, si el

estado actual del codificador es 00, las únicas posibilidades para desplazarse al siguiente estado son: 00, si la entrada es 0, o al estado 10, si la entrada es 1, respectivamente [25].

El diagrama de estados también se puede utilizar para determinar si un codificador convolucional es catastrófico: un código convolucional es catastrófico, sí y solo sí, en el diagrama de estados existe un lazo cerrado que tenga peso nulo, para una secuencia de entrada diferente de cero [30].

Además del diagrama de estados de un código convolucional, la otra representación más utilizada es el diagrama de enrejado, o “trellis”. Esta consiste en un grabado a lo largo del tiempo de las transiciones entre estados. En la Figura 2.13 [25] se muestra el diagrama de trellis del codificador de la Figura 2.11. Las sucesiones de nodos del diagrama de trellis caracterizan los estados del codificador; así, los nodos del primer renglón en la Figura 2.13, corresponden al estado 00, los del segundo renglón al estado 10, los del tercero al estado 01, y los del cuarto al estado 11. En cada unidad de tiempo, el diagrama de trellis requiere 2^{K-1} nodos para representar los estados del codificador, que en el ejemplo son 4. En la dirección horizontal se presentan las salidas para distintas entradas. Cuyas representaciones son: con líneas continuas (para entradas 0), y líneas discontinuas (para entradas 1), sobre las flechas aparecen los valores de salida correspondiente para esa entrada [25].

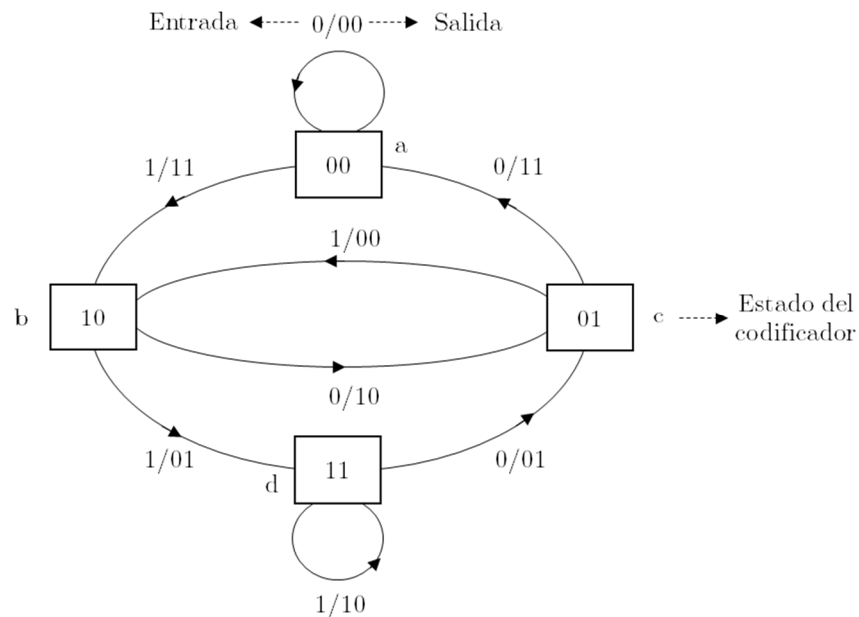


Figura 2.12: Codificador convolucional con longitud de restricción $K=3$ y tasa $R=1/2$.

A manera de ejemplo, la línea marcada con trazo grueso en la Figura 2.13 es el resultado de las transiciones realizadas por el codificador para una secuencia de entrada de valor 1010.

Al comienzo de la codificación en $t=1$, el codificador está en el estado 00, y pasa al estado 10 debido a que la entrada es un 1, con salida de 11. Para $t=2$, el codificador pasa al estado 01 debido a que la entrada es un 0, con salida 10. En $t=3$, el codificador pasa al estado 10 debido a que la entrada es un 1, con salida 00. Para $t=4$, el codificador pasa al estado 01 debido a que la entrada es un 0, con salida 10. Para finalizar la secuencia, el codificador inserta otro bit para regresar al estado 00, con salida 11. Por lo tanto, el flujo de bits de la salida codificada es **11, 10, 00, 10, 11**.

Otro parámetro muy importante que poseen los códigos convolucionales es la **distancia libre** d_f cuya definición es la distancia más corta entre dos distintas secuencias de código. El tamaño de las secuencias tiene que ser suficientemente grande, inclusive mucho más grande que la longitud de restricción del código [30]. Entonces, la distancia libre es el peso mínimo de la palabra de código de cualquier longitud, producida por la secuencia de información cuyo primer bit es diferente de cero, y todos los demás son nulos. También es el peso mínimo de todas las trayectorias en el diagrama de estados, que divergen, y luego convergen, en el estado cero [31]. Para el ejemplo de la Figura 2.13 que hemos venido desarrollando, la distancia libre es de 5.

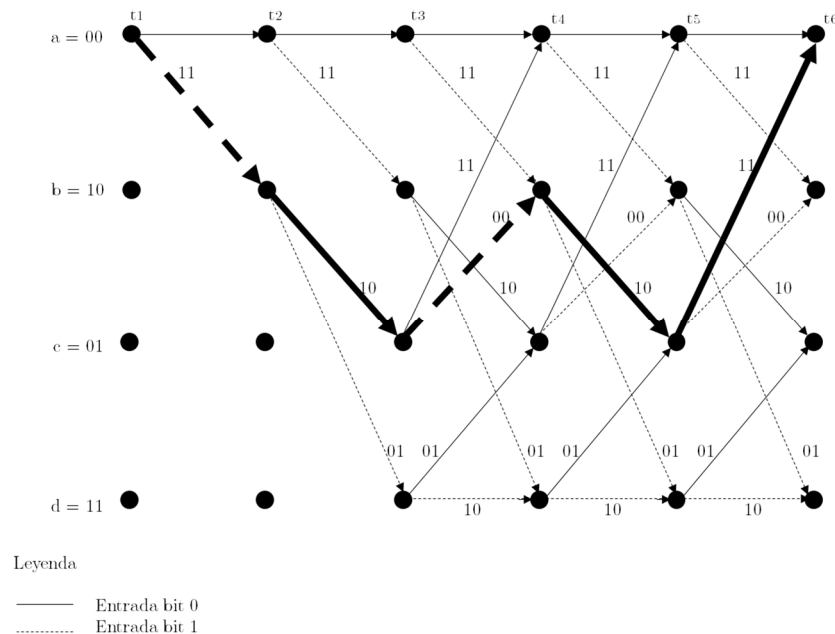


Figura 2.13: Diagrama de trellis con longitud de restricción $K=3$ y tasa $R=1/2$ para una entrada 1010.

2.3.2.2 Códigos convolucionales perforados.

Los códigos convolucionales más usados tienen tasas de $R=1/2$ y $R=1/3$, estos poseen un enrejado con (2^1) trayectorias de entrada en cada nodo, lo que requiere una comparación entre dos caminos que llegan a un mismo nodo para escoger el mejor bajo el “criterio de

máxima verosimilitud”, durante el proceso de cálculo del algoritmo de decodificación de Viterbi. En términos computacionales, esta comparación es relativamente rápida, lo cual justifica el uso frecuente de esas tasas. Sin embargo, dada una aplicación, no siempre se requieren esas tasas, pero el costo computacional aumenta a medida que más bits entran de manera instantánea para su codificación. Por ejemplo, un código de tasa $R=5/6$ posee un enrejado con (2^5) caminos de entrada en cada nodo, para escoger el mejor, la decisión se vuelve prohibitiva computacionalmente, porque hay que escoger entre 32 posibilidades por nodo. Entonces, la complejidad computacional es un problema para la aplicación del decodificador de Viterbi para tasas altas de códigos convolucionales. Este problema se puede reducir mediante el uso de la perforación en los códigos convolucionales. El concepto de los códigos convolucionales perforados fue introducido por Cain, Clark y Geist [34], en el que propusieron desarrollar la decodificación de Viterbi para una tasa de codificación de $R=K/N$, utilizando solamente dos ramas que llegan a cada nodo, en vez de 2^k ramas. Con esto se pueden obtener códigos de tasas diferente: $2/3$, $3/4$, etc., a partir de un código, llamado “código madre”, de tasa de $1/2$. En el proceso de perforado, algunos bits de salida del “código madre” de tasa baja se cancelan, de ahí el nombre de perforación. Esto permite que la estructura del trellis del “código madre” de tasa baja no cambia, aunque se cambie la tasa final $R=K/N$ con la perforación. La justificación es que se sigue manteniendo el mismo número de bits de información que se codifican en cada ciclo [30], pero la tasa final cambia debido a la perforación. En general, el modelo de la perforación es representado por una matriz llamada de perforación P de la forma

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1p} \\ P_{21} & P_{22} & \cdots & P_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{np} \end{bmatrix} \quad (2.18)$$

La matriz de perforación tiene n renglones, uno para cada flujo de salida en un codificador convolucional con n bits de salida. El número de columnas en la matriz de perforación P es el número de bits sobre el cual el modelo de perforación se repite. Los valores de la matriz son binarios P_{ij} , indican si el bit de salida correspondiente se transmite ($P_{ij} = 1$) o no ($P_{ij} = 0$). En la Figura 2.14 se pueden observar los bloques que conforman la codificación convolucional basada en la perforación del código [30].

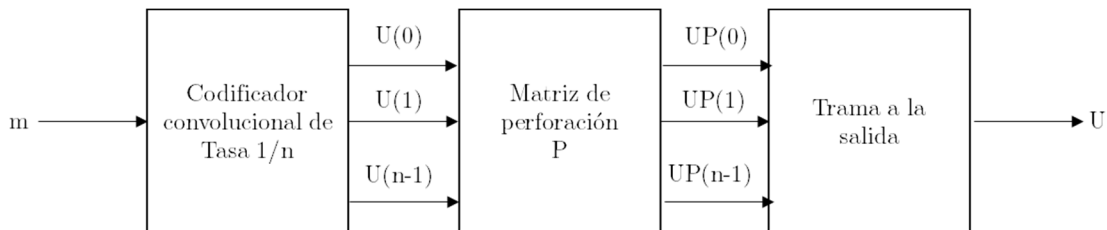


Figura 2.14: Diagrama del codificador convolucional de tasa de codificación de $1/n$ para códigos perforados.

Además de los códigos perforados, se define el concepto de los Códigos Perforados de Tasa Compatible (RCPC por sus siglas en inglés), introducido por primera vez por Hagenauer [35]. Estos códigos constituyen una familia, y son construidos a partir de un “código madre”, haciendo perforaciones periódicas. En la literatura se tiene una serie de artículos donde se muestra la investigación para encontrar códigos perforados óptimos, con ellos es posible obtener diferentes matrices de perforación de las que resultan diferentes tasas altas de codificación [35] [36] [37], a partir de un solo código madre.

2.3.2.3 Decodificación convolucional.

Basada en un criterio de máxima verosimilitud, la decodificación convolucional consiste en encontrar la secuencia codificada que se transmitió en un canal con ruido, dentro de la secuencia de datos recibida. En un proceso de detección de señales con ruido, mientras se transmite la información, es posible que hayan ocurrido diversos factores que representan cambios en los datos recibidos que se traducen en errores. Entonces en el método de la decodificación convolucional, para disminuir la probabilidad de error, se buscará la secuencia más parecida a la original, efectuando la detección y corrección de errores.

Como se vio en la sección 2.3.2.1, el codificador convolucional consiste en una máquina de estados, sin embargo, la decodificación resulta ser más compleja. Existen diversos algoritmos que logran realizar la decodificación convolucional tales como: decodificación de Viterbi, decodificador secuencial y decodificador por umbral. La decodificación de Viterbi se emplea en este trabajo, debido a que es óptima en el sentido de que explora todos los caminos posibles en el enrejado para escoger el mejor, produce decodificación con decisión dura, y otras características que se detallan en este apartado.

La decodificación convolucional mediante el algoritmo de Viterbi está basada en un algoritmo que opera bajo el criterio de máxima verosimilitud. En este contexto, el decodificador escoge una trayectoria $U^{(m)}$ particular como la más probable de que haya sido la secuencia transmitida. Es decir, la medida de verosimilitud $P(Z | U^{(m)})$ (para toda secuencia Z recibida) es la mayor respecto a todas las otras posibles secuencias transmitidas [25].

De esta forma, el algoritmo de Viterbi implica el cálculo de la secuencia de estados a través del diagrama trellis mediante la comparación de las métricas de todas las trayectorias de las derivaciones que entran en cada estado, con respecto al vector recibido correspondiente, escogiendo la trayectoria con menor distancia acumulada. En el cálculo de la métrica se usa el concepto de decisión dura, lo que implica que en el proceso de decisión implícito se trabaja con dos regiones que corresponden a los valores binarios 0 y 1 [25].

El algoritmo de Viterbi se puede describir en los siguientes cálculos principales [31]:

1. En $t=0$, se inicializan los nodos del enrejado con valores de distancias y métricas nulas.
2. En la unidad de tiempo $t = m$, para cada uno de los $2^{(K-1)}$ estados del enrejado, se calculan las distancias de las 2^b ramas que entran a cada estado, con respecto al segmento de la secuencia recibida que corresponde en el mismo tiempo. Se calcula el valor de la métrica acumulada, sumando el valor de la métrica sobreviviente que conecta con los estados en los tiempos anteriores con cada una de las distancias que se acaban de medir. Y para cada estado, se escoge como trayectoria sobreviviente aquella cuya distancia acumulada respecto a la secuencia recibida sea la más baja. En caso de empate se escoge arbitrariamente una de ellas.
3. Si $t < L + m$, se incrementa t en uno y se repite el paso 2. De lo contrario, el algoritmo termina.

Existen $2^{(K-1)}$ rutas sobrevivientes desde la unidad de tiempo m hasta el tiempo L , una por cada uno de los $2^{(K-1)}$ estados. Agregando ceros a la secuencia de datos para terminar en el estado nulo, después del tiempo L hay menos sobrevivientes, puesto que hay menos estados mientras el codificador regresa al estado nulo. Finalmente, en el tiempo $L + m$, solo queda un estado, el estado nulo y, por lo tanto, solo existe una trayectoria sobreviviente, y dado que esta tiene la distancia acumulada más baja respecto a la secuencia recibida, es la de máxima verosimilitud [31]. El algoritmo de Viterbi se resume en la Figura 2.15 [30].



Figura 2.15: Representación del algoritmo de Viterbi.

Para ilustrar el algoritmo de Viterbi, se retomará el codificador de la Figura 2.11, cuyo diagrama de enrejado está en la Figura 2.13. Con la secuencia de datos 1010 polinomios generadores $g_1(x) = 1 + x + x^2$, $g_2(x) = 1 + x^2$, representados en octal (7,5), la codificación resultante fue 11, **10**, 00, 10, 11. Entonces, suponiendo que se produce un error en la transmisión, que afecta al tercer bit, se recibe la secuencia: 11, **00**, 00, 10, 11.

a) El decodificador empieza en el estado nulo y se calcula en todas las posibles trayectorias, la distancia entre la salida asignada a esa transición y el par de bits recibidos. Para los dos primeros bits recibidos y las dos trayectorias correspondientes en el tiempo $t=1$, se tiene: $d(00,11) = 2$ y $d(11,11) = 0$, Figura 2.16 a), se elige la distancia mínima entre ambas distancias, que corresponde al estado 10.

- b) Para la siguiente transición $t = 2$, a partir del estado 10, que es el que tiene la distancia acumulada más baja, sabiendo que pudiera estar en el estado 01 o 11, los posibles símbolos serían 10 y 01, entonces se tiene la medida de distancia $d(10,00) = 1$ y $d(01,00) = 1$, Figura 2.16 b). Como ambas distancias son iguales, entonces se deben analizar ambos estados.
- c) Para la siguiente transición $t = 3$, como se dijo en el inciso anterior, primero se analizará el estado 11: teniendo en cuenta que el símbolo recibido es 00 y sabiendo que se puede estar tanto en el estado 01, como en el estado 11, los posibles símbolos para el estado 11 serían 01 y 10 respectivamente, Figura 2.16 c). Ahora analizando al estado 01: sabiendo que se puede estar tanto en el estado 00, como en el estado 10, los posibles símbolos serían: 11 y 00 respectivamente, Figura 2.16 c). Entonces las distancias acumuladas en el nodo c serían $d(11,00) = 1+2=3$, $d(00,00) = 1+0=1$ mientras que en el nodo d serían $d(01,00)=1+1=2$ y $d(10,00)=1+1=2$, Figura 2.16 c); por lo tanto, se elige el nodo c hacia el nodo b, dado que la distancia acumulada es la más baja.
- d) Para la siguiente transición $t=4$, teniendo en cuenta que el símbolo recibido es 10 y sabiendo que se puede estar tanto en el estado 01, como en el estado 11, Figura 2.16 d), los posibles símbolos serían 10 y 01, entonces se tiene que $d(10,10)=1+0=1$ y $d(01,10)=1+2=3$; por lo tanto, se elige el nodo b hacia el nodo c.
- e) Para la siguiente transición $t=5$, teniendo en cuenta que el símbolo recibido es 11 y sabiendo que se puede estar tanto en el estado 00 como en el estado 10, los posibles símbolos serían 11 y 00, entonces se tiene que $d(11, 11)=1+0=1$ y $d(00,11)=1+2=3$; por lo tanto, se elige el nodo c hacia el nodo a y finaliza el algoritmo. En la Figura 2.16 se describe el ejemplo propuesto, así como en la Figura 2.17 el resultado de la decodificación de Viterbi.

En el ejemplo presentado, se verificó que el algoritmo de Viterbi detecta y corrige correctamente la secuencia recibida. Para el código del ejemplo, si el número de errores hubiera sido mayor a 2, en una longitud de la secuencia recibida de entre 3 y 5 veces la longitud de restricción $K=3$, no habría sido posible encontrar este camino, y daría como resultado un camino erróneo. Entonces teniendo esta premisa, para calcular la capacidad de corrección de errores del código convolucional, se tiene la distancia libre d_f , que permite corregir t errores en una longitud de la secuencia recibida de entre 3 y 5 veces la longitud de restricción K , siempre que [25]:

$$t = \left\lfloor \frac{d_f - 1}{2} \right\rfloor \quad (2.18)$$

Considerando la ecuación (2.18), donde los corchetes cerrados hacia abajo, indican la parte entera del cociente, el número de errores que corrige el código propuesto en el ejemplo anterior es de $t = 2$, en una longitud de entre 9 y 15 bits de la secuencia recibida, dado que $d_f = 5$.

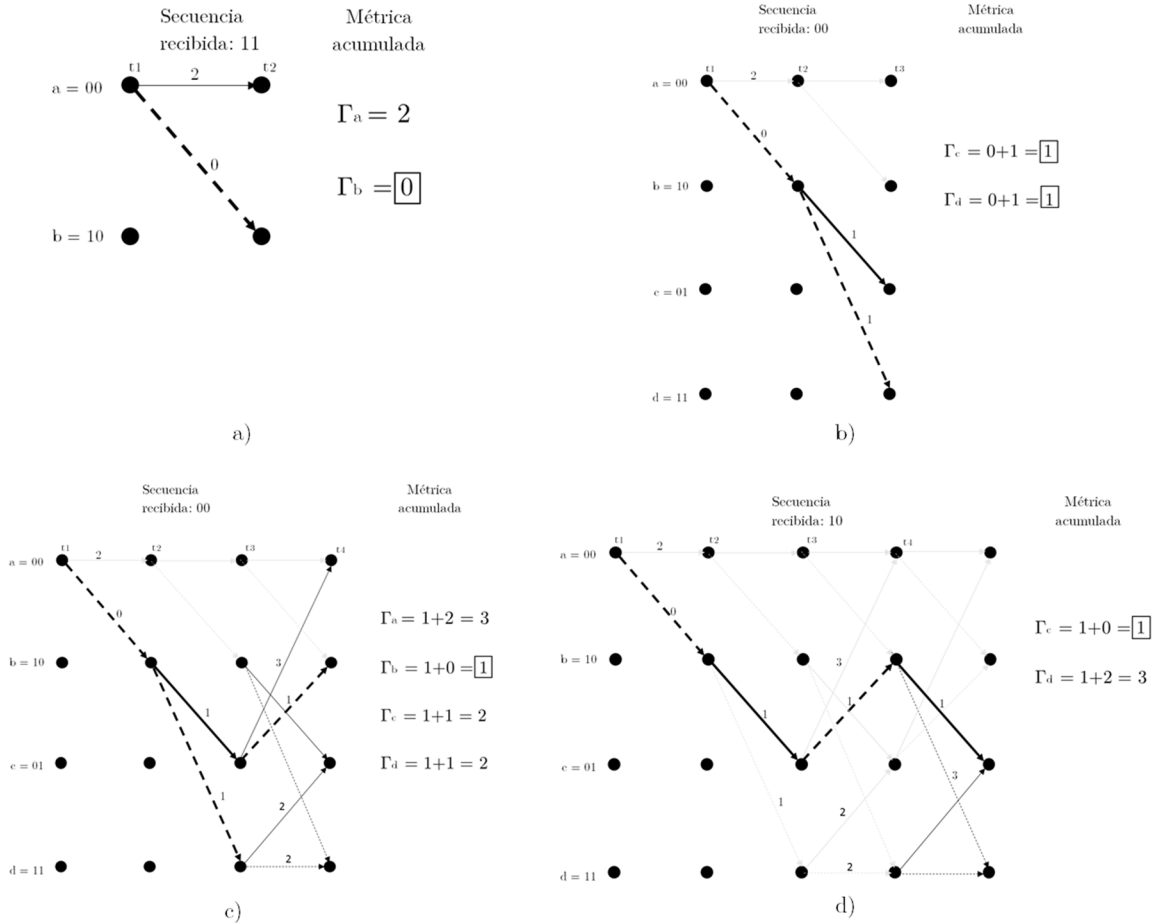


Figura 2.16: Selección de ramas supervivientes para decodificación convolucional.

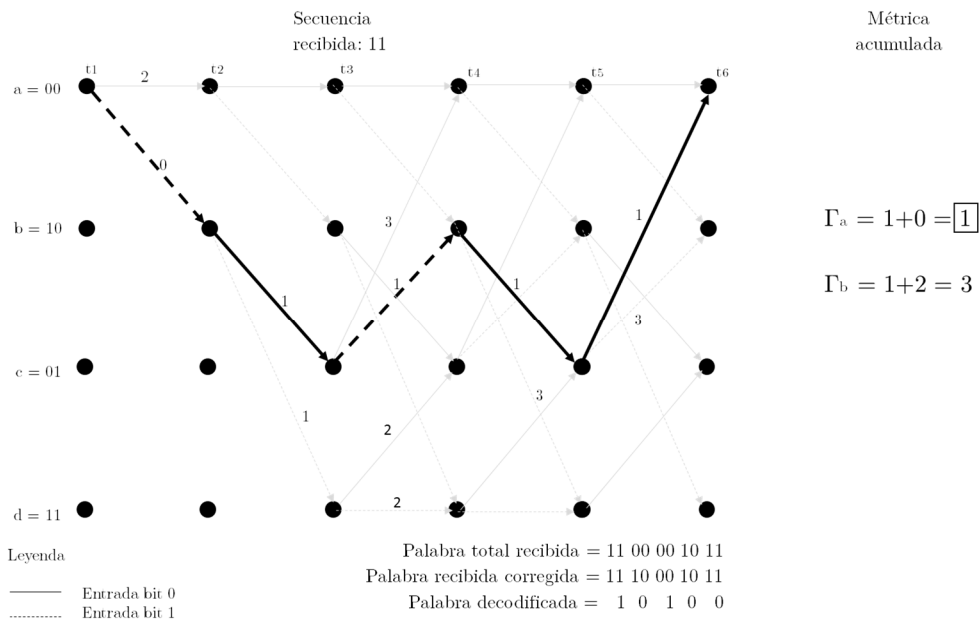


Figura 2.17: Palabra resultante de la decodificación, bajo el criterio de máxima verosimilitud.

2.4 Esquemas de Protección contra errores.

Con el propósito de poder hacer comparaciones y establecer una estrategia, en este trabajo se emplean dos tipos de códigos binarios muy utilizados en los sistemas de comunicaciones digitales, siendo uno de cada tipo: de bloque y convolucional. Los códigos Bose-Chadhuri-Hocquenghem (BCH) y convolucionales, poseen una gran variabilidad de tasa de la codificación y capacidad de corrección para adaptarse a diferentes aplicaciones. Sus parámetros principales son, en el caso de los códigos BCH, su distancia mínima d , capacidad de corrección t , longitud de la palabra codificada n , dimensión del código k , tasa de codificación k/n y polinomio generador $g(x)$ de grado $(n-k)$. En el caso de los códigos convolucionales, se tiene: su distancia libre d_f , capacidad de corrección t errores en una longitud de la secuencia recibida de entre 3 y 5 veces la longitud de restricción K , se generan n bits codificados por cada k bits de información, tasa de codificación k/n y polinomios generadores $g_i(x)$, donde el número i depende de la tasa k/n y el grado de los polinomios depende de la longitud de restricción K .

El objetivo de esta tesis es poder comparar dos estrategias de corrección de errores: EEP y UEP. EEP (equal error protection) es un esquema más simple que se encarga de proteger los datos por igual, sin importar su significado cualitativo, esto significa que independientemente del peso relativo de los bits, el número de bits de paridad en la palabra de código es el mismo [38]. En contraposición, el esquema UEP (unequal error protection) consiste en la asignación de la paridad en la información que se desea transmitir en función de la importancia relativa de sus bits [4], es decir los bits más significativos recibirán mayor protección. La hipótesis consiste en plantear un esquema bajo el principio UEP que requiera en promedio menos redundancia, manteniendo la misma protección de la información.

Utilizando los radios definidos por software, se llevaron a cabo experimentaciones considerando dos escenarios: a) transmisión en un canal inalámbrico sin sistema de control de errores, b) con un sistema de control de errores, aplicando los esquemas EEP y UEP, con códigos de bloque BCH, y convolucionales. Finalmente se hicieron comparaciones para obtener las conclusiones del estudio.

2.5 Plataforma de Radio Definido por Software.

A manera de introducción del Radio Definido por Software, es importante recordar el concepto de radio en las comunicaciones digitales. Un radio es cualquier equipo usado para intercambiar información digital entre dos puntos, en la Figura 2.18 [39] se presenta un diagrama de bloques general de un sistema de radio. En el receptor, los elementos que lo conforman se describen a continuación [39]:

Antena. Las ondas electromagnéticas inciden en la antena y se convierten en señales eléctricas.

RF front end. La señal eléctrica es típicamente débil y puede ser corrompida por niveles bajos de ruido. El ruido de la antena debe ser filtrado y la señal amplificada antes de ser procesada. Este bloque determina la relación señal a ruido (SNR) con la que el resto del radio debe trabajar. Se compone de un filtro, un amplificador con bajos niveles de ruido y un mezclador para mover la señal de la frecuencia del radio a una frecuencia inferior. Por tanto, el mezclador se encarga de trasladar adecuadamente el espectro a otro rango de frecuencias y amplificar la parte central [40].

Procesamiento Digital de Señales (DSP). Se aplica para extraer la información contenida en la señal eléctrica digitalizada. Las funciones implementadas en la sección de DSP se determinan por los detalles de la forma de onda, generalmente incluyen: adquisición de la señal (adquirir la señal que fue transmitida e introducir compensaciones entre lo que se espera y lo que se recibe), demodulación (mapear la señal recibida, actualizar las compensaciones), decodificación (corregir y detectar errores) y descifrado (quitar la seguridad para que los datos estén disponibles para el usuario).

Redes y aplicaciones. El destino de la información transmitida.

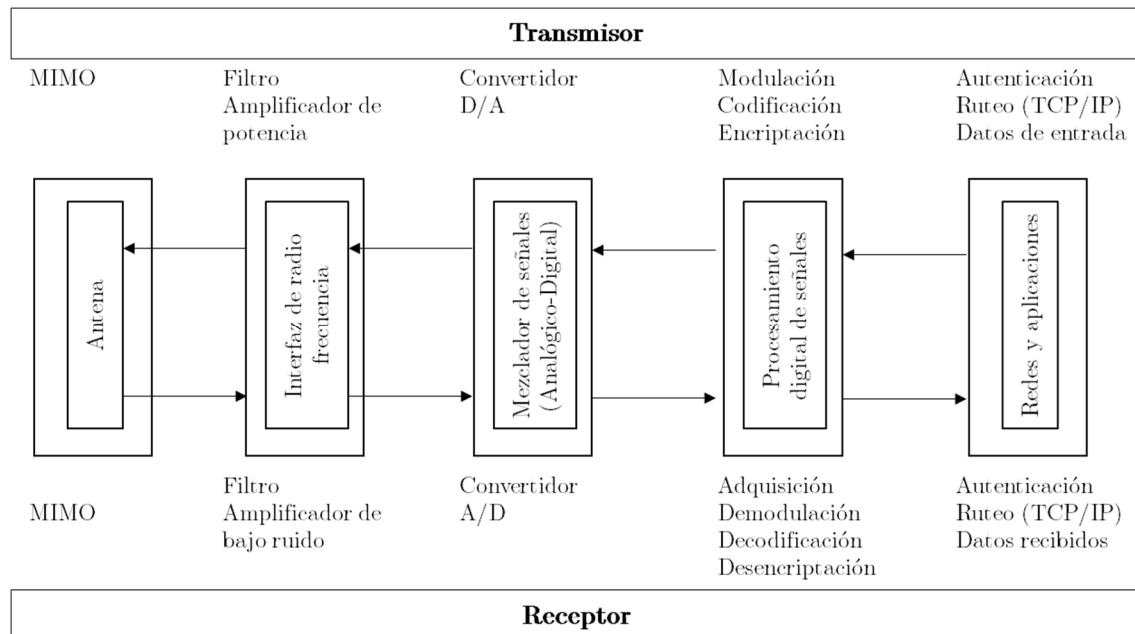


Figura 2.18: Diagrama de bloques general del funcionamiento de un sistema de radio.

2.5.1 Radio Definido por Software (SDR).

Aun no existe una definición única del sistema SDR, pero el común denominador de todas las definiciones es que se trata de una plataforma abierta de hardware que puede ser fácilmente configurable. Actualmente, la mayoría de los radios no están definidos por

software, sino controlados por software como en el caso del teléfono celular, que puede soportar diferentes tecnologías (GSM, WCDMA, LTE). El usuario no requiere accionar un interruptor físico para acceder a la red, la selección de la red es manejada por el software. Entonces, la evolución que se mencionó en el apartado 2.1.2, consiste en que ahora el radio controlado por software se convirtió en un radio definido por software, es decir, el software define la interconexión entre los bloques que constituyen el aparato, de tal modo que puede realizar cualquier algoritmo de procesamiento de señales [39]. Por lo tanto, la tecnología SDR está conformada por hardware y software, que puede ser reconfigurada dinámicamente para lograr una variedad de algoritmos y protocolos de comunicaciones.

La estructura general de un SDR con su respectivo software y hardware, se muestra en la Figura 2.19 [40].

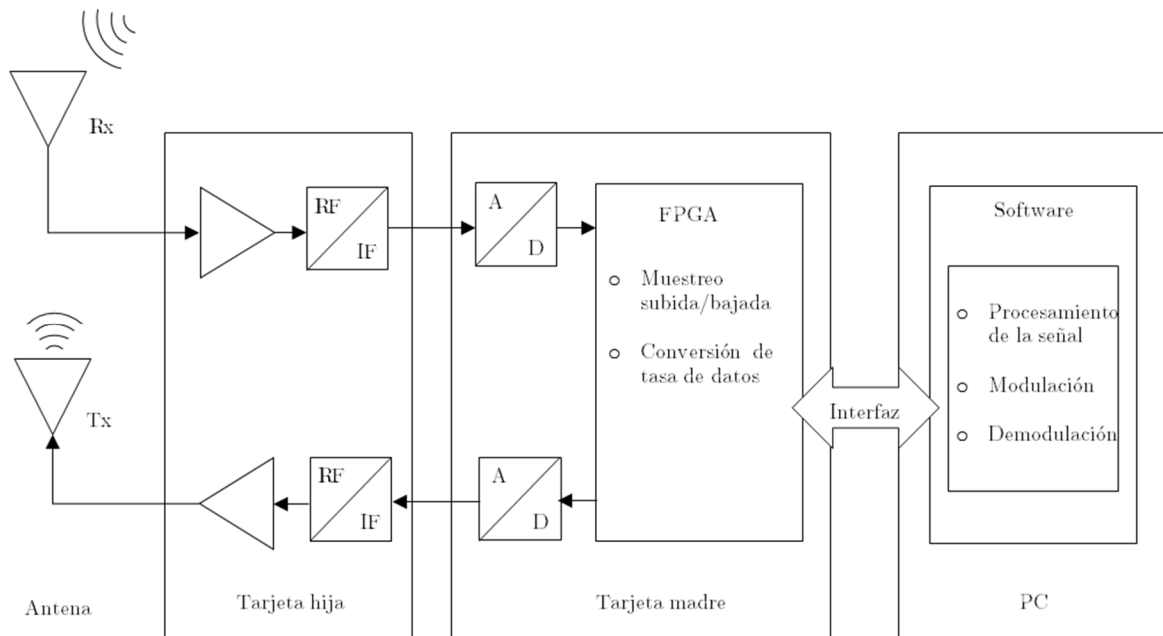


Figura 2.19: Diagrama de bloques general de la tecnología SDR.

En el SDR, de acuerdo a la arquitectura la transmisión empieza en la tarjeta llamada “madre”, la cual se compone principalmente de un FPGA (del inglés Field Programmable Gate Array) y convertidores analógico digital y digital – analógico. Desde el punto de vista electrónico, el FPGA consiste principalmente en elementos lógicos reconfigurables y una matriz de conmutación para realizar el ruteo de las señales entre ellas, estos dispositivos pueden ser configurados para desarrollar operaciones lógicas simples, o más complejas, como filtros digitales, o modulaciones [44]. La señal en banda base es transferida al FPGA para ser interpolada (DDC por sus siglas en inglés) con el fin de alcanzar la frecuencia y la tasa de muestreo de salida deseada. Posteriormente realiza la conversión D/A, la señal analógica resultante entra al modulador de cuadratura, luego se pasa por un mezclador junto con la

frecuencia del oscilador local, después por un filtro pasa banda para llevarla a un amplificador de potencia y, por último, es transmitida por la antena.

Para la recepción, la señal de radio es recibida por la antena, pasa a través de la tarjeta llamada “hija”, o tarjeta secundaria, dentro de dicha tarjeta, la señal recibida se amplifica por medio de un amplificador de nivel bajo de ruido, después se sintoniza a la frecuencia del oscilador local, posteriormente ambas frecuencias (la del oscilador y la de la señal de radio) se mezclan para obtener la señal en banda base. En seguida el resultado es enviado a la tarjeta madre para realizar la conversión D/A y, finalmente la señal resultante se pasa por el proceso de decimación (DUC por sus siglas en inglés), y así la señal proveniente del SDR es transferida a la PC mediante una interfaz para realizar el procesamiento en software, con la plataforma de desarrollo de software que el usuario desee usar [40].

2.5.1.1 Justificación del uso de SDR.

En la sección 1.2 del capítulo 1, se observan las aplicaciones y avances que se han empleado con la tecnología SDR, por lo que es importante centrar la importancia y los motivos de su uso. Como se sabe, estas funciones poseen ventajas y desventajas [39] que se mencionan en la Tabla 2.1

Tecnología SDR	
Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ Interoperabilidad: el SDR puede comunicarse sin problemas con múltiples radios incompatibles o actuar como puente entre ellos. ▪ Uso eficiente de los recursos bajo condiciones variables. ▪ Reutilización de frecuencias (radio cognitiva): el SDR puede utilizar el espectro que no se esté usando hasta que vuelva a ser usado. 	<ul style="list-style-type: none"> ▪ Costo: la inversión puede ser elevada debido a que el hardware de un SDR es complejo. ▪ Consumo de potencia: dos componentes en el SDR contribuyen a mayor consumo de energía: aumento de la complejidad en el DSP y mayor ancho de banda en la señal RF mixta. Sin embargo, la adaptación de la forma de onda habilitado por el SDR puede reducir la potencia de transmisión, por consiguiente, ahorrar energía.

<ul style="list-style-type: none"> ▪ Obsolescencia reducida: el SDR puede actualizarse en cuanto a los últimos estándares de comunicaciones, ahorrando costo de un nuevo hardware y la labor de instalación. ▪ Bajo costo: el SDR puede adaptarse a múltiples mercados y múltiples aplicaciones. ▪ Investigación y desarrollo: El SDR se puede usar para desarrollar diferentes aplicaciones y analizar su rendimiento en tiempo real. 	<ul style="list-style-type: none"> ▪ Alcance limitado. Al hablar de SDR es importante tener en cuenta que solo se refiere a la capa física, por lo que el usuario no puede tomar ventaja del rendimiento del enlace sin la cooperación de las capas superiores.
---	--

Tabla 2.1: Ventajas y desventajas del SDR.

Conforme lo descrito en la Tabla 2.1, la mayor ventaja que se puede tener mediante la tecnología SDR para el propósito de esta tesis, es en el aspecto de la investigación y el desarrollo, debido a que se pueden realizar múltiples configuraciones para implementar y probar el sistema de comunicaciones propuesto sin necesidad de invertir mucho tiempo, y costo, en componentes y programación sofisticada.

2.5.1.2 Hardware USRP2 de Ettus Research².

El *Universal Software Radio Peripheral* (USRP) es una plataforma SDR desarrollada por la compañía Ettus Research desde 2010, elegida por ingenieros, científicos y estudiantes de todo el mundo para la creación y desarrollo de algoritmos, así como prototipos de tecnología inalámbrica, realizando varias aplicaciones. A manera de resumen y con objeto de ubicar la versión utilizada en esta tesis, en la Tabla 2.2 [41] [42] [45] [46] se presenta a la familia de USRP actualmente en existencia.

² Ettus Research empresa líder proveedora de SDR.

Modelo	Rango de Frecuencia de operación ³	Interfaz	Procesamiento		Consumo de energía	Precio
			Tasa ADC (MS/s)	Tasa DAC (MS/s)		
USRP X310	1.2 GHz a 6 GHz 50 MHz a 2.2 GHz 400 MHz a 4.4 GHz	Gigabit Ethernet	200	800	45 W	\$99,445.00 MX
USRP X300	1.2 GHz a 6 GHz 50 MHz a 2.2 GHz 400 MHz a 4.4 GHz	Gigabit Ethernet	200	800	45 W	\$80,765.00 MX
USRP N210	400 MHz a 4.4 GHz 1.5 GHz a 2.1 GHz 50 MHz a 2.2 GHz	Gigabit Ethernet	100	400	13.8 W	\$35,645.00 MX
USRP N200	400 MHz a 4.4 GHz 1.5 GHz a 2.1 GHz 50 MHz a 2.2 GHz	Gigabit Ethernet	100	400	13.8 W	\$31,460.00 MX
USRP B100	50 MHz a 2.2 GHz 400 MHz a 4.4 GHz	USB 2.0	64	128	9.6 W	\$12,203.00 MX
USRP1	750 MHz a 1050 MHz	USB 2.0	64	128	10.2 W	\$14,680.00 MX
USRP E100	50 MHz a 2.2 GHz 400 MHz a 4.4 GHz	Embebido	64	128	13.8 W	\$24,406.00 MX

Tabla 2.2: Características de USRP por modelo.

El USRP que se emplea en este trabajo es el denominado USRP N210 el cual se ilustra en la Figura 2.20 [41], consiste de dos tarjetas principales llamadas: tarjeta madre y tarjeta hija.

La **tarjeta madre** contiene toda la electrónica en donde se desarrolla el procesamiento de la señal en banda base, está constituida de un FPGA (para ver más detalles, sus especificaciones están dadas en [42]); la **tarjeta hija**, cuenta con un transmisor y receptor que comparten la misma circuitería. El modelo de la tarjeta hija es el RFX1800, el cual no es especificado en los datos de fabricante del USRP2, se puede consultar mediante hardware (abriendo el radio) y/o software (en GNU Radio con el comando `$ uhd_usrp_probe`), el rango de frecuencias va de 1.5 a 2.1 GHz, con un ancho de banda de 40 MHz y salida de 31 mW.

³ Depende de la tarjeta hija que tenga integrada.



Figura 2.20: Imagen del SDR USRP N210.

Las principales razones por las que se eligió esta serie de USRP son: su precio, su gran disponibilidad en software libre, su interfaz Gigabit Ethernet, ofrece un FPGA más grande que la serie N200 para aplicaciones que demandan recursos de lógica, memoria y DSP adicionales. Además, el firmware del FPGA se puede recargar a través de la interfaz Ethernet y ofrece la posibilidad de procesar hasta 100 MS/s tanto en transmisión como en recepción [41]. El USRP N210 soporta Linux, Mac OSX, Windows. De acuerdo a lo que menciona Ettus Research, la mayoría de los SDR USRP tienen soporte de software en: GNU Radio, LabView y Matlab [43].

Para los propósitos de este proyecto, el desarrollo se enfoca en verificar y validar los algoritmos de codificación a partir de una fuente de datos en un canal real, por lo que solo se usó el USRP para poder transmitir y recibir los datos, especificando que no se está validando el funcionamiento interno del USRP, tampoco se ha programado a bajo nivel, las funciones internas del radio son independientes de los requerimientos de este proyecto.

2.5.2 GNU Radio

En el presente apartado, se explica el funcionamiento de la plataforma de desarrollo GNU Radio, utilizada.

De acuerdo a la bibliografía, el proyecto de GNU Radio fue fundado por Eric Blossom con la intención de crear y abrir un marco para SDR. GNU Radio es un conjunto de herramientas de desarrollo de software libre y de código abierto que proporcionan bloques de procesamiento de señales para implementarlos en los SDR; se puede desarrollar en hardware o en un entorno de simulación. Es ampliamente usado en entornos académicos, comerciales y de aficionados para apoyar la investigación de las comunicaciones

inalámbricas y sistemas de radio en el mundo real [47]. GNU Radio corre en el sistema operativo Linux, también lo puede hacer en Mac y Windows. Sin embargo, es algo complejo debido a que es necesario usar la herramienta Cygwin.

2.5.2.1. Arquitectura GNU radio

Usando GNU Radio, un SDR puede ser construido mediante grafos donde los vértices son bloques de señales procesadas y los enlaces entre vértices, representan el flujo de datos entre ellos. Los componentes de GNU Radio están conectados como se observa en la Figura 2.21 [40], los bloques de la señal procesada están implementados en C++ y los grafos son construidos y corridos en Python junto con la interfaz SWIG. Python es un lenguaje de programación orientado a objetos interpretado, lo que significa que no se compila y el mismo sistema operativo lo ejecuta directamente.

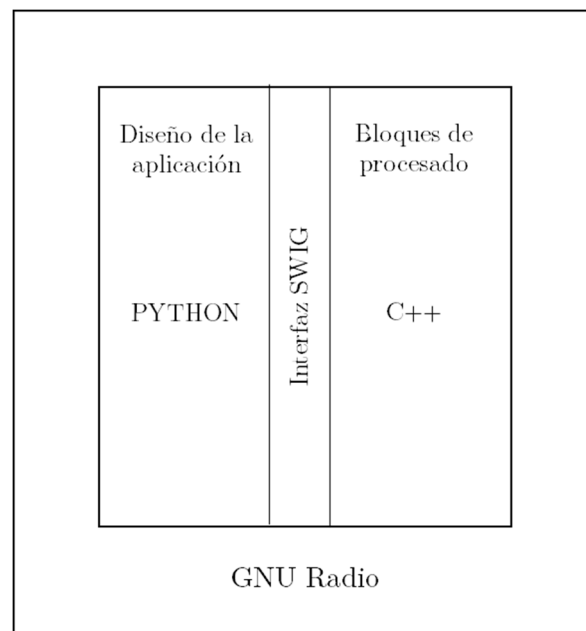


Figura 2.21: Arquitectura GNU Radio.

La herramienta gráfica para crear grafos de flujo de señales y generar el código fuente del grafo se le llama GNU Radio Companion (GRC) (parecida a Matlab Simulink). Los bloques principales empleados se pueden describir de la siguiente manera [40]:

a) Fuentes (Sources). Para cualquier tipo de fuente como audio, imagen, generador de señales, un USRP, micrófono, etc.

- b) Bloques de procesamiento de señal. Son los bloques que corresponden a un procesamiento de señal de cualquier prototipo tal como filtros, amplificadores, moduladores, etc.
- c) Sumideros (Sink). Es el destino de una señal recibida, puede ser un archivo, audio o el mismo USRP, también se refiere a visualización señales como osciloscopio, analizador de espectros, etc.

En las Figuras 2.22 y 2.23 se presenta un ejemplo de un diseño realizado en GRC, del cual se observa además del sistema de bloques descritos anteriormente, otros bloques que representan variables e identificadores. Como se puede observar, es una forma amigable de diseñar un sistema, y sin necesariamente saber programar en Python, o C, ++. La limitación es que solamente se puede trabajar con las funciones predeterminadas en el GRC, por lo que, si se desea agregar algún modulo, es necesario programarlo y agregarlo al GRC como lo citan en [48], en donde mencionan los archivos y extensiones que se deben de crear para personalizar un módulo (o bloque).

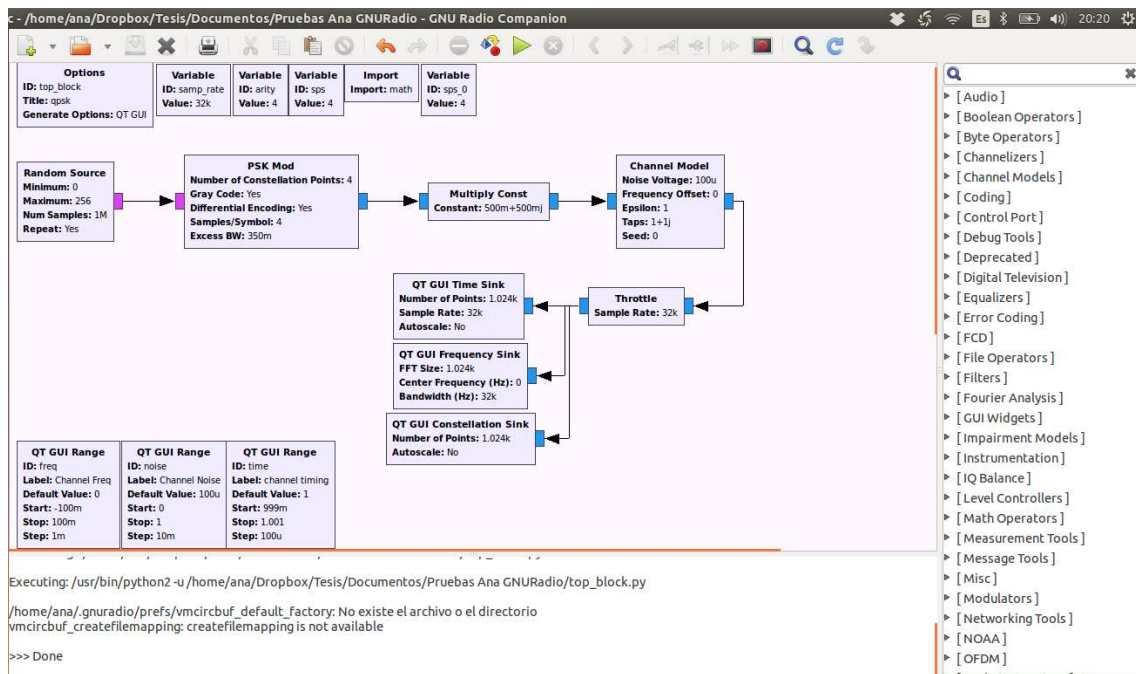


Figura 2.22: Ejemplo de la transmisión y modulación de una señal en GNU Radio Companion.

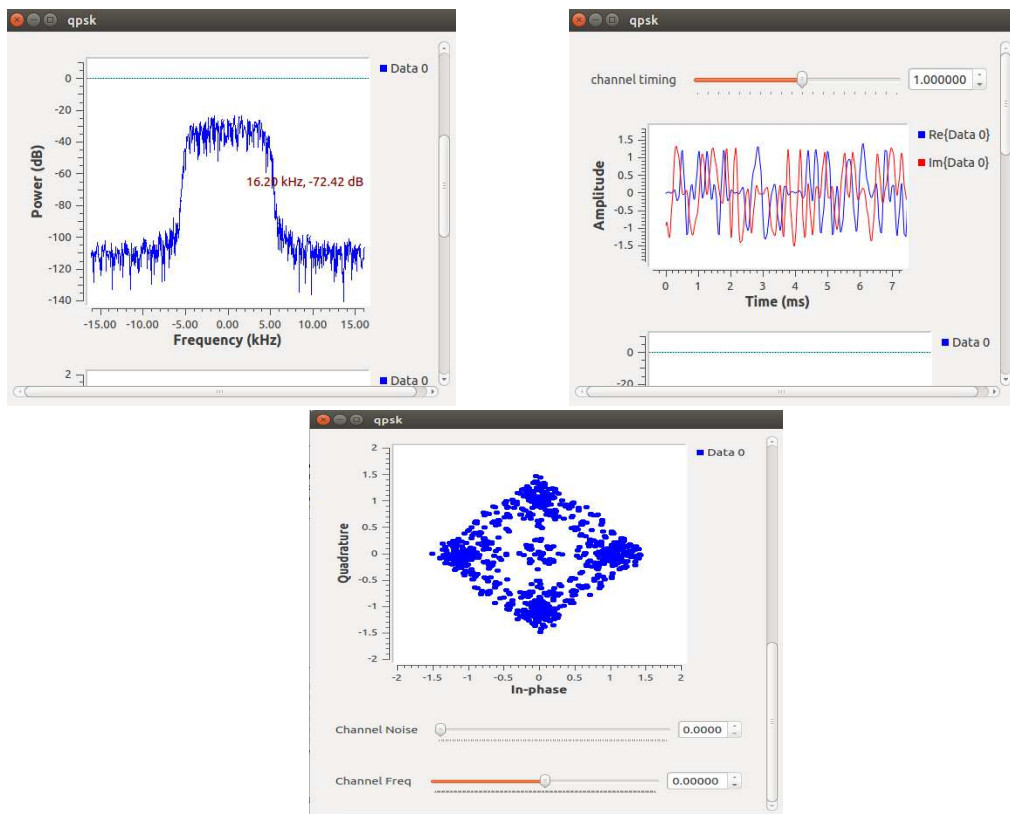


Figura 2.23: Señal Rx del ejemplo de la Figura 2.22.

2.5.2.2. Comunicación entre USRP N210 y GNU Radio

En los apartados anteriores referidos a USRP N210 y GNU Radio se han descrito las funciones de cada uno de dichos componentes, con esto se tiene un panorama completo de la relación que tienen ambas plataformas. En nuestro trabajo, la etapa de transmisión, comienza con el desarrollo del sistema de comunicaciones en GRC, en el que todo el procesamiento se realiza en banda base, esto incluye los procesos de codificación de canal, codificación de fuente, decodificación de canal, decodificación de fuente, modulación, demodulación, filtros, etc., los paquetes resultantes del proceso son enviados en tramas por Ethernet al Controlador de Hardware Universal (UHD por sus siglas en inglés), las mismas tramas se envían también al USRP N210.

El FPGA recibe las tramas mismas que pasan por un proceso de interpolación, de tal manera que debe coincidir con la tasa de entrada de datos (100 MS/s). También por un proceso de sincronización para la frecuencia central requerida, después se hace la conversión digital – analógica de las muestras adquiridas y el resultado se envía a la tarjeta hija, la

cual traslada la señal a la frecuencia central deseada, para finalmente, encaminar la señal a la antena para transmitirla ya como ondas electromagnéticas. Para la etapa de recepción, es exactamente el mismo procedimiento, solo con algunas variantes, por ejemplo, dado que la tarjeta madre realiza la conversión analógico – digital, el FPGA recibe la señal digitalizada para después realizar el proceso de decimación, posteriormente envía las tramas de los datos por la interfaz Ethernet, y así poder procesar la información recibida en el GRC.

Capítulo 3

Metodología propuesta para el diseño de un sistema de control de errores desigual UEP, en imágenes

En el presente capítulo, se recapitulan los objetivos de esta tesis de manera general. Los cuales consisten en: implementar, y probar el esquema de UEP sobre una imagen en el sistema de comunicaciones propuesto (usando simulaciones por computadora, y por medio de hardware con el uso de la tecnología SDR). Para posteriormente, poder definir una estrategia de diseño que permita realizar comparaciones de la imagen recibida resultante con el esquema EEP, y con el sistema sin codificación.

3.1 Descripción del sistema de comunicaciones propuesto

El primer paso para emplear la técnica UEP es dejar claramente establecidos los elementos que conforman nuestro sistema de comunicaciones. En esta sección, se presentan las partes principales que constituyen el sistema de comunicaciones propuesto, como se muestra en la Figura 3.1, y a continuación se detalla cada bloque:

Imagen binaria original: el usuario puede transmitir cualquier imagen que desee, debido a que este bloque realiza el procesamiento sobre la imagen original, de forma que entrega al siguiente bloque una matriz binaria. El procesamiento consiste en: i) cambiar la imagen a tonos de gris con 256 niveles, ii) convertir la imagen al formato bmp, iii) ajustar las dimensiones de la imagen, según la necesidad que se requiera (opcional), iv) dividir la imagen en 8 capas binarias para hacerlas vectores, y v) hacer un arreglo con los 8 vectores binarios dentro de una matriz.

Codificador de canal. Es el encargado de agregar redundancia a la imagen, para protegerla contra errores, los cuales son producidos por el ruido del canal inalámbrico al momento en el que ocurre la transmisión de la imagen. Este bloque está conformado por dos tipos de codificadores de canal: de bloque y convolucional (cada uno empleado en el sistema de comunicaciones, tanto simulado, como real). Además, en la codificación de canal, se pueden utilizar las técnicas de EEP y UEP. Los parámetros de los códigos empleados en los sistemas de comunicaciones propuestos, se darán en las siguientes secciones.

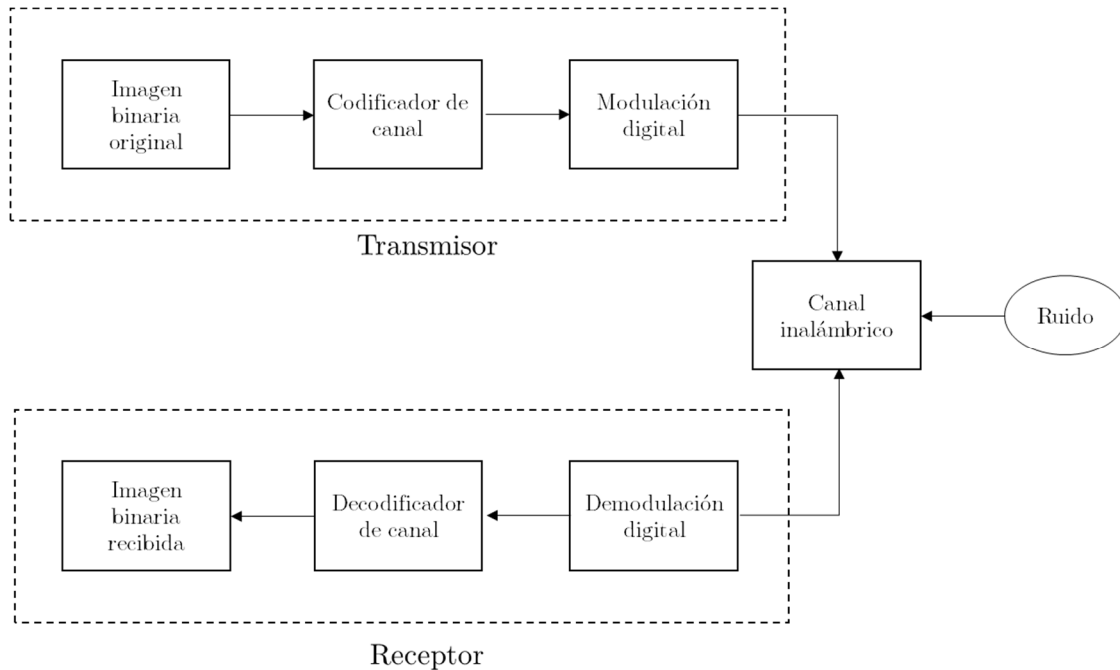


Figura 3.1: Sistema de comunicaciones digital inalámbrico empleado.

Modulación digital. En esta etapa, la modulación digital empleada está conformada por dos moduladores: BPSK (usada en el sistema simulado), y GFSK (para el sistema real), mismas que fueron seleccionadas por sus características previamente expuestas en las secciones 2.2.1, y 2.2.2 del capítulo dos de este trabajo. Los parámetros de diseño se especifican en las siguientes secciones.

Canal inalámbrico. En este bloque se emplean dos tipos de canales: el canal AWGN para el escenario simulado, cuyas características fueron previamente descritas en la sección 2.2.3 del presente trabajo, y el canal real que está dado por la tecnología SDR.

Demodulación digital. En la parte del receptor, se manejan los demoduladores digitales binarios BPSK y GFSK, respectivamente. Encargados de recuperar la señal original

transmitida para cada caso, cuyas características fueron descritas en las secciones 2.2.1 y 2.2.2 de este trabajo.

Decodificador de canal. La función de este bloque es detectar, y corregir los bits erróneos causados por el ruido. Están presentes en la información entregada por el demodulador digital. Con fines de comparación, los decodificadores de canal utilizados en este bloque son: decodificador BCH (bloque), y decodificador de Viterbi (convolucional), descritos en la sección 2.3 de esta tesis, ambos empleados tanto para el escenario simulado, como real.

Imagen binaria recibida. Son los datos resultantes del desarrollo antes descrito, los cuales se procesan conforme a las siguientes etapas: i) recupera la matriz binaria, ii) separa ocho vectores binarios de cada columna de la matriz, iii) convierte los ocho vectores binarios en matrices binarias, iv) suma las matrices para obtener la imagen recibida, v) la imagen se convierte al tamaño, y formato que el usuario prefiera.

3.2 Descripción de la metodología propuesta.

El segundo paso, consiste en diseñar una metodología haciendo uso de la técnica UEP para la transmisión de imágenes digitales, empleando los codificadores de canal BCH y convolucional, con el propósito de evaluar el ahorro en redundancia con respecto a la técnica EEP. Para empezar con la estrategia de diseño, se requiere conocer las medidas de calidad de la imagen, que permitirán evaluar el desempeño de cada codificador y esquema de protección implementado. Las medidas de calidad de la imagen usadas en este trabajo son: PSNR, BER y SSIM, las cuales se describen en la siguiente sección.

3.2.1 Métodos de medición de calidad de imagen.

Los métodos de medición de calidad de imagen se aplican para medir la degradación de una imagen por el ruido, o para cuantificar la similitud entre imágenes. Los métodos que se emplean en este trabajo, definidos en la literatura, son los siguientes:

- **PSNR.** Para definir la PSNR, primero se calcula el Error Cuadrático Medio (MSE, por sus siglas en inglés). Su propósito es medir el promedio de los errores al cuadrado, para este caso, la diferencia al cuadrado entre la imagen original, y la imagen recibida, entre las dimensiones de la imagen. Luego, retomando la definición de PSNR, se calcula en escala logarítmica la razón entre la máxima energía posible de una imagen ficticia, sobre el error cuadrático medio. Las expresiones matemáticas de ambos términos son:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I(i,j) - K(i,j)|^2 \quad (3.1)$$

$$PSNR = 10 \log_{10} \left(\frac{Max^2}{MSE} \right) \quad (3.2)$$

Donde $Max = 2m - 1$, m representa el número de bits de profundidad de la imagen. El **caso ideal** sería si el MSE fuera de cero al no haber error en la imagen recibida; por lo tanto, el valor de la PSNR daría como resultado **infinito**.

- **BER.** Suponiendo ergodicidad, es la razón del número de bits erróneos entre el número total de bits procedentes de la imagen transmitida, lo cual equivale a la probabilidad de que un bit sea recibido incorrectamente; por lo tanto, el **caso ideal** para el BER es de **cero**. La probabilidad de bit en error (P_b) y la tasa de error de bit (BER) son términos que en la literatura se usan a menudo por igual, aunque estrictamente son conceptos diferentes. P_b es de carácter matemático (teórico), y BER es un registro empírico (práctico). En la Tabla 3.1, se presenta el cálculo de P_b y BER para las modulaciones BPSK [25] y GFSK [50], que fueron empleadas en este trabajo. El alcance que se tiene en este trabajo es del empleo del parámetro BER en la sección de la implementación.

Modulación	P_b	BER
BPSK	$Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$	$\frac{1}{N_{bits}} \sum_{n=1}^{N_{bits}} (x_{tx}[n] - x_{rx}[n])$
GFSK	$Q_1(a, b) - \frac{1}{2} \exp\left(-\frac{a^2 + b^2}{2}\right) I_0(a, b)$ <p>donde:</p> $a, b = \sqrt{\frac{E_b}{2N_0} \left(1 \pm \sqrt{1 - \left(\frac{\sin(2\pi h)}{2\pi h}\right)^2}\right)}$ <p>$Q_1(\cdot, \cdot)$ es el primer término de la función Q de Marcum, y I_0 es el orden 0 de la función de Bessel.</p>	$\frac{1}{N_{bits}} \sum_{n=1}^{N_{bits}} (x_{tx}[n] - x_{rx}[n])$

Tabla 3.1: Esquemas de modulación en términos de Probabilidad de bit en error (P_b), y Tasa de bit de error (BER).

- **SSIM.** Es una métrica de calidad de imagen, que evalúa el impacto visual de las tres características principales de una imagen: luminancia, contraste y estructura; su algoritmo se detalla en [49]. Cuando las dos imágenes son idénticas, **su valor es igual a 1.**

Su expresión matemática está dada por:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.3)$$

Donde:

μ_x = media de x

μ_y = media de y

σ_x^2 = varianza de x

σ_y^2 = varianza de y

σ_{xy} = covarianza de x e y

$C_1 = (k_1L)^2$ y $C_2 = (k_2L)^2$ = dos variables para estabilizar la división

L = es el rango dinámico de los valores de los pixeles ($2^{n^\circ \text{ de bits por pixel}} - 1$)

$k_1 = 0.01$ y $k_2 = 0.03$ = valores de forma predeterminada.

Dados los valores ideales de cada una de las medidas de calidad aplicadas a la imagen recibida, es posible empezar a valorar las diferentes metodologías propuestas para cada codificación de canal, utilizando las técnicas UEP y EEP en los diferentes escenarios del sistema de comunicaciones, empleados en este trabajo.

3.2.2 Diseño UEP para códigos correctores binarios: de bloque BCH y convolucional.

En esta sección, se presenta la metodología del diseño de la técnica UEP para cada sistema de comunicaciones propuesto en este trabajo de tesis, el primero basado en codificación de bloque con códigos BCH, y el otro basado en codificación convolucional. Estas pruebas tienen el propósito de obtener diversos códigos que estimen el desempeño del esquema UEP, con respecto al esquema EEP, y al sistema sin codificación. La metodología que se expone a continuación, comprende los escenarios simulados y reales.

UEP con códigos BCH.

1. Conforme a los resultados teóricos ideales discutidos en la sección anterior, se pretende que la imagen recibida presente los siguientes valores de calidad: a) PSNR=Inf [dB], b) BER=0, y c) SSIM=1. De acuerdo a esto, se busca un código BCH (n,k) en el esquema EEP que cumpla con estos criterios, para un valor de razón señal a ruido E_b/N_0 de interés, previamente seleccionado.
2. En la metodología se establece que para comenzar el diseño de la codificación UEP en la imagen, se requiere conocer los valores (n, k) del código BCH, corrector de errores en EEP, que lograron alcanzar el criterio ideal de los valores de calidad, mismos que se toman en cuenta como referencia para proteger la capa binaria más significativa de la imagen. Es conveniente aclarar que no necesariamente se debe aplicar el código BCH en EEP encontrado para la capa binaria más significativa, sin embargo, se busca que sea lo más cercano posible a ese código.
3. Se emplean diferentes códigos BCH (n, k) (respetando lo establecido en el paso 2), para aplicarlos en cada capa binaria, considerando la cantidad de información en cada una de ellas, debido a que la capa que contenga mayor información, debe estar mejor protegida que las capas subsecuentes. Es importante la protección en cada capa de la imagen, sin embargo, por un criterio de optimización, no se debe perder de vista que el esquema UEP debe satisfacer las siguientes condiciones: RedUEP < RedEEP, siendo RedUEP la redundancia promedio resultante de las capas de la imagen en el esquema UEP, y que los valores de calidad (PSNR, BER, y SSIM) deben ser lo más cercanos a su caso ideal. De acuerdo a nuestra experimentación, se recomienda que por lo menos se alcancen los siguientes valores: PSNR > 60 [dB], SSIM > 0.9, y BER < 1E-03, y se verifica visualmente la calidad de la imagen recibida.
4. Si lo anterior se cumple, se da por terminada la metodología, de lo contrario se regresa al paso 3.

UEP con códigos convolucionales.

1. Conforme a los resultados teóricos ideales discutidos en la sección anterior, aquí también se pretende que la imagen recibida presente los siguientes valores: a) PSNR=Inf [dB], b) BER=0, y c) SSIM=1. A partir de lo anterior, se busca un código convolucional con parámetros (R, K) en el esquema EEP, con distancia libre d_{free} , que cumpla con estos criterios, para un valor de razón señal a ruido E_b/N_0 de interés, previamente seleccionado.
2. La metodología establece que para comenzar el diseño de la codificación UEP en la imagen, se requiere conocer los valores (R, K) del código convolucional en EEP, que lograron alcanzar el criterio ideal de los valores de calidad, mismos que se toman en cuenta como referencia para proteger la capa binaria más significativa de la imagen.

Es conveniente aclarar que no necesariamente se debe aplicar el código convolucional en EEP encontrado para la capa binaria más significativa, sin embargo, se busca que sea lo más cercano posible a ese código.

3. Se emplean diferentes códigos convolucionales (R, K) (respetando lo establecido en el paso 2), para aplicarlos a cada capa binaria, considerando la cantidad de información en cada una de ellas, debido a que la capa que contenga mayor información, debe estar mejor protegida que las capas subsecuentes. Es importante la protección en cada capa binaria de la imagen, sin embargo, por un criterio de optimización, no se debe perder de vista que el esquema UEP debe satisfacer las siguientes condiciones: $R_{UEP} < R_{EEP}$, siendo R_{UEP} la tasa promedio resultante de las capas de la imagen en el esquema UEP, $K_{UEP} < K_{EEP}$, siendo K_{UEP} la longitud de restricción promedio resultante de las capas de la imagen, también en el esquema UEP, y que los valores de calidad (PSNR, BER, y SSIM) sean lo más cercano al caso ideal. De acuerdo a la experimentación realizada, se recomienda que por lo menos se alcancen los siguientes valores: PSNR >60 [dB], SSIM > 0.9, y BER < 1E-03, y verificar visualmente la calidad de la imagen recibida.
4. Si lo anterior se cumple, avanzar al paso 5, de lo contrario se regresa al paso 3.
5. Considerando la perforación (*punctured*), en los códigos convolucionales se permite reducir aún más la tasa de codificación, respetando el número de estados del diagrama de trellis de cada capa, empleando matrices de perforación a su salida mediante el concepto RCPC (*rate compatible punctured convolutional code*). Se recomienda utilizar matrices de perforación existentes en la literatura (generalmente con “códigos madre” de tasa 1/2 y 1/3) en las capas binarias menos significativas de la imagen, hasta lograr obtener los siguientes criterios: $R_{UEPpunc} < R_{UEP}$, donde $R_{UEPpunc}$ es la tasa de código después de la perforación, y dado que la perforación degrada el desempeño de un código, que los valores de calidad (PSNR, BER, y SSIM) no se degraden más allá del 10% de lo obtenido antes de la perforación, para no afectar la calidad visual de la imagen recibida.
6. Si lo anterior se cumple, se da por terminada la metodología, de lo contrario se regresa al paso 5.

UEP con códigos BCH y convolucionales en un escenario real.

Para ambos sistemas basados en codificación BCH y convolucional, la metodología que fue desarrollada para el escenario simulado, puede ser aplicada de igual forma para un escenario real, únicamente difiere en el paso 1. Donde se menciona que existe un valor de interés de la razón señal a ruido E_b/N_0 , entonces debido a la limitante sobre el control de la razón señal a ruido en este sistema de comunicaciones inalámbrico, para el caso real se consideran las pérdidas en el enlace, y el interés es sobre una cierta distancia entre el transmisor y el receptor.

Capítulo 4

Implementación y análisis de resultados en el entorno de simulación por computadora

Con base al sistema de comunicaciones descrito en la sección 3.1 de esta tesis, en este capítulo se presenta la implementación de dicho sistema sobre dos escenarios, uno basado en simulación computacional, y el otro en un canal inalámbrico real (descrito en el capítulo 5). Considerando imágenes digitales como fuente de información, el sistema de comunicaciones simulado fue realizado en lenguaje Matlab, para efectuar las etapas de descomposición, codificación y decodificación de canal. Mientras que, el sistema de comunicaciones real, se implementó con la herramienta de software GNU Radio, consistente en la etapa de transmisión/recepción de la imagen, mediante el uso de dos USRP2, con el propósito de aplicar, verificar, y validar la metodología propuesta en la sección 3.2.2. Esta metodología fue aplicada primero en el sistema simulado, para escoger los códigos BCH, o convolucionales, basados en el esquema EEP, y luego en UEP, para así calcular, y comparar el comportamiento de los valores de calidad que cuantifican el desempeño de cada esquema de protección. Esta comparación se hace con respecto al sistema sin protección contra errores. Y posteriormente, los resultados se adaptan al sistema real.

4.1 Selección y procesamiento de la imagen.

Para ilustrar el procesamiento de la imagen, en la Figura 4.1 se presenta la imagen llamada “Tux”, seleccionada para esta serie de pruebas con las siguientes características, está en escala de grises, de tamaño 256x256 píxeles y 8 bits de profundidad. Su procesamiento se describe a continuación:

- a. Se divide la imagen en 8 matrices (capas) binarias, las cuales se ordenan de mayor a menor significancia.
- b. Se convierten las matrices en vectores.
- c. Se completa con ceros (*padding*) cada vector para adecuar su tamaño a los parámetros del codificador que se pretenda emplear.

En la Figura 4.2, se ilustran las capas binarias obtenidas del procedimiento anterior. En el sistema de comunicaciones simulado, tanto el tratamiento de la imagen, descrito en el procedimiento anterior, como la implementación de la codificación/decodificación de canal, se realizan en un script de Matlab, guardando en variables los datos de la fuente (Imtx) y el destino (Imrx); la etapa de transmisión/recepción, fue realizada con la herramienta Simulink de Matlab, como se muestra en la Figura 4.3.

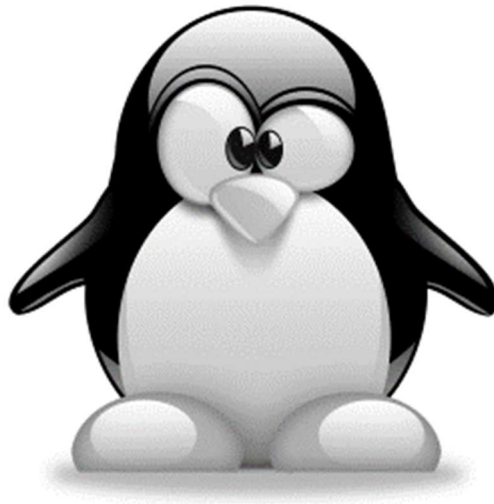


Figura 4.1: Imagen original “Tux”.

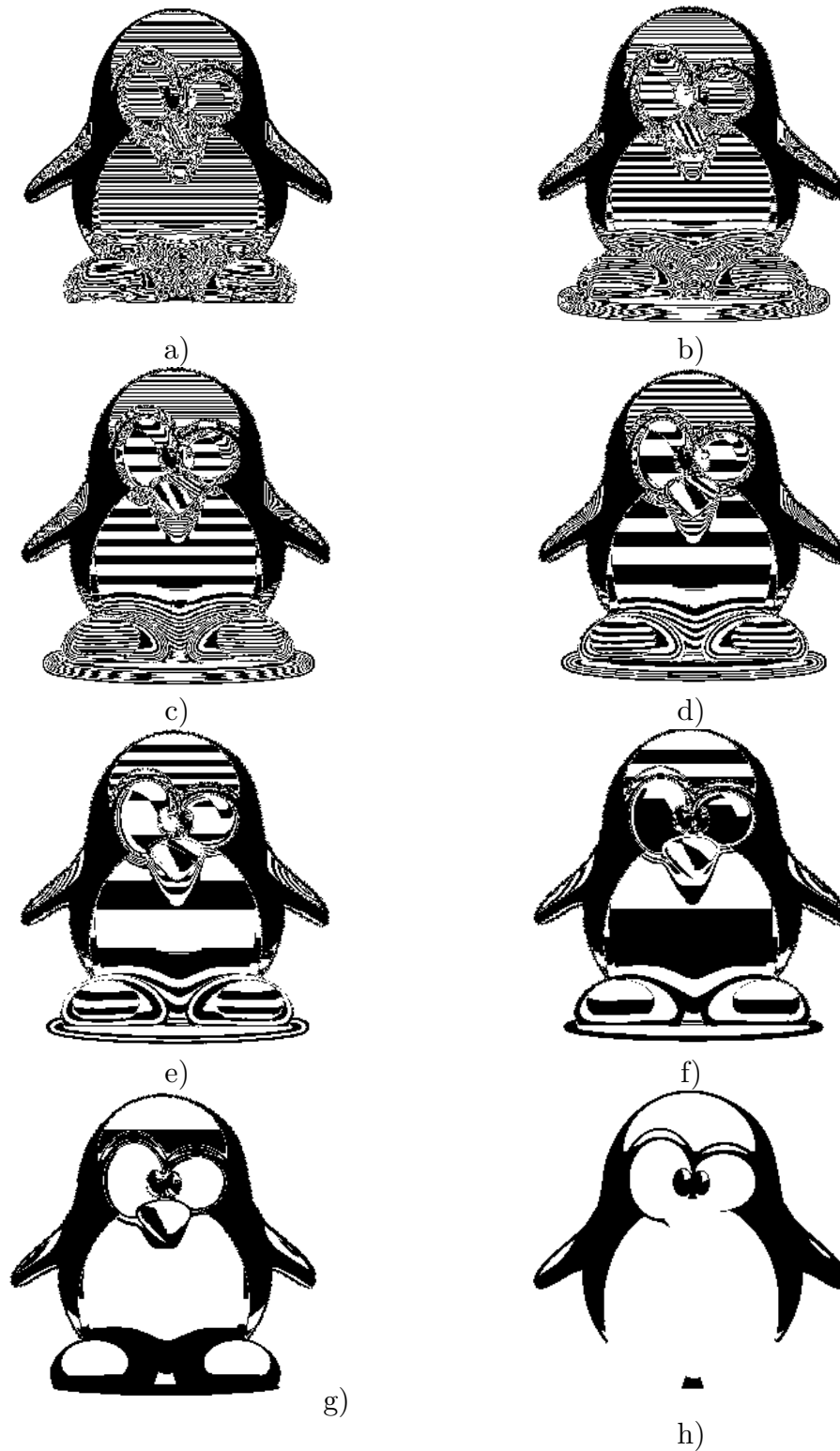
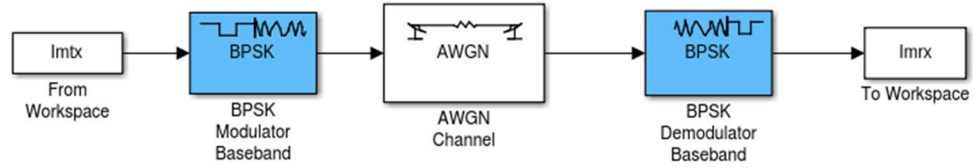


Figura 4.2: Imagen “Tux” dividida en 8 planos ordenados por su significancia: a) capa menos significativa. h) capa más significativa.



System over an AWGN channel
2016. Ana Rodríguez Armas @ UNAM

Figura 4.3: Sistema de comunicaciones sin codificación de canal en Simulink.

Para mostrar el efecto que causa el ruido en cada una de las capas binarias de la imagen, se realizaron diversas pruebas en el sistema de comunicaciones simulado sin codificación de canal, que se muestra en la Figura 4.3. Para esto, en cada capa binaria se agregaron errores en la transmisión de la imagen, mediante un canal AWGN con una razón señal a ruido $E_b/N_0 = 4$ [dB] ($BER \approx 10^{-3}$), dando por resultado las Figuras de la 4.4, a la 4.11.

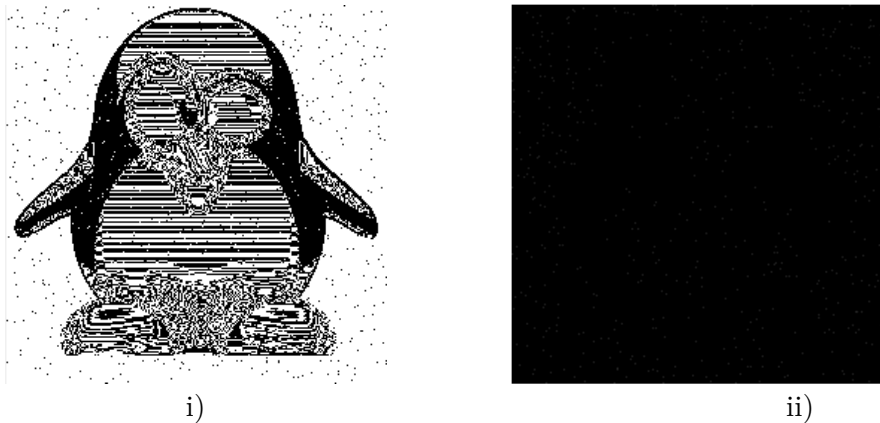


Figura 4.4: Imagen recibida con una $BER \approx 10^{-3}$. i) Capa 1 con una $E_b/N_0 = 4$ [dB]. ii) Errores en “Tux”.

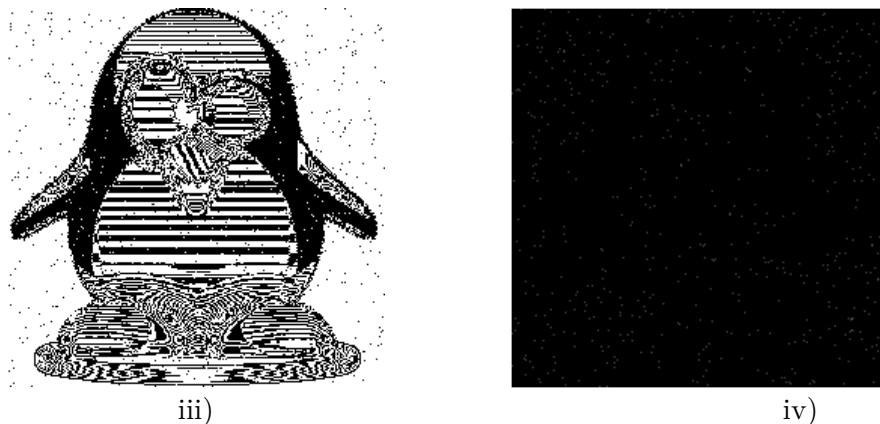


Figura 4.5: Imagen recibida con una $BER \approx 10^{-3}$. iii) Capa 2 con una $E_b/N_0 = 4$ [dB]. iv) Errores en “Tux”.

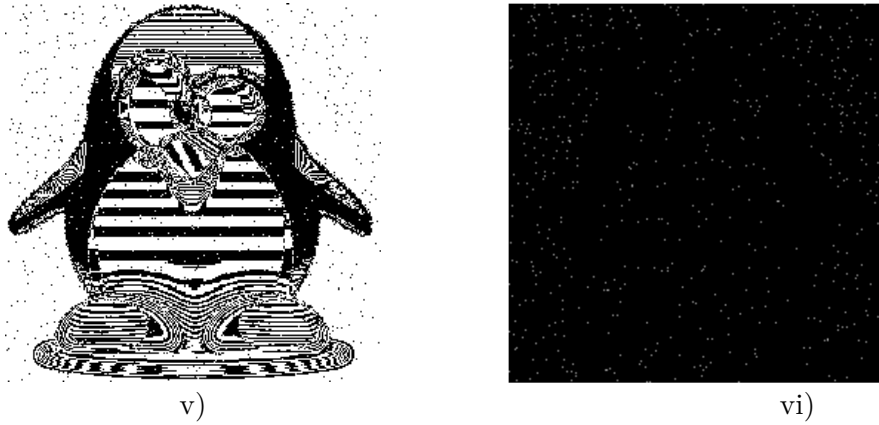


Figura 4.6: Imagen recibida con una BER $\approx 10^{-3}$. v) Capa 3 con una $E_b/N_0 = 4$ [dB]. vi) Errores en "Tux".

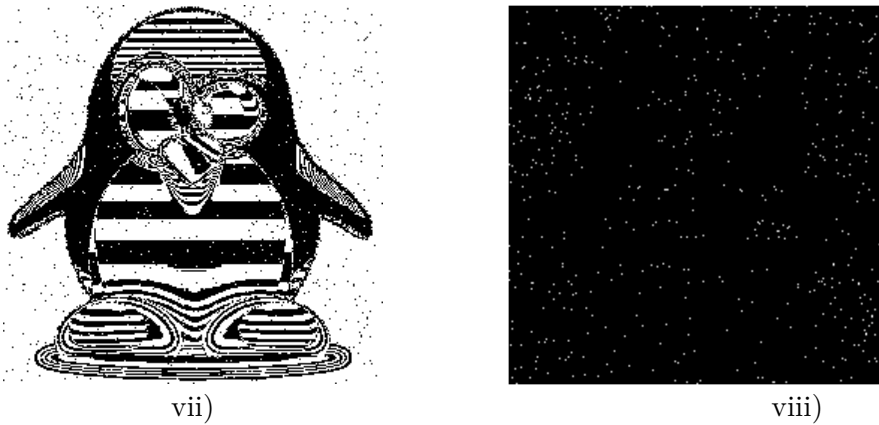


Figura 4.7: Imagen recibida con una BER $\approx 10^{-3}$. vii) Capa 4 con una $E_b/N_0 = 4$ [dB]. viii) Errores en "Tux".

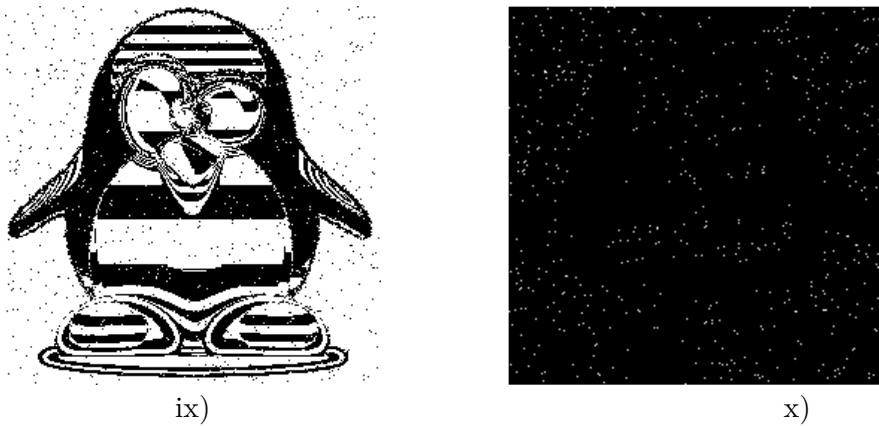


Figura 4.8: Imagen recibida con una BER $\approx 10^{-3}$. ix) Capa 5 con una $E_b/N_0 = 4$ [dB]. x) Errores en "Tux".

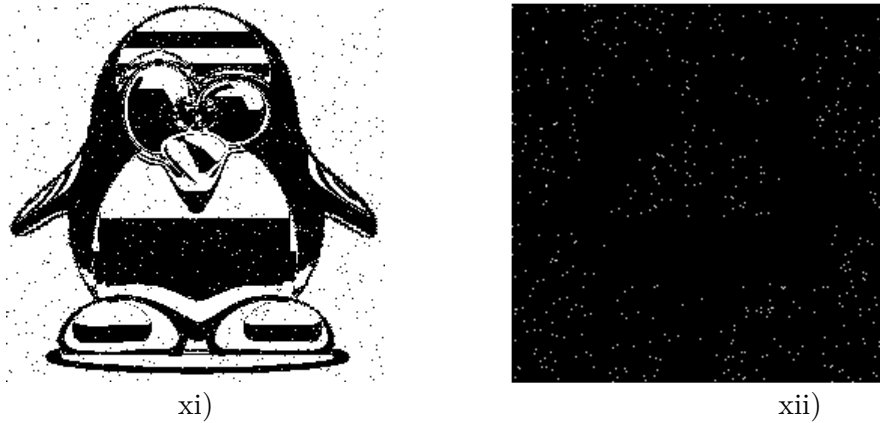


Figura 4.9: Imagen recibida con una BER $\approx 10^{-3}$. xi) Capa 6 con una $E_b/N_0 = 4$ [dB]. xii) Errores en “Tux”.

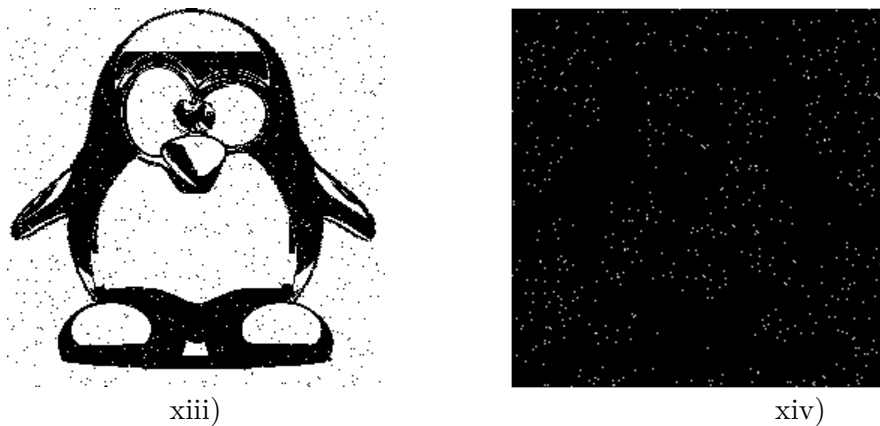


Figura 4.10: Imagen recibida con una BER $\approx 10^{-3}$. xiii) Capa 7 con una $E_b/N_0 = 4$ [dB]. xiv) Errores en “Tux”.

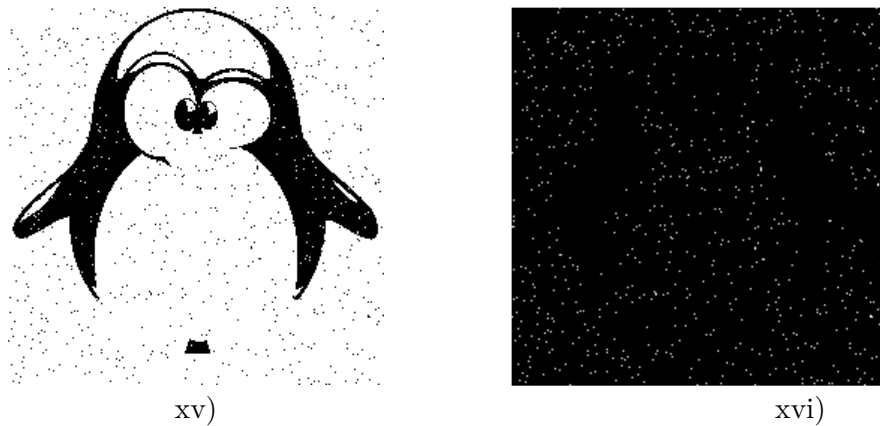


Figura 4.11: Imagen recibida con una BER $\approx 10^{-3}$. xv) Capa 8 con una $E_b/N_0 = 4$ [dB]. xvi) Errores en “Tux”.

Estas pruebas, Figuras de la 4.4 a la 4.11, evidencian el efecto progresivo del ruido en las diferentes capas binarias de la imagen, dependiendo de su peso relativo. Concluyendo que en el esquema UEP es necesario proteger contra el ruido utilizando un código con mayor capacidad de corrección en las capas más sensibles, permitiendo así aplicar la metodología descrita en el capítulo 3.

4.2 Pruebas y resultados del esquema UEP realizadas en el sistema simulado en computadora.

En el área de telecomunicaciones, Matlab es un software de alto desempeño, debido a sus innumerables funciones, y programas, dentro de un ambiente flexible para el usuario; como es sabido, puede ser usado para desarrollar algoritmos, adquisición de datos, modelado, obtención de gráficas, simulación, etc. Su mayor desventaja es el tiempo de procesamiento. Por lo anterior, se decidió utilizar Matlab para implementar el sistema de comunicaciones simulado en computadora, y probar así la metodología planteada en el capítulo 3, dando resultados que serán discutidos a lo largo de esta sección. En esta tesis se utilizó la versión de prueba de Matlab R2014b.

4.2.1 Simulación sin codificación, y con codificación para códigos BCH⁴, empleando los esquemas UEP- EEP

El proceso descrito en la metodología definida en la sección 3.2.2, se resume en el diagrama de flujo de la Figura 4.12, mismo que se adaptó al ambiente de software Matlab para desarrollar las pruebas correspondientes, empleando los códigos (de bloque) binarios BCH.

Para validar los resultados, se empleó la imagen “Tux”, y la razón señal a ruido de interés E_b/N_0 se fijó en 10 [dB], por considerarlo un valor medio en pruebas de recepción. A continuación, se utilizan unas tablas para dar los detalles, y comentar los resultados, de las pruebas.

⁴ Los presentes códigos BCH fueron consultados en [25].

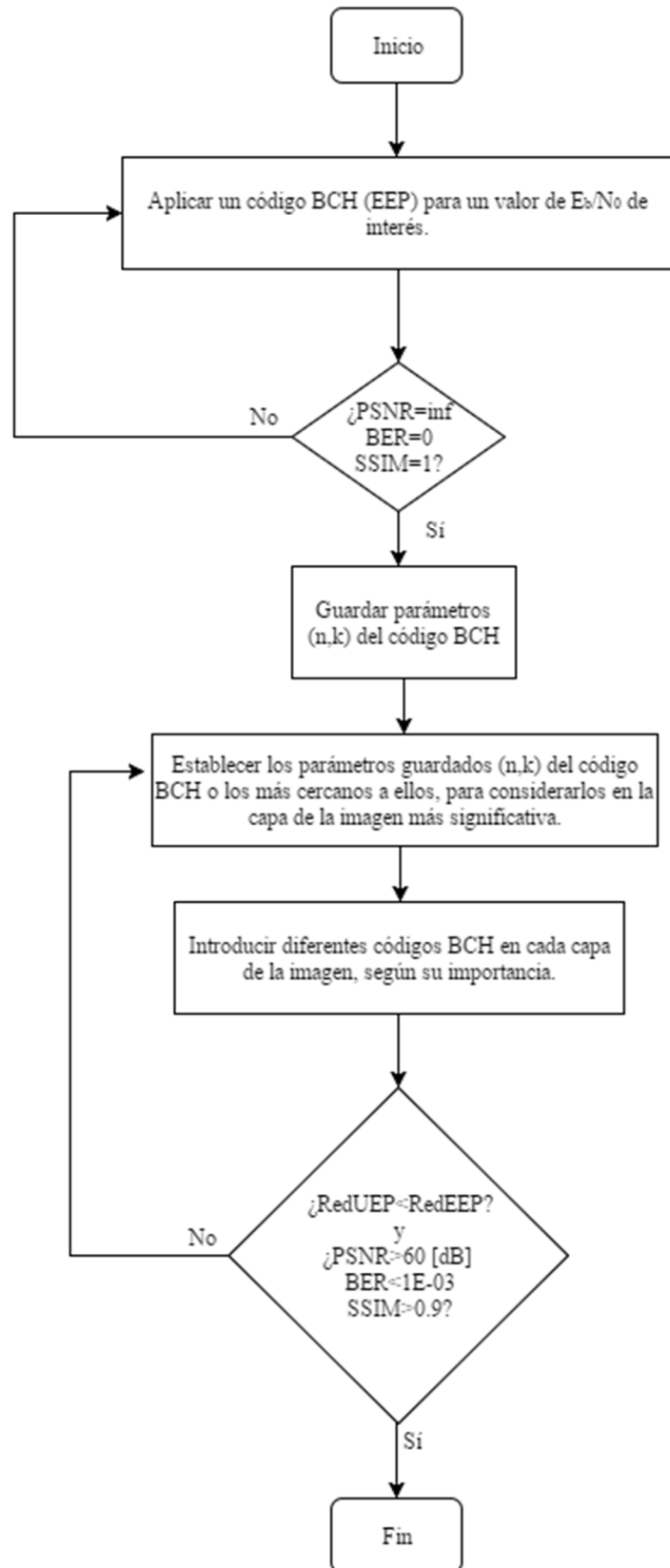


Figura 4.12: Diagrama de flujo adaptado al software Matlab de la metodología para códigos BCH presentada en el capítulo 3.

Etapa 1. Aplicación de un código BCH con la estrategia EEP, para un valor de E_b/N_0 de interés.

Prueba 1: Estrategia EEP, codificada con códigos BCH.

Eligiendo el código de parámetros (n, k) , corrector de t errores:

Capa	n	k	t	Redundancia
1-8	63	10	13	53

Tabla 4.1: Código BCH escogido para la prueba 1 empleando el esquema EEP.

Resultados de la prueba 1.

E_b/N_0 [dB]	BER_{EEP}	$SSIM_{EEP}$	$PSNR_{EEP}$ [dB]
1	0.2686682	0.0745333	26.732787
2	0.2303783	0.0884757	27.147306
3	0.1812785	0.1103098	27.855202
4	0.1243029	0.1451601	29.104516
5	0.0695852	0.2089647	31.247919
6	0.0291697	0.3833025	34.716515
7	0.0077705	0.7247947	40.204769
8	0.0011363	0.9530952	48.505876
9	9.19E-05	0.9967191	61.354773
10	0	1	Inf

Tabla 4.2: Valores de calidad resultantes de la prueba 1 con un código BCH para diferentes E_b/N_0 para la imagen "Tux".

De la tabla 4.2, se obtuvieron los valores ideales de calidad BER, SSIM y PSNR para la E_b/N_0 de 10 [dB] deseada. Únicamente para probar la convergencia del estudio se hicieron medidas con otro código BCH, con parámetros cercanos al anterior, para obtener los valores de calidad, para una E_b/N_0 de preferencia menor a 10 [dB], sin embargo, en comparación con la prueba anterior, se incrementó la redundancia, lo cual penalizaría el ancho de banda. Por consiguiente, se elige el código BCH empleado en la prueba 1, anterior.

Etapa 2. Valores de calidad deseados ¿PSNR=inf, BER=0, y SSIM=1?

De acuerdo a la metodología, dado que sí se obtuvieron los valores de calidad deseados, para la razón señal a ruido fijada de $E_b/N_0=10$ [dB] en la **prueba 1**. Se continúa con la siguiente etapa.

Etapa 3. Guardar los parámetros del código BCH (n, k) , corrector de t errores.

Se guardan los parámetros del código BCH escogido: $n=63$, $k=10$, corrector de $t=13$ errores.

Etapa 4. De acuerdo a la estrategia UEP, se utiliza el código BCH (n,k), corrector de t errores guardado, para codificar la capa binaria más significativa de la imagen digital. Para la capa más significativa de la imagen de prueba “Tux”, se escoge como primera opción, el código BCH con los parámetros (n, k), corrector de t errores, guardado en la Etapa 3 del método.

Etapa 5. Usar diferentes parámetros (n, k), t, de códigos BCH, según la importancia de calidad, para cada capa binaria de la imagen.

La estrategia consiste en utilizar los códigos que fueron seleccionados en función del número de errores t, que pueden corregir.

Las pruebas de la presente etapa se realizaron para un rango de valores de E_b/N_0 que va desde 1, hasta 19 [dB]. El criterio de término para una $E_b/N_0 = 19$ [dB], se debe a que para ese valor se obtuvieron los valores ideales de los parámetros de calidad de la imagen, para el caso en el que la imagen fue transmitida sin codificación. Los resultados se presentan mediante gráficas, y tiempos de ejecución del sistema.

Cabe resaltar que en este escrito solo se exponen las pruebas más significativas, que indicaron el resultado óptimo del diseño UEP en la imagen de prueba.

Prueba 2: Comparación completa: sin codificación.vs.EEP.vs.UEP BCH.

Para esta prueba se emplearon los códigos BCH con los parámetros dados en la Tabla 4.3, para la capa binaria que se indica.

Capa	N	k	t	Redundancia
1	15	5	3	10
2	15	5	3	10
3	31	6	7	25
4	31	6	7	25
5	31	6	7	25
6	63	10	13	53
7	63	10	13	53
8	63	10	13	53
			Promedio	32
			Redundancia	

Tabla 4.3: Códigos BCH para la prueba 4 empleando el esquema UEP.

Resultados de la prueba 2.

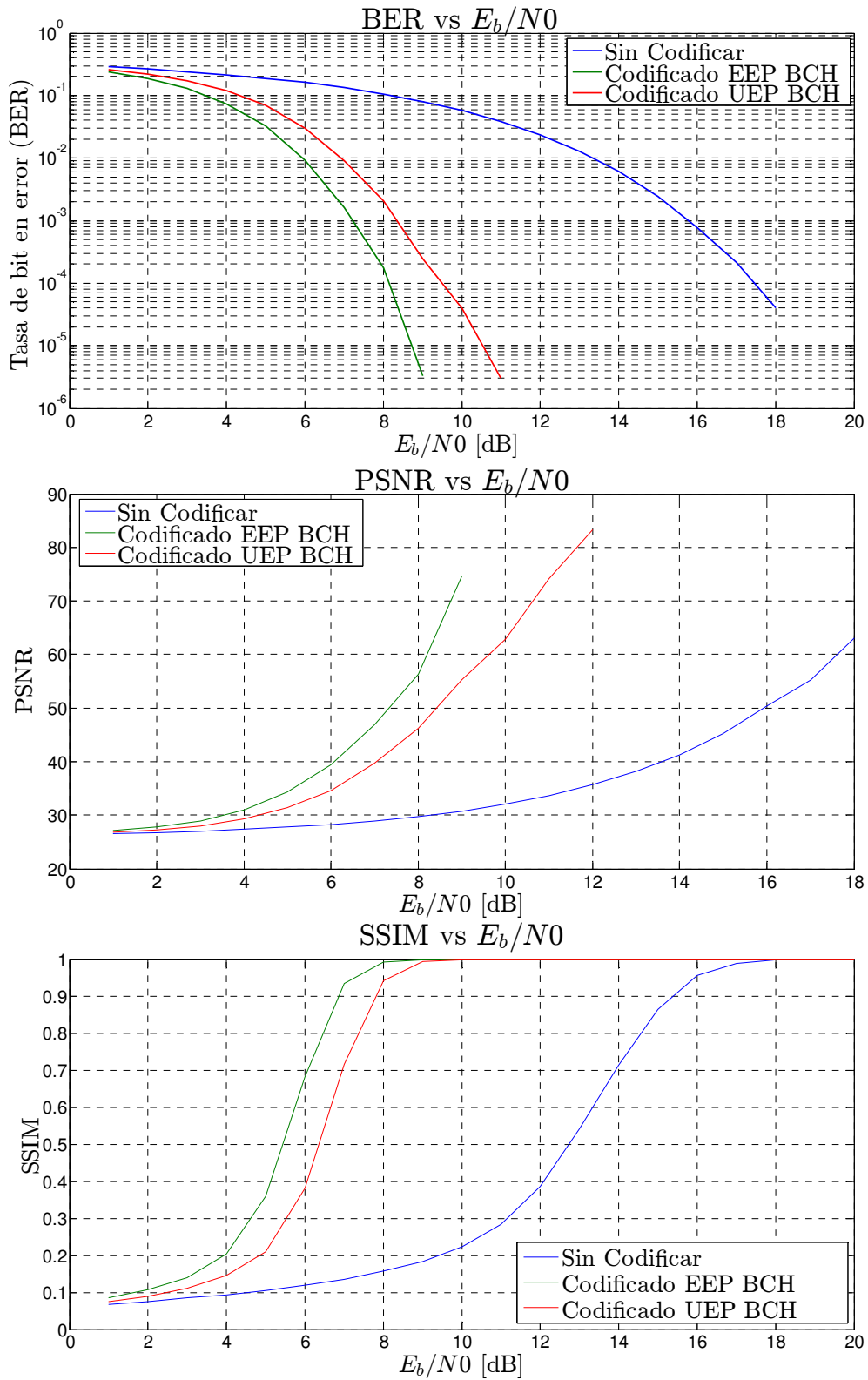


Figura 4.13: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 2, obtenidos para la imagen "Tux".

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	26.99
BCH EEP	243.01
BCH UEP	129.92

Tabla 4.4: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 2.

De la Figura 4.13, para una razón deseada $E_b/N_0 = 10$ [dB] se obtuvieron los valores de calidad: PSNR = 62.789 [dB], SSIM = 0.9995 y BER = 1.54E-04, lo cual cumple con lo establecido en la metodología; además se puede comparar el esquema UEP con respecto al esquema EEP, ambos mejoran su desempeño con respecto al resultado que se obtuvo sin aplicar codificación. Además, aunque la calidad del esquema UEP es ligeramente menor que la que se obtiene con el esquema EEP, la redundancia generada con este último es menor en un 40 %.

En la Tabla 4.4 se dan los tiempos de procesamiento computacional usando el software Matlab, en una computadora personal de 1.6 GHz, con sistema operativo Ubuntu 14.04, y memoria de 6 GB. Se comprueba que el esquema UEP mejoró el tiempo de ejecución con respecto al esquema EEP, permitiendo una reducción de hasta un 47%.

Con el propósito de mejorar los valores de calidad BER, SSIM y PSNR, manteniendo la misma razón $E_b/N_0 = 10$ [dB], se llevará a cabo otro experimento, manteniendo aproximadamente la misma redundancia promedio del sistema. Dando la siguiente prueba:

Prueba 3: Comparación completa: sin codificación.Vs.EEP.Vs.UEP BCH.

De acuerdo al objetivo fijado, para las capas binarias 7 y 8, se busca un código con parámetros (n, k) , (63,7) que corrija más errores, para aumentar su protección.

Capa	n	k	t	Redundancia
1	15	5	3	10
2	31	11	5	20
3	31	6	7	25
4	63	18	10	45
5	63	10	13	53
6	63	10	13	53
7	63	7	15	56
8	63	7	15	56
Promedio				39

Tabla 4.5: Códigos BCH para la prueba 3 empleando el esquema UEP.

Resultados de la prueba 3.

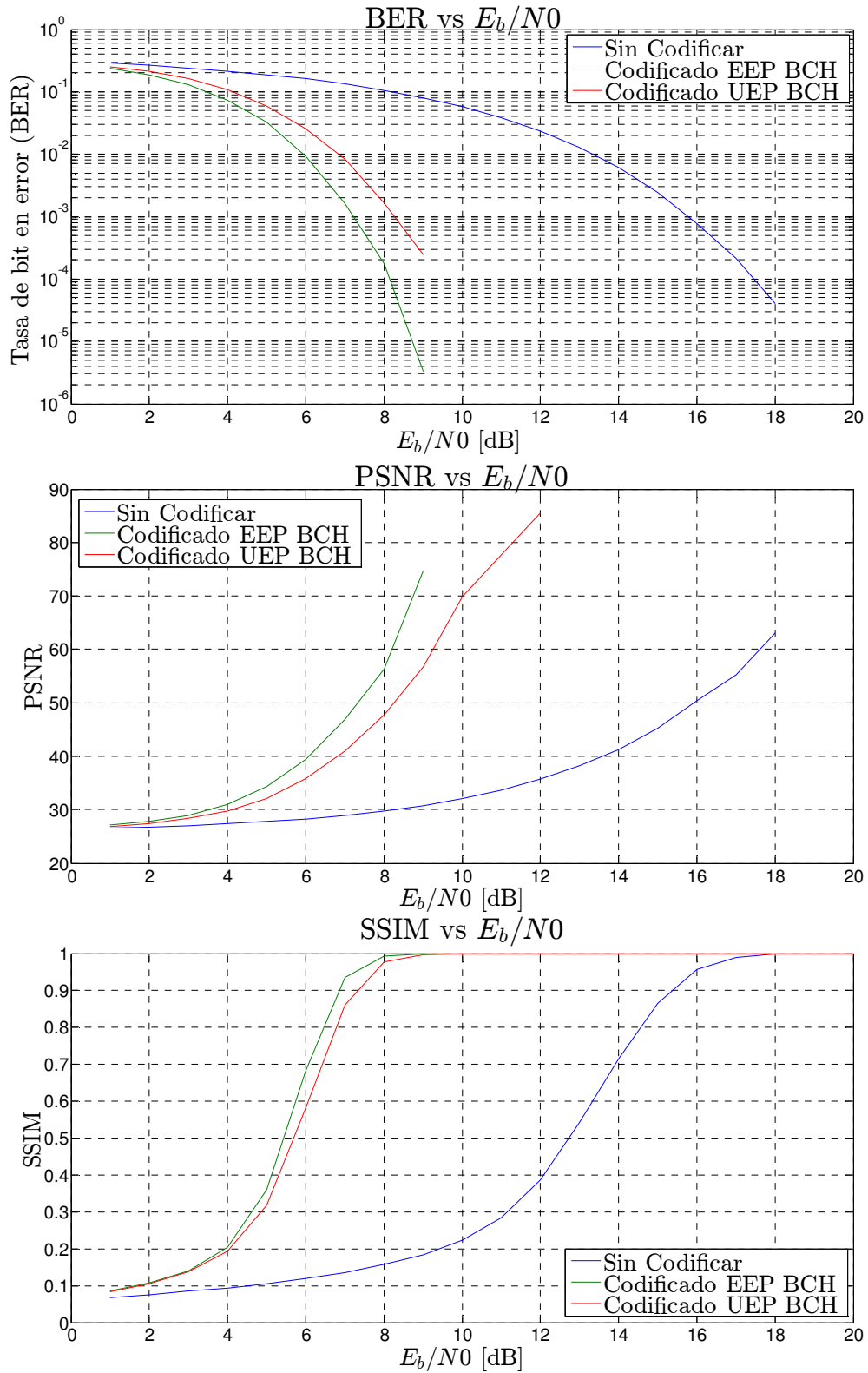


Figura 4.14: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 3, obtenidos para la imagen “Tux”.

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	26.99
BCH EEP	243.01
BCH UEP	183.28

Tabla 4.6: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 3.

En la Figura 4.14, se lograron mejorar los valores de: PSNR = 69.935 [dB], SSIM = 0.9998, y BER = 1.01E-04 para una misma $E_b/N_0 = 10$ [dB]. Esto nos permite validar el buen funcionamiento de la metodología propuesta, agregando que con respecto a la prueba 2, se pueden apreciar que las curvas del esquema UEP tienen un comportamiento más cercano a las curvas de la codificación EEP, y ambas están todavía más alejadas de la curva correspondiente al caso sin aplicar codificación. Además, la redundancia con UEP disminuyó en un 26 % con respecto a la redundancia con EEP.

En la Tabla 4.6, se dan los tiempos de procesamiento computacional con las mismas características del equipo de cómputo mencionado anteriormente. Se comprueba que el esquema UEP mejoró el tiempo de ejecución con respecto al esquema EEP, permitiendo una reducción de hasta un 24%.

Buscando acercar más el comportamiento del esquema UEP respecto al esquema EEP, mejorando los valores de calidad, para una misma razón $E_b/N_0 = 10$ [dB], respetando la condición de la redundancia en el esquema UEP, la cual debe ser menor que la obtenida en el esquema EEP. Se realizó este nuevo experimento:

Prueba 4: Comparación completa: sin codificación.vs.EEP.vs.UEP BCH.

Capa	n	k	t	Redundancia
1	15	5	3	10
2	31	6	7	25
3	31	6	7	25
4	63	10	13	53
5	63	10	13	53
6	63	10	13	53
7	63	7	15	56
8	63	7	15	56
Promedio				41

Tabla 4.7: Códigos BCH para la prueba 4 empleando el esquema UEP.

Se obtuvieron los siguientes resultados de la prueba 4.

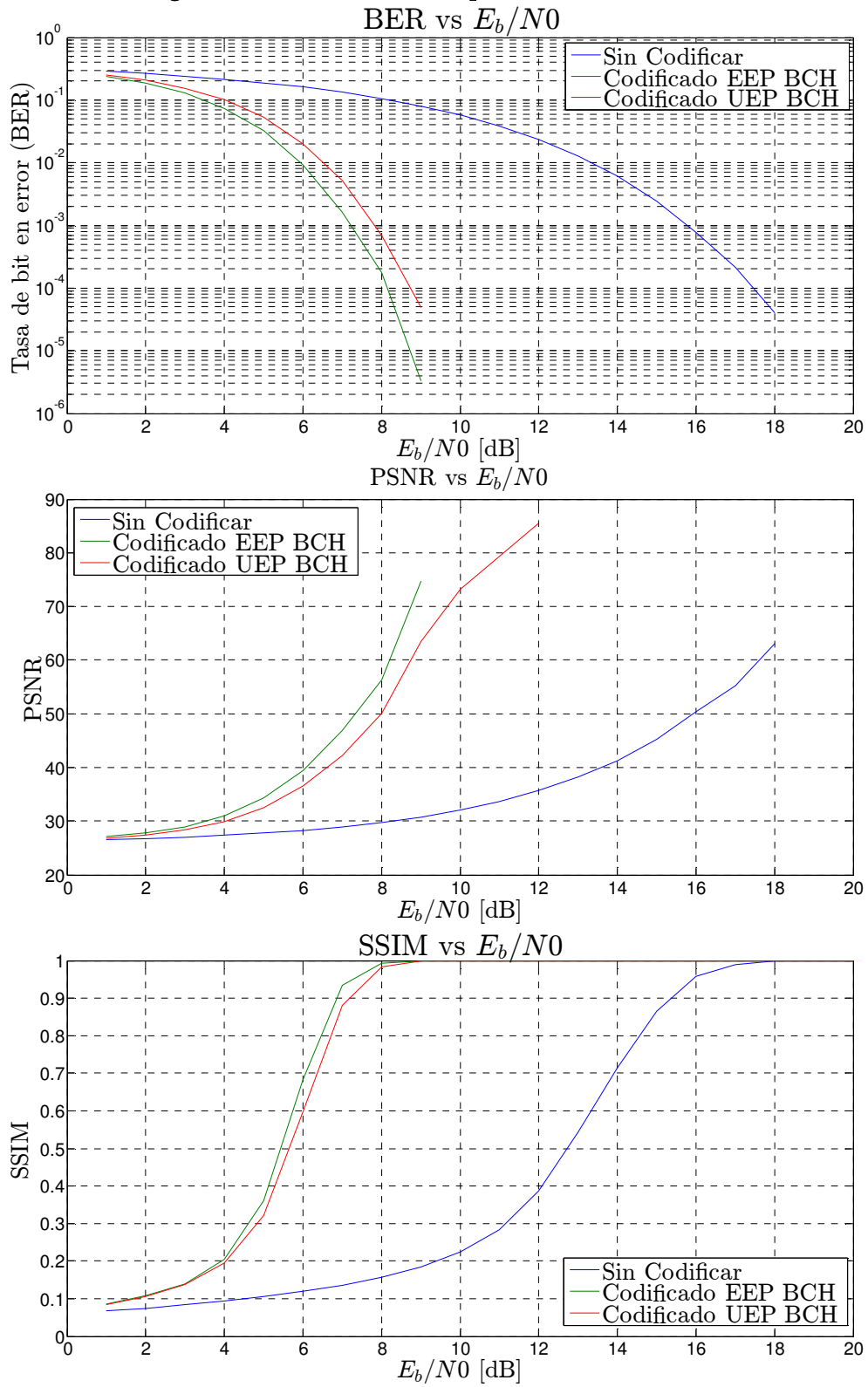


Figura 4.15: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 4, obtenidos para la imagen "Tux".

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	26.99
BCH EEP	243.01
BCH UEP	200.19

Tabla 4.8: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 4.

De la Figura 4.15, para una $E_b/N_0 = 10$ [dB] se tienen los valores de: PSNR = 73.233 [dB], SSIM = 0.9999 y BER = 5.15E-05. Con estos resultados se vuelve a validar la metodología propuesta, y se obtiene un mejor desempeño con respecto a la prueba 3., además, se puede apreciar que las curvas del esquema UEP son mucho más cercanas de las correspondientes al esquema EEP, por lo que la redundancia del esquema UEP disminuyó en un 23 %, con respecto a la redundancia del esquema EEP.

En la Tabla 4.8, con las mismas características del equipo de cómputo mencionado anteriormente, se muestra el tiempo de procesamiento computacional en Matlab, donde aún existe un ahorro del tiempo de ejecución del esquema UEP con respecto al del esquema EEP, cercano a 17 %. Sin embargo, se puede comentar que el ahorro no es tan considerable como el que se obtuvo en las pruebas 2 y 3. Hay un comportamiento asintótico.

En la Tabla 4.9, se puede observar el resumen de los resultados más representativos de las pruebas 2, 3, y 4 del estudio realizado.

N° de prueba	Prueba 2	Prueba 3	Prueba 4
E_b/N_0 [dB]	10	10	10
Redundancia	32	39	41
PSNR _{UEP} [dB]	62.789	69.935	73.233
SSIM _{UEP}	0.9995	0.9998	0.9999
BER _{UEP}	1.54E-04	1.01E-04	5.15E-05
Tiempo de ejecución [s]	129.92	183.28	200.19

Tabla 4.9: Concentrado de resultados con códigos BCH, empleando el esquema UEP.

Para ilustrar los resultados, en la Figuras 4.13, 4.14 y 4.15, se presentan las imágenes recibidas con su respectiva imagen de errores para una $E_b/N_0 = 10$ [dB], estas últimas amplificadas con un factor de 40 para que los errores sean visibles. Estas imágenes, fueron obtenidas de las diferentes pruebas realizadas, con el propósito de compararlas de manera visual, para finalmente elegir los códigos BCH que satisfagan lo estipulado en la metodología de diseño, respetando la calidad de la imagen recuperada en términos de los valores de calidad, y ahorro de tiempo de ejecución en las simulaciones por computadora.



1.a)



1.b)



2.a)



2.b)



3.a)



3.b)

Figura 4.16: Imágenes “Tux” resultantes y de error de las pruebas 2, 3 y 4.

Etapa 6. Análisis de la calidad, ¿RedUEP<RedEEP? y ¿PSNR>60 [dB] BER<1E-03 SSIM>0.9?

Dados los resultados numéricos, y perceptuales de las pruebas 2, 3 y 4, las tres pruebas cumplen con la condición de la Etapa 6. Sin embargo, para comenzar a descartar pruebas, es conveniente evaluar los resultados con respecto a los recursos de cómputo disponibles y la calidad de la imagen en cada prueba.

4.2.1.1 Análisis de resultados de la experimentación con los códigos BCH, con la estrategia UEP.

En esta sección, se presenta un análisis de resultados, conforme a las pruebas realizadas en la sección 4.2.1.

Análisis visual: Para la prueba 3, visualmente la calidad de la imagen no se vio afectada por el ruido del canal como en la prueba 2. La prueba 4, fue la menos afectada de las tres pruebas; por lo tanto, las pruebas 3 y 4 fueron las más aceptables a partir de esta observación.

Análisis de desempeño computacional: se corroboró que el ahorro en el tiempo de ejecución de la prueba 2 fue mucho mayor con respecto a las pruebas 3 y 4, obteniendo respectivamente los porcentajes de 47%, 24% y 17%.

Subrayando que para el esquema UEP empleado en la imagen, se busca ahorrar ancho de banda con respecto al esquema EEP, y que la calidad de la imagen recibida sea lo más aproximado a la imagen original. Conforme a los valores obtenidos de los valores de calidad, a continuación, se presenta la conclusión de dicho análisis.

- La **prueba 2 BCH** mostró un gran desempeño computacional, al ahorrar mucho más del 25% de tiempo de ejecución del sistema; sin embargo, visualmente fue la más deficiente, al arrojar valores bajos de calidad con respecto a las pruebas posteriores.
- La **prueba 3 BCH** mostró un desempeño computacional bueno, al ahorrar el 24% de tiempo de ejecución del sistema; los valores de los parámetros de calidad fueron superiores, ya que mejoró su desempeño con respecto a la prueba 2, y respecto al resultado visual, es aceptable, dado que el ojo humano difícilmente detecta los errores en la imagen recibida.
- La **prueba 4 BCH** mostró un desempeño computacional bajo, al ahorrar el 17% de tiempo de ejecución del sistema, aunque en calidad visual fue la más alta de las tres pruebas.

Finalmente, la prueba que cumplió con todos los requerimientos de la metodología propuesta en la sección 3.2.2, fue la **prueba 3**. Dado que conserva la calidad de la imagen recibida a nivel visual, y cuantitativamente, sin sacrificar de más los tiempos de ejecución del sistema.

Una vez hecho este estudio con una imagen, se decidió corroborar el comportamiento del diseño del esquema de codificación con otra imagen de características texturales diferentes. Para lo cual se retomaron los códigos empleados en la Tabla 4.5 de la **prueba 3, con códigos BCH**, para la imagen “Lena”, obteniendo los siguientes resultados.

Resultados de la imagen “Lena”.

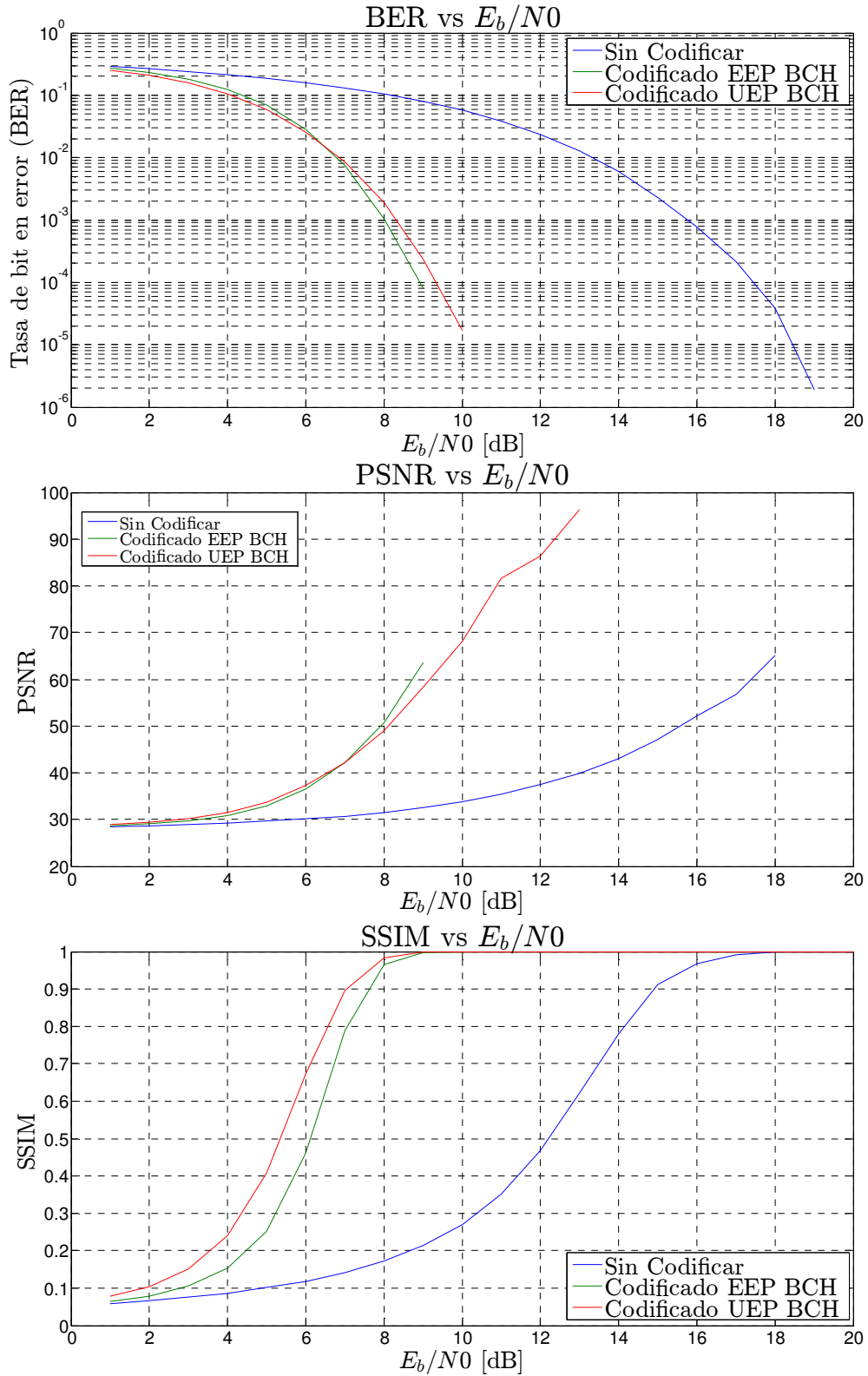


Figura 4.17: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación BCH de la prueba 4, obtenidos para la imagen “Lena”.

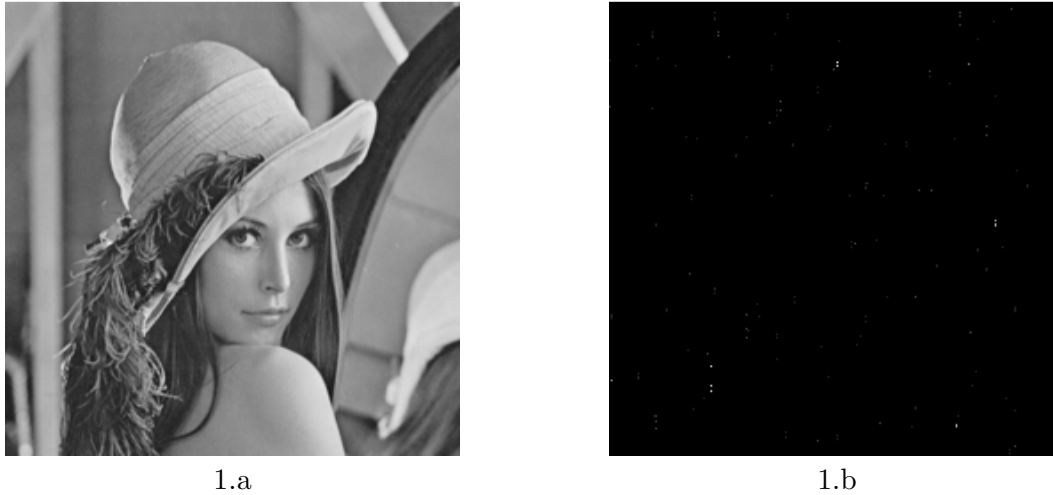


Figura 4.18: “Lena” resultante de la prueba 3, con códigos BCH. 1) Imagen recibida. 2) Errores de la imagen recibida.

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	20.39
BCH EEP	177.86
BCH UEP	131.35

Tabla 4.10: Tiempos de ejecución para distintos esquemas con codificación BCH de la prueba 3, obtenidos de “Lena”.

Imagen	“Tux”	“Lena”
E_b/N_0 [dB]	10	10
Redundancia	39	39
$PSNR_{UEP}$ [dB]	69.935	68.088
$SSIM_{UEP}$	0.9998	0.9998
BER_{UEP}	1.01E-04	1.11E-4
% Ahorro en ejecución	24%]	26%

Tabla 4.11: Comparación de los resultados de las imágenes “Tux” y “Lena” empleando códigos BCH con UEP.

Como se puede concluir, los códigos BCH de la prueba 3 causaron casi los mismos efectos en ambas imágenes para una $E_b/N_0 = 10$ [dB]. Por lo que la metodología propuesta genera buenos resultados de codificación, independientemente del contenido textural de las imágenes digitales.

4.2.2 Simulación sin codificación y con codificación con códigos convolucionales⁵ empleando los esquemas UEP- EEP

En teoría de códigos correctores de errores además de los códigos de bloque, ya utilizados en esta tesis, se tienen los códigos convolucionales, también de amplio uso en los sistemas de comunicaciones digitales. Con el propósito de probar su eficacia, y hacer de nuestro estudio algo más completo, se adaptó a este caso el proceso propuesto en la metodología de la sección 3.2.2, y se resume en el diagrama de flujo de la Figura 4.19.

Para hacer comparables las pruebas también se usó la imagen de “Tux”. Los datos de entrada que requiere Matlab para usar el algoritmo de los códigos convolucionales predeterminado son: la tasa del código (R), la longitud de restricción (K) y los polinomios generadores del código (Code), expresados en octal.

⁵ Los presentes códigos convolucionales fueron consultados en [51] [52].

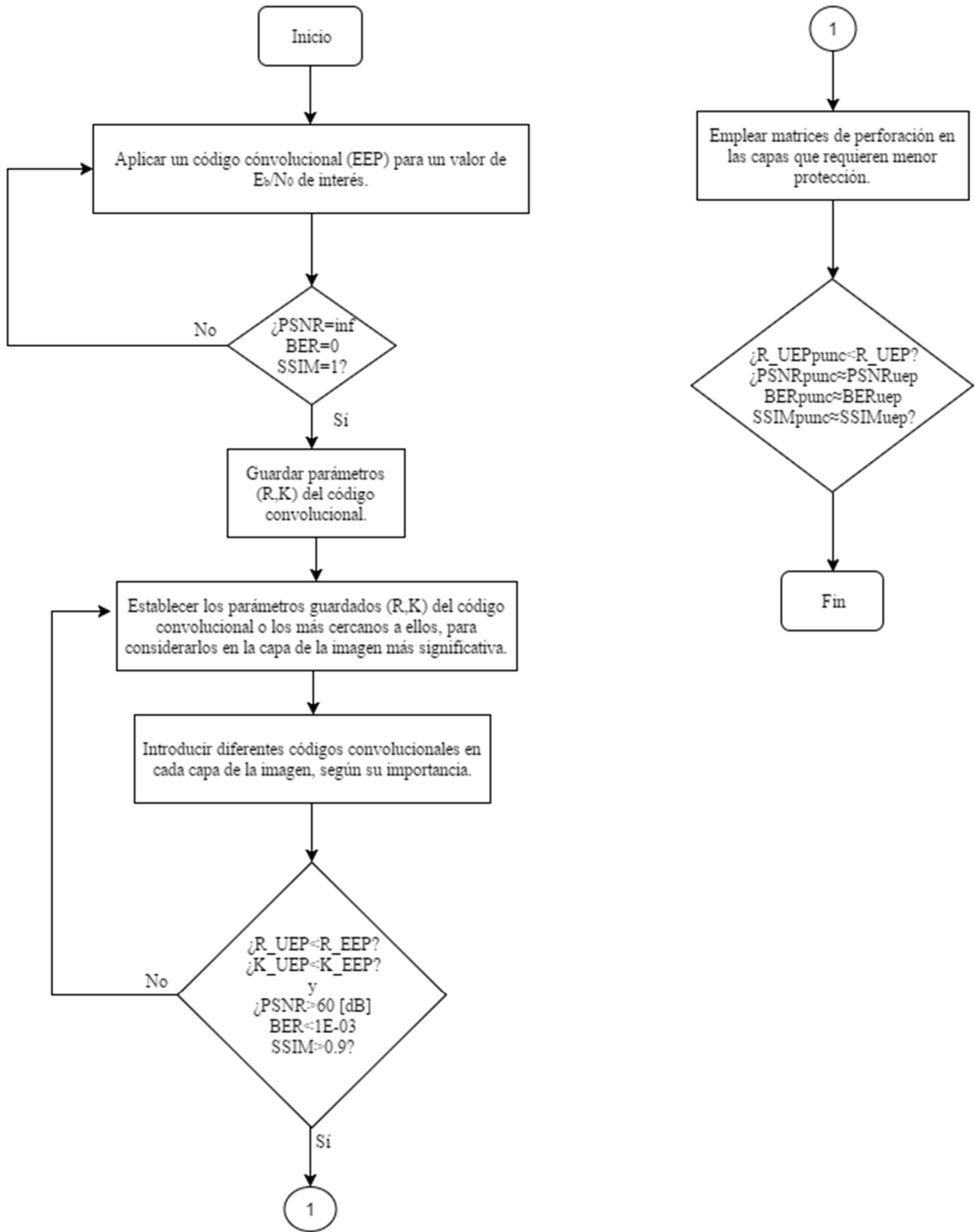


Figura 4.19: Diagrama de flujo adaptado de la metodología propuesta en el capítulo 3 con códigos convolucionales.

Siguiendo las etapas de la Figura 4.19, se tiene:

Etapla 1. Aplicar diversos códigos convolucionales con el esquema EEP para distintos valores de razón señal a ruido E_b/N_0 .

Prueba 1: EEP Convolutacional.

Capa	R	K	d_f	t	Code
1-8	1/6	8	34	16	253 375 331 235 313 357

Tabla 4.12: Código convolutacional escogido para la prueba 1 empleando el esquema EEP.

Resultados de la prueba 1.

E_b/N_0 [dB]	BER_{EEP}	$SSIM_{EEP}$	$PSNR_{EEP}$ [dB]
1	0.3068433	0.0631	26.34074126
2	0.1805226	0.1172	27.75354328
3	0.0678335	0.2472	31.1234953
4	0.0151745	0.6348	37.12032374
5	0.0017128	0.9410	46.56169849
6	0.0001827	0.9946	55.26939042
7	1.89E-05	0.9996	65.27449676
8	5.45E-06	0.9997	71.93627167
9	1.53E-06	0.9999	85.53778615
10	0	1	Inf

Tabla 4.13: Valores de calidad resultantes de la prueba 1 con un código convolutacional para diferentes E_b/N_0 obtenidos de con la imagen "Tux".

De la Tabla 4.13, se tienen los valores ideales de BER, SSIM y PSNR para una E_b/N_0 de 10 [dB]. A manera de validación, se probaron con otros códigos convolucionales cercanos a este, con el propósito de obtener el valor ideal de las medidas de calidad con una E_b/N_0 menor que 10 [dB], lo cual no fue posible, debido a que se volvieron a obtener los valores ideales de BER, SSIM y PSNR para una E_b/N_0 de 13 [dB], cuyo objetivo de esta prueba era encontrar un código convolutacional que al aumentar la distancia libre (d_f), corrigiera por tanto más errores, y que se lograra obtener los valores de calidad ideales para una E_b/N_0 menor que 10 [dB]. Por lo tanto, se elige el código de la prueba 1 como resultado concluyente y se pasa a la etapa siguiente.

Etapa 2. Valores de calidad deseados: ¿PSNR=inf, BER=0, y SSIM=1?

De acuerdo a la metodología, dado que sí se obtuvieron los valores de calidad deseados, para la razón señal a ruido fijada de $E_b/N_0=10$ [dB] en la **prueba 1**. Se continúa con la siguiente etapa.

Etapa 3. Guardar los parámetros del código convolucional (R, K).

Dado que la prueba 1 resultó concluyente, se guardan los valores de la Tabla 4.13.

Etapa 4. Escoger los parámetros del código convolucional guardado (R, K), o los más cercanos a ellos (en cuanto a su distancia libre), para considerarlos en la capa binaria más significativa de la imagen.

Para la capa binaria más significativa de la imagen “Tux”, se escogerá como primera opción, los parámetros guardados en la Etapa 3.

Etapa 5. Para lograr el esquema UEP, se escogen diferentes códigos convolucionales en cada capa binaria de la imagen, según su importancia.

Los códigos fueron seleccionados de acuerdo a la distancia libre, y por ende al número de errores que pueden corregir.

En las pruebas de la presente etapa, se dan los valores para diversas E_b/N_0 , yendo desde 1 hasta 19 [dB], debido a que para una $E_b/N_0 = 19$ [dB], se obtuvieron los valores buscados de calidad de la imagen. Se compara con el caso en el que la imagen fue transmitida sin codificación; y se presentan los resultados mediante gráficas, y tiempos de ejecución del sistema. Con propósitos de síntesis, solo se exponen las pruebas más significativas que dieron el mejor resultado del diseño UEP en la imagen.

Prueba 2: Comparación completa entre el caso sin codificación vs. EEP vs. UEP, para códigos convolucionales.

Capa	R	K	d_f	t	Code
1	1/2	7	10	4	133 171
2	1/3	8	16	7	225 331 367
3	1/3	8	16	7	225 331 367
4	1/4	5	16	7	25 27 33 37
5	1/4	5	16	7	25 27 33 37
6	1/5	7	25	12	175 131 135 135 147
7	1/5	8	28	13	257 233 323 271 357
8	1/6	8	34	16	253 375 331 235 313 357
Promedio	0.279	7			

Tabla 4.14: Código convolucional escogido para la prueba 2 empleando el esquema UEP.

Resultados de la prueba 2.

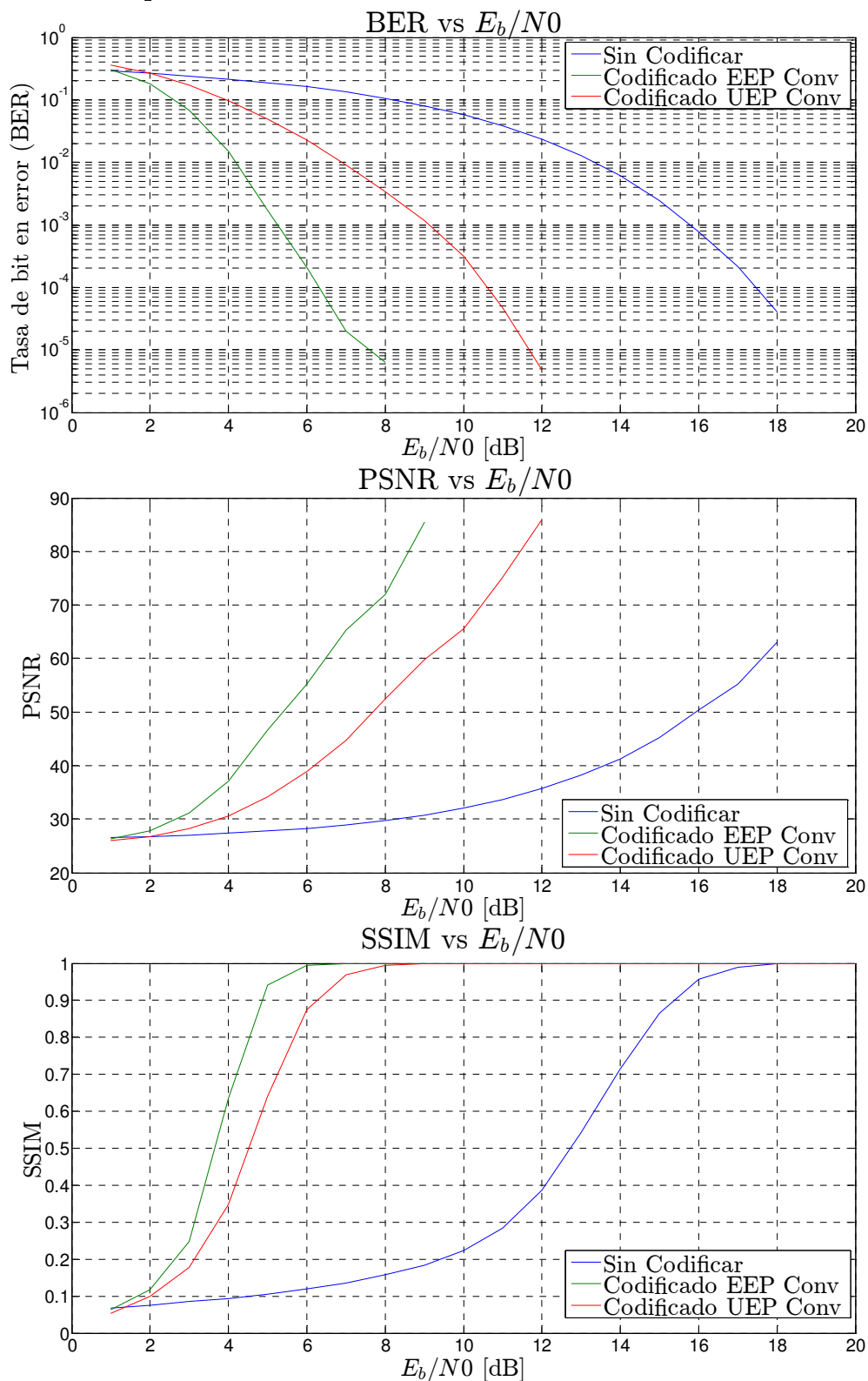


Figura 4.20: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional de la prueba 2, obtenidos para la imagen "Tux".

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	21.18
Convolutacional EEP	58.64
Convolutacional UEP	46.51

Tabla 4.15: Tiempos de ejecución para distintos esquemas con codificación convolutacional de la prueba 2.

Con estos resultados, de la Figura 4.20, para una $E_b/N_0 = 10$ [dB] se obtuvo: PSNR = 65.549 [dB], SSIM = 0.9997 y BER = 3.11E-04. Lo cual cumple con lo establecido en la metodología. Al igual que para los códigos BCH, de la Figura 4.20, se puede apreciar la progresión del esquema UEP hacia el esquema EEP, mejorando también su desempeño con respecto al caso sin aplicar codificación. Además, la redundancia con UEP disminuyó en un 40 % con respecto a la utilizada con EEP, mientras que la longitud de restricción promedio fue de 7. En la Tabla 4.15 se dan los tiempos del procesamiento computacional en Matlab, utilizando una computadora personal con procesador Intel Core i5 @ 1.60 GHz, con sistema operativo Windows 8.1, y memoria de 6 GB, donde se comprueba que el esquema UEP mejoró con respecto al esquema EEP, con un ahorro de hasta un 20% en tiempo de ejecución.

Para tratar de mejorar la calidad de la imagen en $E_b/N_0 = 10$ [dB], se hará otro experimento, buscando aumentar los valores de calidad sin afectar la redundancia del sistema.

Prueba 3: Comparación completa entre el caso sin codificación.vs.EEP.vs.UEP para códigos convolucionales.

Capa	R	K	d_f	t	Code
1	1/3	8	16	7	225 331 367
2	1/3	8	16	7	225 331 367
3	1/3	8	16	7	225 331 367
4	1/3	8	16	7	225 331 367
5	1/3	8	16	7	225 331 367
6	1/5	8	28	13	257 233 323 271 357
7	1/5	8	28	13	257 233 323 271 357
8	1/6	8	34	16	253 375 331 235 313 357
Promedio	0.279	8			

Tabla 4.16: Código convolutacional escogido para la prueba 3 empleando el esquema UEP.

Resultados de la prueba 3.

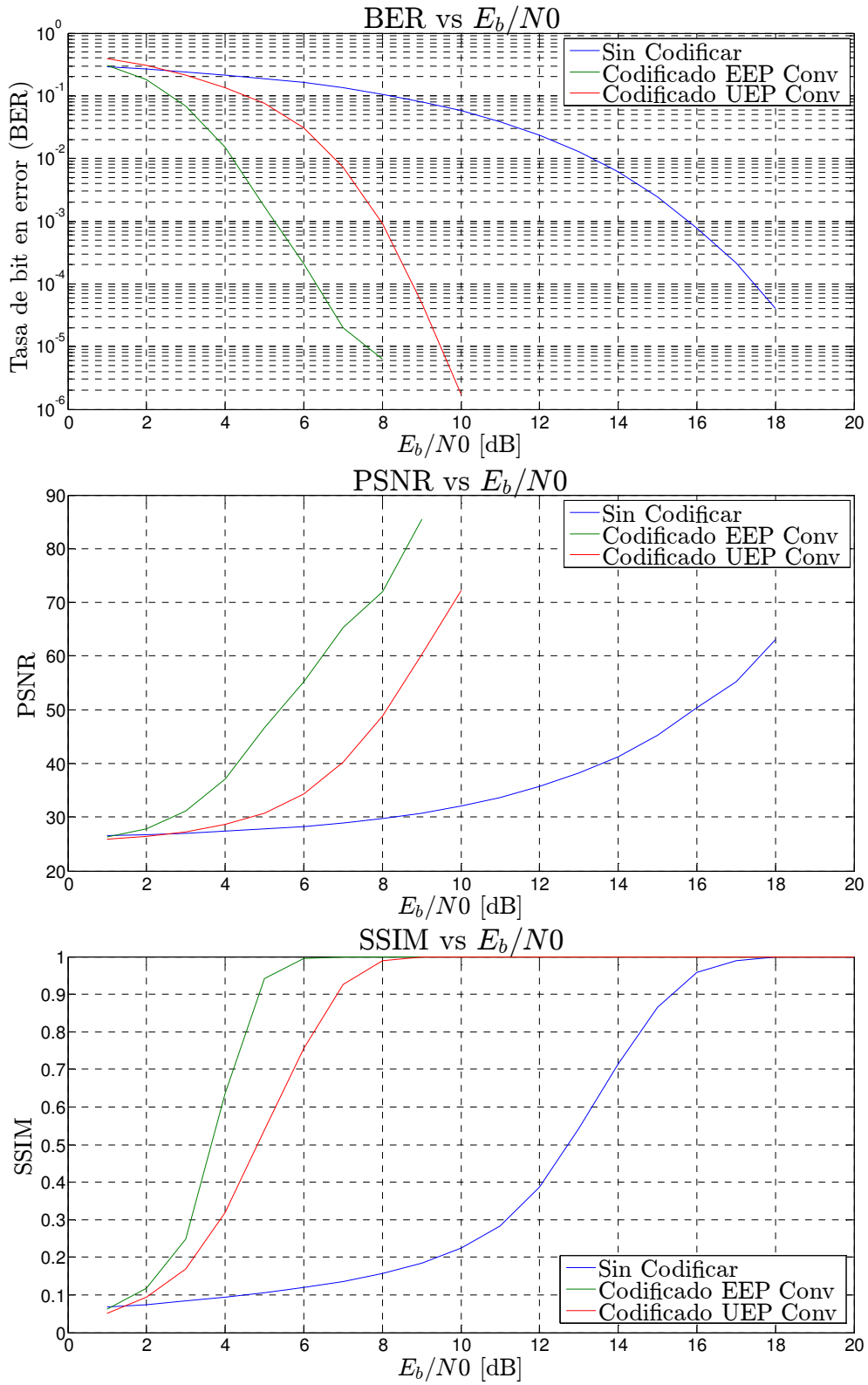


Figura 4.21: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional de la prueba 3, obtenidos para la imagen "Tux".

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	21.18
Convolutacional EEP	58.64
Convolutacional UEP	50.56

Tabla 4.17: Tiempos de ejecución para distintos esquemas con codificación convolutacional de la prueba 3.

De la Figura 4.21, se constata que se lograron mejorar para una $E_b/N_0 = 10$ [dB] los valores de calidad: PSNR = 72.264 [dB], SSIM = 0.9999 y BER = 1.69E-06. Lo cual valida lo establecido en la metodología propuesta, también se pueden apreciar que las curvas de la codificación UEP permaneciéndose sitúan en medio de los esquemas EEP y sin codificación, lo cual significa que el esquema UEP mejora los valores de calidad de la imagen con respecto al sistema sin codificación. Además, el ahorro en redundancia con el esquema UEP (la tasa del código promedio), resultó igual al de la prueba 2, es decir, de 40% con respecto a la redundancia con EEP, mientras que la longitud de restricción promedio con UEP fue de 8, igual que con EEP. En la Tabla 4.17, se dan los tiempos de procesamiento computacional con Matlab y las mismas características de cómputo mencionadas en las mediciones anteriores, observando que el tiempo de ejecución de la codificación UEP con respecto a la codificación EEP tiene un ahorro del 13%. Esta reducción es menor con respecto a la prueba 2, que fue de 20%.

Considerando que el tiempo de ejecución se incrementó, y los valores de calidad aumentaron con respecto a la prueba 2, se realizará otra prueba con la finalidad de encontrar los códigos convolutacionales que reduzcan el tiempo de ejecución, sin alterar los valores de calidad obtenidos en esta prueba para $E_b/N_0 = 10$ [dB].

Prueba 4: Comparación completa entre el caso sin codificación.vs.EEP.vs.UEP para códigos convolutacionales.

Para disminuir la complejidad de la decodificación, y por ende los tiempos de procesamiento, se probará en la capa binaria más significativa una tasa de 1/5 y una longitud de restricción de 8.

Capa	R	K	d_f	t	Code
1	1/2	6	8	3	53 75
2	1/3	6	13	6	47 53 75
3	1/3	6	13	6	47 53 75
4	1/4	6	18	8	51 55 67 77
5	1/4	6	18	8	51 55 67 77
6	1/5	5	20	9	37 27 33 25 35

Capa	R	K	d_f	t	Code
7	1/5	8	28	13	257 233 323 271 357
8	1/5	8	28	13	257 233 323 271 357
Promedio	0.283	6			

Tabla 4.18: Código convolucional escogido para la prueba 6 empleando el esquema UEP.

Resultados de la prueba 4.

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	21.18
Convolucional EEP	58.64
Convolucional UEP	37.35

Tabla 4.19: Tiempos de ejecución para distintos esquemas con codificación convolucional de la prueba 4.

De la Figura 4.22, se tiene para una $E_b/N_0 = 10$ [dB] los valores de calidad: PSNR = 69.683 [dB], SSIM = 0.9999 y BER = 1.62E-04. Estos resultados validan lo establecido en la metodología propuesta, mejorando con respecto a las pruebas 2 y 3. Además, el ahorro correspondiente a la redundancia con el esquema UEP (la tasa del código promedio), fue de 48% con respecto a la redundancia del esquema EEP, mientras que la longitud de restricción promedio en UEP disminuyó de 8 a 6, con relación al esquema EEP. En la Tabla 4.19, se dan los tiempos de procesamiento computacional con Matlab y las mismas características de equipo de cómputo, dando que el tiempo de ejecución de la codificación UEP con respecto a la codificación EEP tuvo un ahorro de 36%. Obteniendo por tanto mejores resultados que los obtenidos con las pruebas 2 y 3.

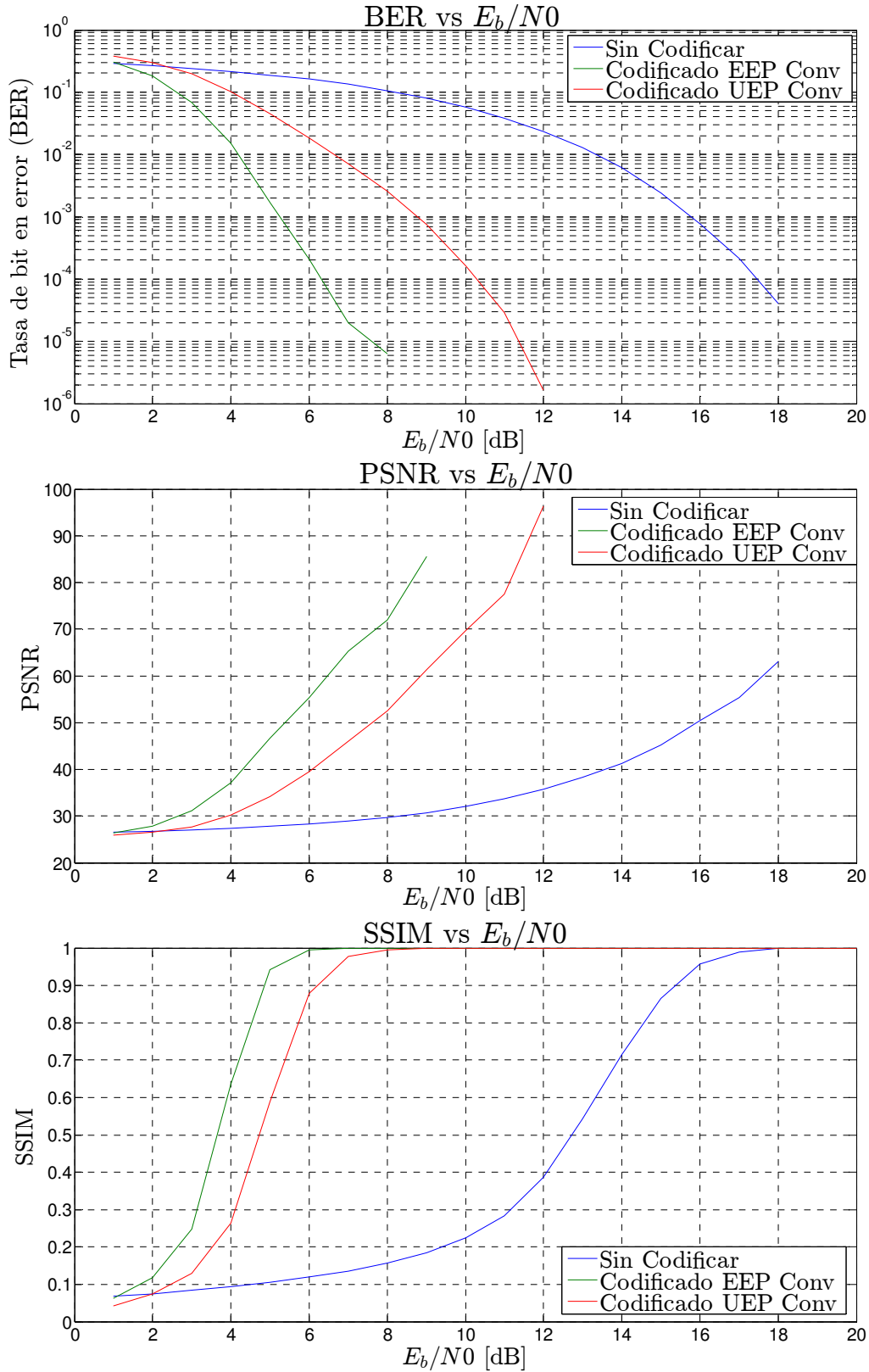


Figura 4.22: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional de la prueba 4, obtenidos para la imagen “Tux”.

En la Tabla 4.20, se da un resumen de los resultados más representativos de las pruebas 4, 5 y 6 de esta sección.

N° de prueba	Prueba 2	Prueba 3	Prueba 4
E_b/N_0 [dB]	10	10	10
Tasa de código promedio	0.279	0.279	0.283
Longitud de restricción promedio	7	8	6
$PSNR_{UEP}$ [dB]	65.549	72.264	69.683
$SSIM_{UEP}$	0.9997	0.9999	0.9999
BER_{UEP}	3.11E-04	1.69E-06	1.62E-04
Tiempo de ejecución [s]	46.51	50.56	37.35

Tabla 4.20: Concentrado de resultados con códigos convolucionales empleando el esquema UEP.

En la Figuras 4.39, 4.40 y 4.41, se presentan las imágenes recibidas con su respectiva imagen de errores, estas últimas amplificadas con un factor de 40 para que los errores se puedan percibir. Estas imágenes, fueron obtenidas de las diferentes pruebas realizadas con el propósito de compararlas de manera visual, para así poder elegir los códigos convolucionales que satisfagan lo estipulado en la metodología de diseño, respetando la calidad de la imagen recuperada en términos de sus valores de calidad, y ahorro de tiempo de ejecución en las simulaciones por computadora.



1.a)



1.b)

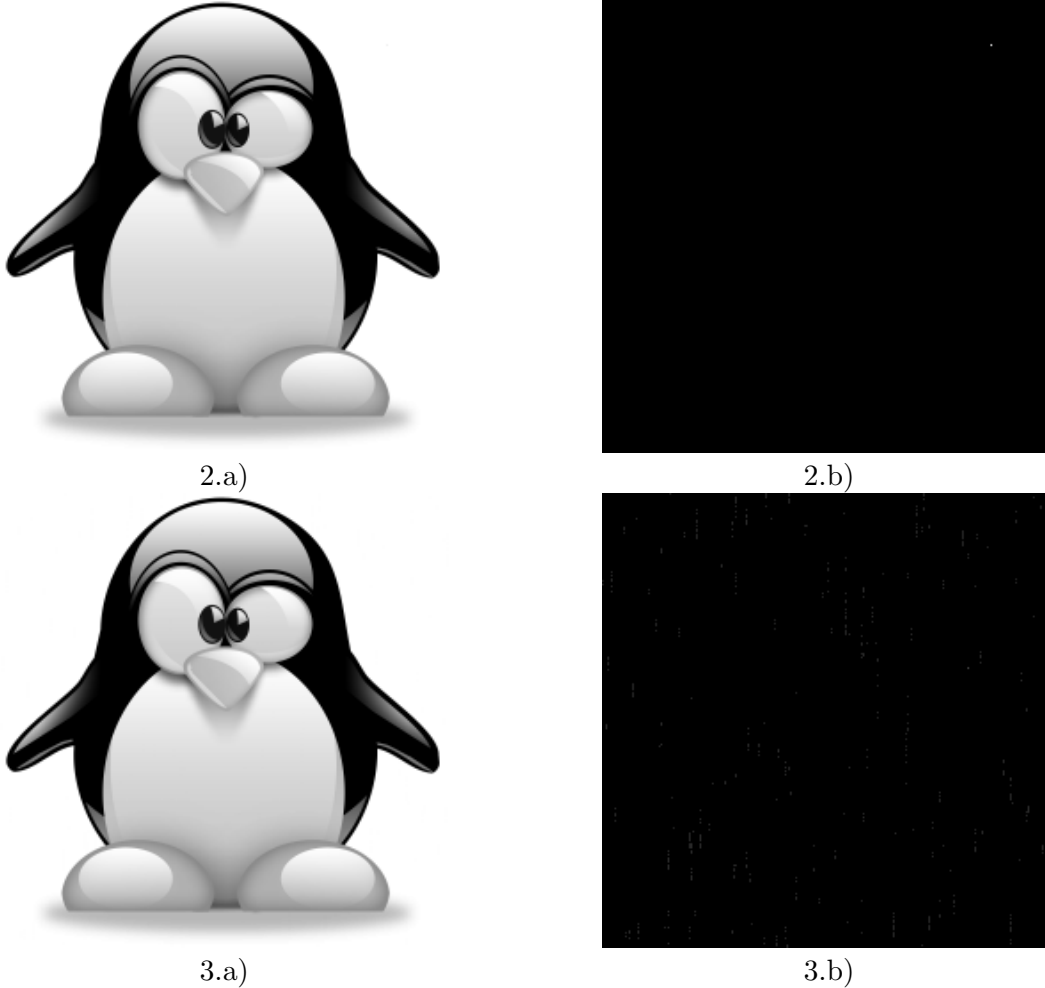


Figura 4.23: Imágenes “Tux” resultantes y de error de las pruebas 2, 3 y 4.

Etapa 6. Análisis de la calidad, ¿ $R_{UEP} < R_{EEP}$?, ¿ $K_{UEP} < K_{EEP}$?, y ¿ $PSNR > 60$ [dB] $BER < 1E-03$ $SSIM > 0.9$?

Dados los resultados cuantitativos y cualitativos de las pruebas 2, 3 y 4 con códigos convolucionales, se tiene que las tres pruebas cumplen con la condición de la Etapa 6. Sin embargo, para continuar con la **Etapa 7**, como un criterio de selección, también es conveniente evaluar los resultados conforme el ahorro en el tiempo de ejecución, en redundancia (tasa del código), en la longitud de restricción (que tiene que ver con la complejidad del decodificador), y la calidad de la imagen en términos de BER, PSNR, y SSIM. A continuación, se presenta el análisis de resultados previo a la Etapa 7.

Análisis visual: Para la prueba 4, perceptualmente la calidad de la imagen no se vio afectada por el ruido del canal como en la prueba 2. La prueba 3, fue la menos afectada de las tres pruebas. En conclusión, las pruebas 3 y 4 fueron las más aceptables a partir de esta apreciación.

Análisis de desempeño computacional: en base a la experimentación, se validó que el ahorro en el tiempo de ejecución de la prueba 4 fue mucho menor con respecto a las pruebas 2 y 3. Esto, debido a que fue la prueba en la cual se obtuvo una tasa de código promedio mayor, pero con una longitud de restricción promedio menor con respecto a las pruebas 2 y 3. Pero cabe subrayar que ambos resultados mejoraron con respecto a lo obtenido con el esquema EEP. Los porcentajes de ahorro en tiempo de ejecución en el esquema UEP de cada prueba, con respecto al esquema EEP, fueron respectivamente los siguientes: 36%, 20% y 13%.

Recordando que para el esquema UEP empleado en la codificación, uno de los objetivos es ahorrar ancho de banda con respecto al esquema EEP y que, conforme a los valores de calidad obtenidos, la imagen recibida sea lo más aproximado a la imagen original. A continuación, se presenta el resultado de dicho análisis del esquema UEP, con respecto al EEP.

- La **prueba 2 con codificación convolucional** mostró un buen desempeño computacional, al ahorrar 20% de tiempo de ejecución del sistema. Sin embargo, visualmente fue la más deficiente, al arrojar valores bajos de calidad con respecto a las pruebas subsiguientes.
- La **prueba 3 con codificación convolucional** mostró un desempeño computacional bajo, al ahorrar 13% de tiempo de ejecución, y además, en calidad visual fue la más alta de las tres pruebas.
- La **prueba 4 con codificación convolucional** mostró un gran desempeño computacional, al ahorrar el 36% de tiempo de ejecución del sistema; los valores de calidad fueron aceptables, con respecto a las pruebas 2 y 3. El resultado visual, también es aceptable, dado que el ojo humano difícilmente detecta los errores de la imagen decodificada.

En consecuencia, la prueba que mejor cumplió con lo establecido en la metodología propuesta en la sección 3.2.2 de esta tesis, fue la **prueba 4**. Conserva la calidad de la imagen recibida, tanto desde el punto de vista cualitativo, como cuantitativo, sin sacrificar de más los tiempos de ejecución del sistema.

Etapa 7. Uso de matrices de perforación (punctured matrices) en las capas binarias que requieren menor protección.

Utilizando los resultados de la prueba seleccionada en la **Etapa 6**, se pretende elevar la tasa de codificación aplicando el concepto de perforación, muy usado en los sistemas

comunicaciones digitales con codificación convolucional. Sin, por lo tanto, afectar de manera significativa la calidad de la imagen, y respetando la longitud de restricción obtenida.

Con fines de resumen, solo se exponen las pruebas más relevantes que produjeron el mejor resultado del diseño UEP en la imagen. Todas las matrices de perforación empleadas en las siguientes pruebas, provienen de [35], [36], y [37].

Prueba 5: Comparación completa: sin codificación.vs.EEP.vs.UEP con codificación convolucional y con la estrategia RCPC (rate-compatible punctured convolucional codes).

Código Madre						Código RCPC		
Capa	R	K	d _f	t	Code	Matriz	RCPC	df
1	1/2	6	8	3	53 75	1;1;0;1;1;1;1;1;1	5/8	6
2	1/3	6	13	6	47 53 75	N/A	N/A	N/A
3	1/3	6	13	6	47 53 75	N/A	N/A	N/A
4	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
5	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
6	1/5	5	20	9	37 27 33 25 35	N/A	N/A	N/A
7	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
8	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
						Tasa de código promedio	0.29	

Tabla 4.21: Código convolucional con estrategia RCPC, establecido para la prueba 5 y con UEP.

Resultados obtenidos de la prueba 5.

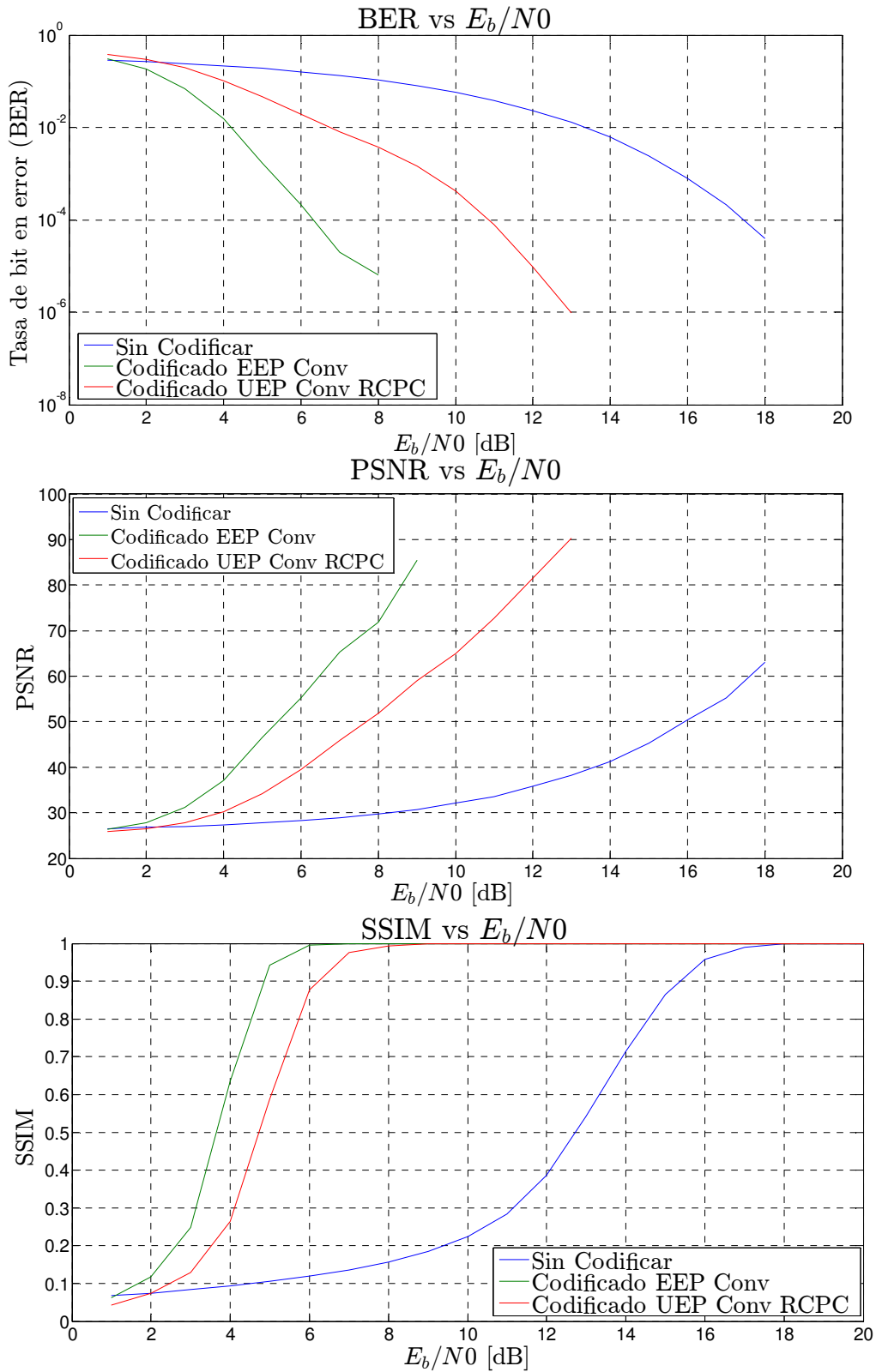


Figura 4.24: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 5, obtenidos para la imagen "Tux".

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	21.18
Convolutacional EEP	58.64
Convolutacional UEP	37.35
Convolutacional RCPC UEP	34.94

Tabla 4.22: Tiempos de ejecución para distintos esquemas con codificación convolutacional, con RCPC de la prueba 5.

De la Figura 4.24, se tiene que para una $E_b/N_0 = 10$ [dB] los valores de calidad son: PSNR = 65.029 [dB], SSIM = 0.9997 y BER = 4.27E-04. Cumpliendo por tanto con lo establecido en la metodología propuesta en esta tesis, se observa también que las curvas del esquema UEP (con matrices de perforación) siguen situándose entre el esquema EEP, y sin codificación. Además, la redundancia (tasa de código) con UEP empleando matrices de perforación disminuyó en un 42 % con respecto a la redundancia con EEP. De la Tabla 4.38, se tiene que el tiempo de procesamiento computacional en Matlab del esquema UEP, con las mismas características de cómputo anteriores, disminuyó en un 40 %, con respecto al esquema EEP.

Buscando mejorar la calidad de la imagen, manteniendo el valor $E_b/N_0 = 10$ [dB], se llevó a cabo otro experimento, donde los valores de calidad obtenidos se incrementen, pero sin afectar la redundancia, ni la longitud de restricción del sistema.

Prueba 6: Comparación completa: sin codificación. vs.EEP.vs.UEP, con codificación convolutacional y esquema RCPC.

Código Madre					Código RCPC			
Capa	R	K	d_f	t	Code	Matriz	RCPC	df
1	1/2	6	8	3	53 75	1;0;1;1;1;1;1;1	4/7	4
2	1/3	6	13	6	47 53 75	1;1;1;1;1;1;1;1; 1;1;1;1;1;1;1;1; 1;0;0;0;1;0;0;0	4/9	10
3	1/3	6	13	6	47 53 75	N/A	N/A	N/A
4	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
5	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
6	1/5	5	20	9	37 27 33 25 35	N/A	N/A	N/A
7	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
8	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
Tasa de código promedio							0.306	

Tabla 4.23: Código convolutacional con RCPC, escogido para la prueba 6 con UEP.

Resultados obtenidos de la prueba 6.

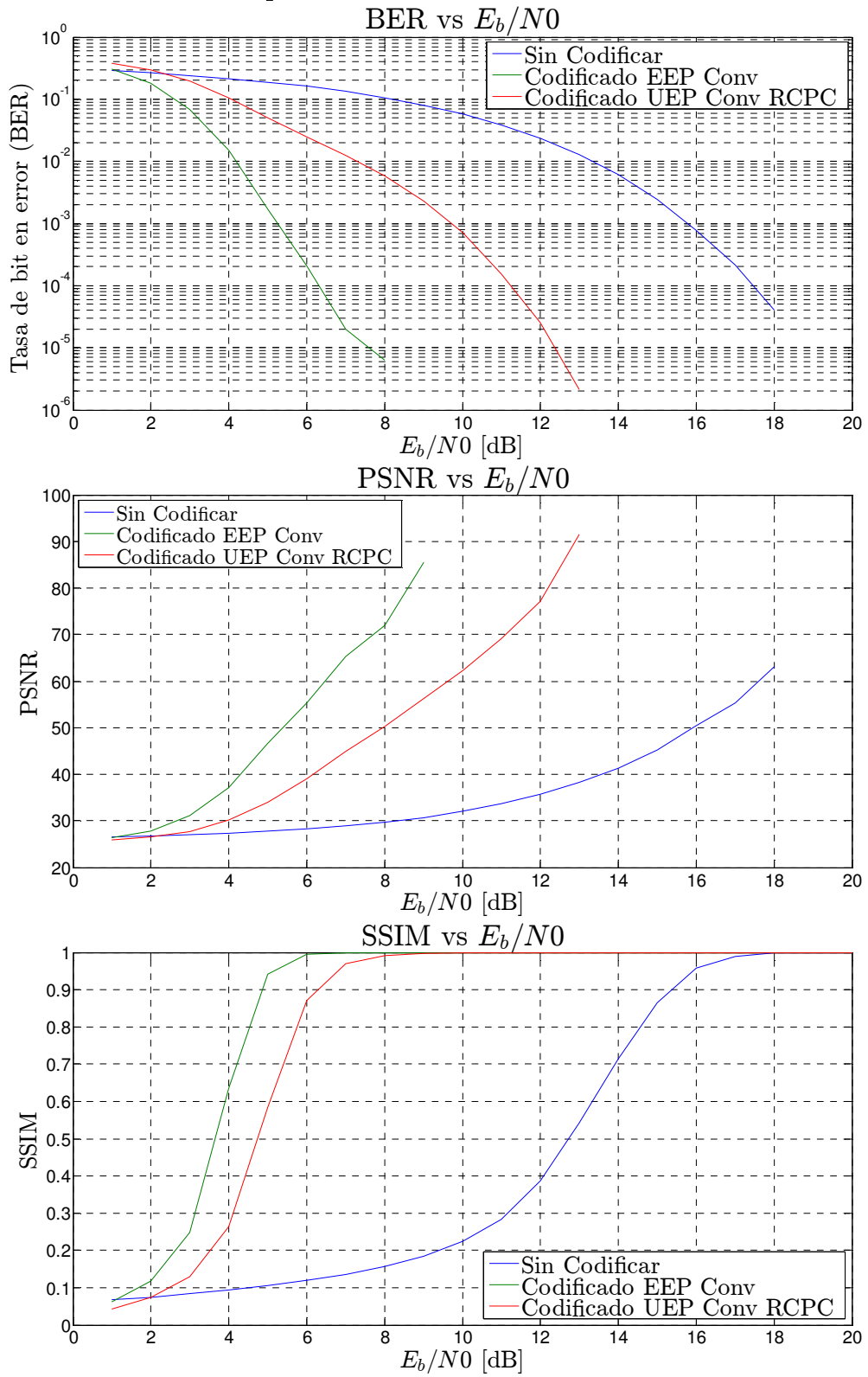


Figura 4.25: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 6, obtenidos para la imagen “Tux”.

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	21.18
Convolutacional EEP	58.64
Convolutacional UEP	37.35
Convolutacional RCPC UEP	33.98

Tabla 4.24: Tiempos de ejecución para distintos esquemas con codificación convolutacional con RCPC, de la prueba 6.

De la Figura 4.25, se tiene que para una $E_b/N_0 = 10$ [dB] los valores de calidad son: PSNR = 61.752 [dB], SSIM = 0.9994 y BER = 8.24E-04, verificando por tanto la metodología propuesta. Se observa que las curvas mantienen el mismo comportamiento que el obtenido en la prueba 7. Además, la redundancia (tasa del código) con UEP, resultante de haber aplicado las matrices de perforación, alcanzó una disminución de un 45 %, con respecto a la redundancia del esquema EEP. De la Tabla 4.24, se tiene que el tiempo del procesamiento computacional en Matlab, con las mismas características de cómputo anteriores, disminuyó en un 42% con el esquema UEP, respecto a EEP.

Un último experimento, manteniendo $E_b/N_0 = 10$ [dB], y buscando mejorar los valores de calidad, sin afectar la redundancia, ni la longitud de restricción del sistema, arrojó los siguientes resultados.

Prueba 7: Comparación completa: sin codificación.vs.EEP.vs.UEP, con codificación convolutacional y RCPC.

Código Madre						Código RCPC		
Capa	R	K	d_f	t	Code	Matriz	RCPC	df
1	1/2	6	8	3	53 75	1;0;1;1;1;1;1 1;1;1;1;1;1;1;1;1;	4/7	4
2	1/3	6	13	6	47 53 75	1;1;1;1;1;1;1;1;1; 1;0;0;0;1;0;0;0 1;1;1;1;1;1;1;1;1;	4/9	10
3	1/3	6	13	6	47 53 75	1;1;1;1;1;1;1;1;1; 1;1;0;0;1;1;0;0	2/5	11
4	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
5	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
6	1/5	5	20	9	37 27 33 25 35	N/A	N/A	N/A
7	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
8	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
Tasa de código promedio							0.314	

Tabla 4.25: Código convolutacional con RCPC, escogido para la prueba 7 con UEP.

Resultados obtenidos de la prueba 7.

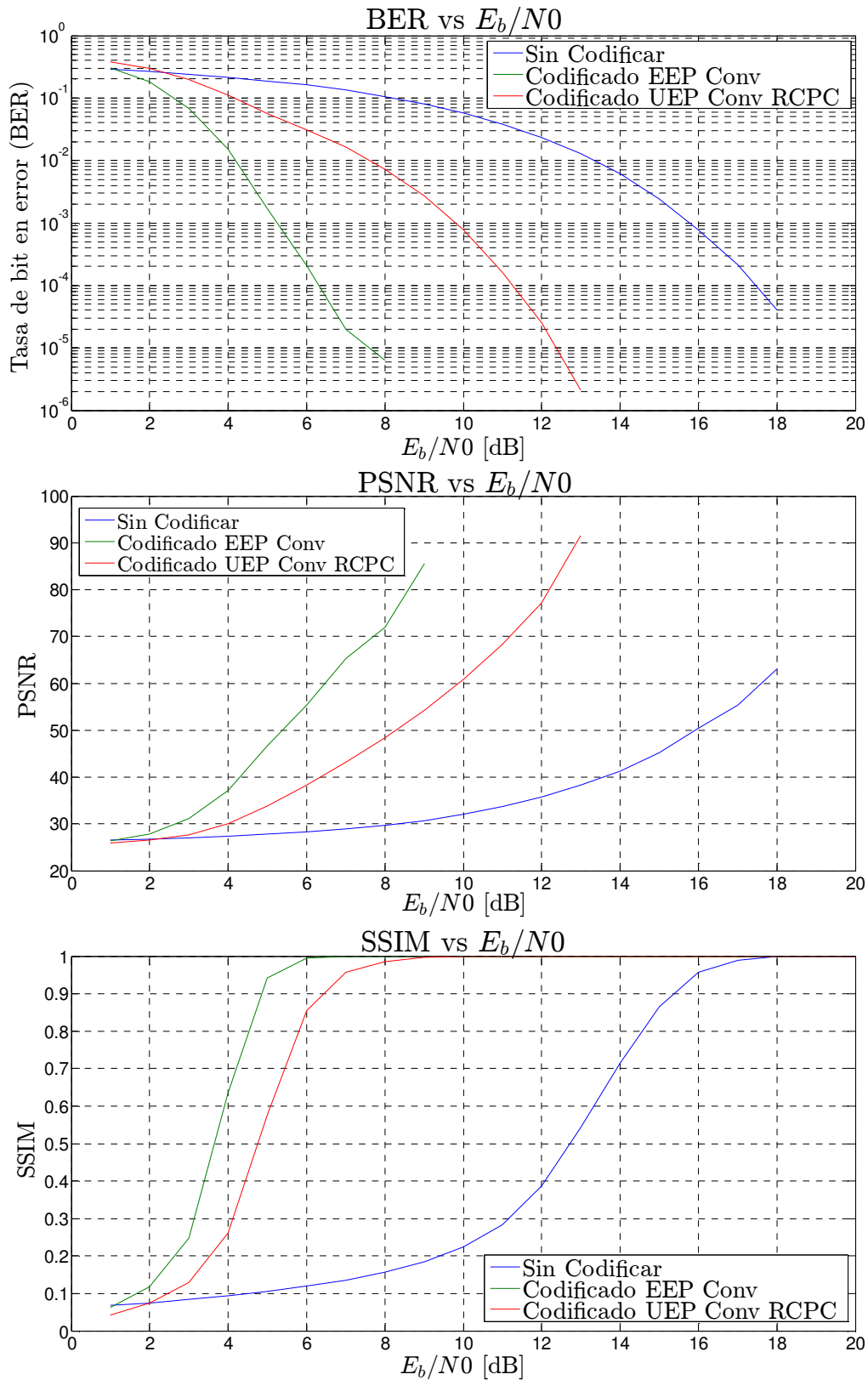


Figura 4.26: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 7, obtenidos para la imagen “Tux”.

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	21.18
Convolutacional EEP	58.64
Convolutacional UEP	37.35
Convolutacional RCPC UEP	32.94

Tabla 4.26: Tiempos de ejecución para distintos esquemas con codificación convolutacional con RCPC, de la prueba 7.

De la Figura 4.26, se obtuvo para una $E_b/N_0 = 10$ [dB] los valores de calidad: PSNR = 60.490 [dB], SSIM = 0.9992 y BER = 8.90E-04. Validando nuevamente lo establecido en la metodología propuesta. La redundancia (tasa del código) con UEP, obtenida usando las matrices de perforación, disminuyó en un 47 %, con respecto a la redundancia con EEP. De la Tabla 4.26, se tiene el tiempo del procesamiento computacional en Matlab, con las mismas características del equipo usadas anteriormente, se puede observar una gran mejora en tiempos de ejecución de la codificación UEP con respecto a la codificación EEP, de hasta un 44 % de ahorro.

En la Tabla 4.27, se da el resumen de los resultados más representativos de las pruebas 5, 6 y 7 de este apartado.

N° de prueba	Prueba 5	Prueba 6	Prueba 7
E_b/N_0 [dB]	10	10	10
Tasa de código promedio	0.29	0.306	0.314
Longitud de restricción promedio	6	6	6
PSNR _{UEP} [dB]	65.029	61.752	60.490
SSIM _{UEP}	0.9997	0.9994	0.9992
BER _{UEP}	4.27E-04	8.24E-04	8.90E-04
%Ahorro en ejecución [s]	40	42	44
%Ahorro en redundancia	42	45	47

Tabla 4.27: Concentrado de resultados con códigos convolucionales con RCPC, empleando UEP.

En la Figuras 4.51, 4.52 y 4.53, se presentan las imágenes decodificadas con su respectiva imagen de errores, estas últimas amplificadas con un factor de 40 para poder percibirlos. Estas imágenes, fueron obtenidas de las diferentes pruebas realizadas con el propósito de comparación, para finalmente elegir los códigos convolucionales con RCPC que satisfagan lo estipulado en la metodología de diseño. Respetando la calidad de la imagen recuperada en términos de sus valores de calidad, y ahorro de tiempo de ejecución en las simulaciones.



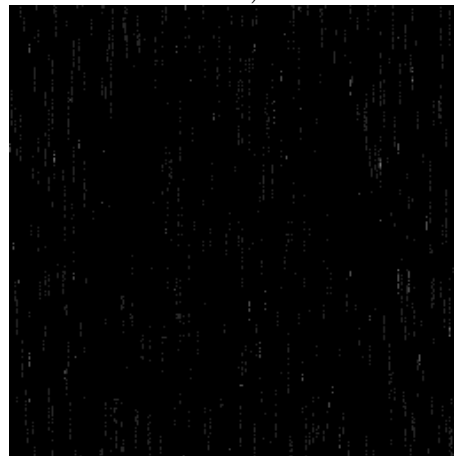
1.a)



1.b)



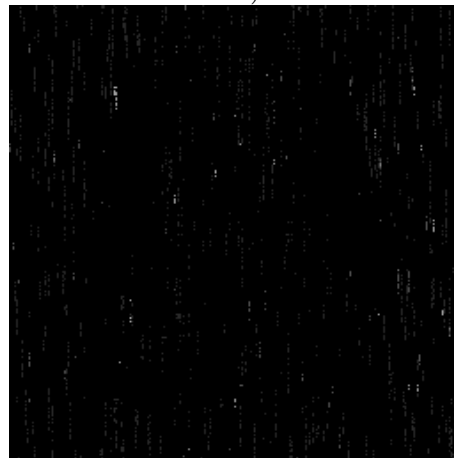
2.a)



2.b)



3.a)



3.b)

Figura 4.27: Imágenes “Tux” resultantes y de error de las pruebas 5, 6 y 7.

**Etapa 8. Análisis de calidad: ¿ $R_UEP_{punc} < R_UEP?$ y ¿ $PSNR_{uep} \approx PSNR_{punc}$
 $BER_{uep} \approx BER_{punc}$ $SSIM_{uep} \approx SSIM_{punc}$?**

Conforme a los resultados cuantitativos y cualitativos de las pruebas 5, 6 y 7, se puede concluir que las tres pruebas cumplen con las condiciones de la Etapa 8. Sin embargo, para hacer una selección, también es conveniente evaluar los resultados respecto al tiempo de ejecución del sistema.

4.2.2.1 Análisis de los resultados provenientes de los códigos convolucionales, con RCPC y UEP.

En esta sección, se presenta el análisis de los resultados, conforme a las pruebas desarrolladas en la sección 4.2.2, con relación al uso de RCPC en los códigos convolucionales.

Análisis visual: Para la prueba 6, perceptualmente la calidad de la imagen no se vio tan afectada por el ruido del canal, como en la prueba 7, y la prueba 5, fue la menos afectada de las tres. Por lo tanto, las pruebas aceptables a partir de esta observación fueron la 5 y la 7. Pero cabe hacer notar que aún en la prueba 7, los errores en la imagen tampoco logran ser francamente perceptibles. Lo cual muestra la robustez de nuestra metodología.

Análisis del desempeño computacional: El empleo del concepto RCPC en estas últimas pruebas, favoreció el resultado en términos de la redundancia (tasa) requerida, y del desempeño computacional. El ahorro alcanzado en el tiempo de ejecución en la prueba 7 fue mucho menor con respecto a las pruebas 5 y 6, porque se tiene una mayor tasa del código, dado que se usó el concepto de perforación en las tres capas binarias menos significativas de la imagen. En general el resultado de estas tres últimas pruebas mejoró con respecto al obtenido con UEP sin RCPC y EEP. Es por tanto remarcable el caso de la prueba 7, el cual casi alcanza un ahorro del 50%, con respecto al tiempo de EEP.

De acuerdo a las observaciones anteriores, se tiene el resultado siguiente:

- La **prueba 5 con codificación convolucional, con RCPC en una capa binaria**, mostró un tiempo de ejecución muy parecido al obtenido en la prueba 4 (UEP sin RCPC). Es decir, solo mejoró en un 4%. Los valores de calidad fueron superiores con respecto a los obtenidos en las pruebas subsiguientes, detectando visiblemente muy poco los errores residuales.
- La **prueba 6 con codificación convolucional, con RCPC en dos capas binarias**, mostró un desempeño computacional regular, puesto que mejoró en un 6% el tiempo

de ejecución, con respecto al resultado de la prueba 6 (UEP sin RCPC). Y en cuanto a calidad, los valores bajaron con relación a la prueba 5; visualmente, los errores no se logran detectar.

- La **prueba 7 con codificación convolucional, RCPC en tres capas binarias**, mostró un gran desempeño computacional, al ahorrar hasta un 44% de recursos del sistema con respecto a EEP. Y los valores de calidad resultantes fueron lo suficiente para que visualmente, no sea posible detectar los errores en la imagen.

En conclusión, la prueba que cumplió adecuadamente con lo establecido en la metodología propuesta en la sección 3.2.2, fue la **prueba 7**. Dado que cumple con las condiciones de calidad cualitativa y cuantitativa. En el sistema de comunicaciones empleado de manera simulada, logró alcanzar un ahorro de casi 50%, tanto en redundancia (tasa), como en desempeño computacional, con respecto al esquema EEP.

Un último experimento, con el fin de comprobar en otro tipo de imágenes el uso de los códigos seleccionados en la **prueba 7 convolucional, con RCPC**, provenientes de la Tabla 4.25, se aplicó el diseño con UEP en una de las imágenes clásicas utilizadas en el procesamiento digital de imágenes, denominada “Lena”. Obteniendo los siguientes resultados:

Resultados obtenidos con la imagen “Lena”.

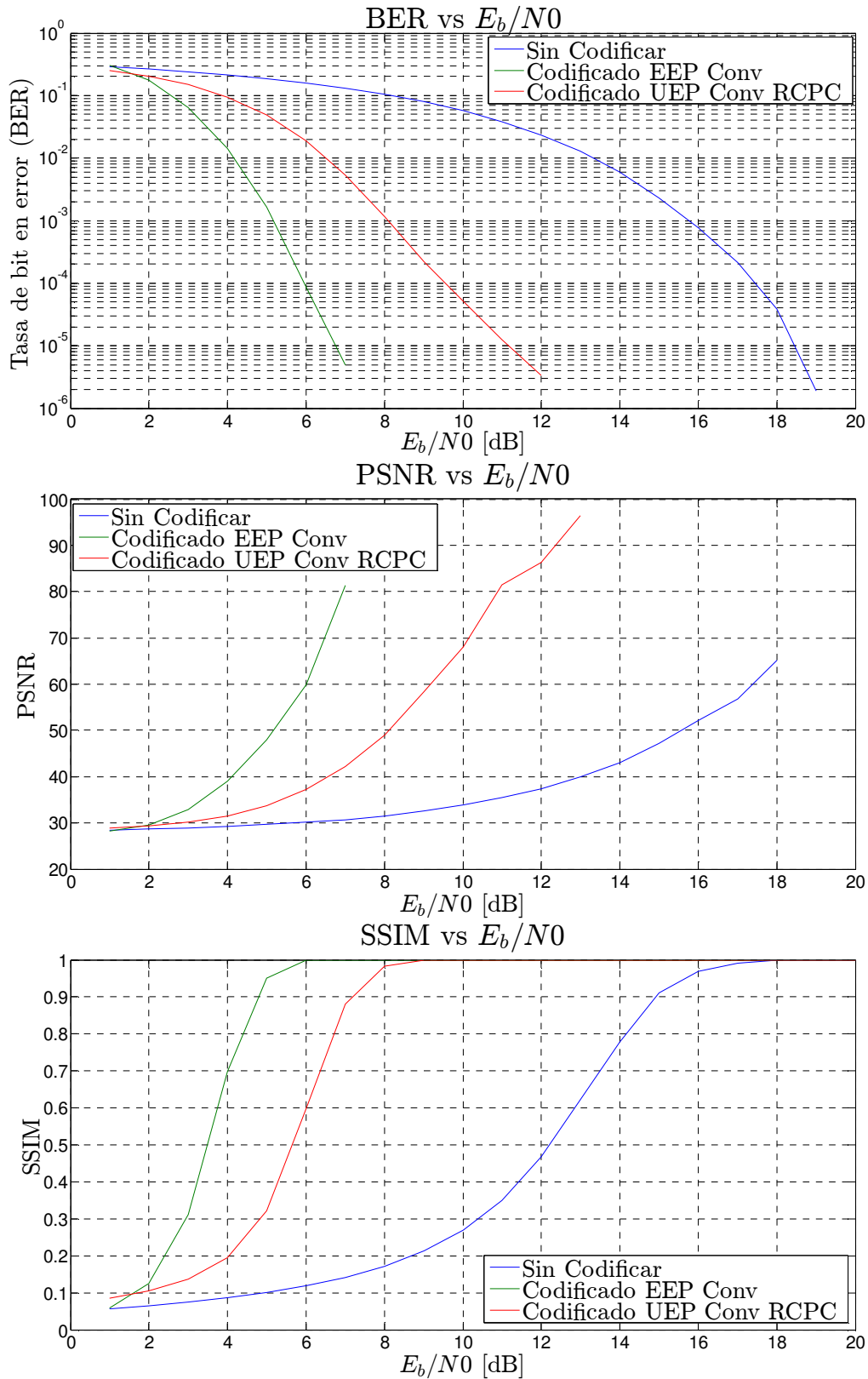


Figura 4.28: Comparación de BER, PSNR, y SSIM para distintos esquemas, con codificación convolucional con RCPC de la prueba 7, obtenidos para la imagen “Lena”.

Tipo de esquema	Tiempo de ejecución [s]
Sin codificación	24.54
Convolutiva EEP	110.72
Convolutiva RCPC UEP	61.77

Tabla 4.28: Tiempos de ejecución para distintos esquemas con codificación convolutiva con RCPC, de la prueba 7 obtenidos con “Lena”.

De la Figura 4.28, se tiene para una $E_b/N_0 = 10$ [dB], los valores de calidad: PSNR = 70.689 [dB], SSIM = 0.9999 y BER = 1.63E-04. Que confirma la validez de la metodología propuesta, y las curvas resultantes mantienen la misma distribución, con respecto a las obtenidas con la imagen “Tux”. Corroborando que la curva del esquema UEP está favorablemente alejada del esquema sin codificación. Por otro lado, se observa que para “Lena”, el código convolutiva de la Tabla 4.25 de la prueba 7, logra alcanzar los criterios ideales en el esquema EEP, para una $E_b/N_0 = 8$ [dB], esto indica que es posible reducir aún más la redundancia del código para una $E_b/N_0 = 10$ [dB], con “Lena”.

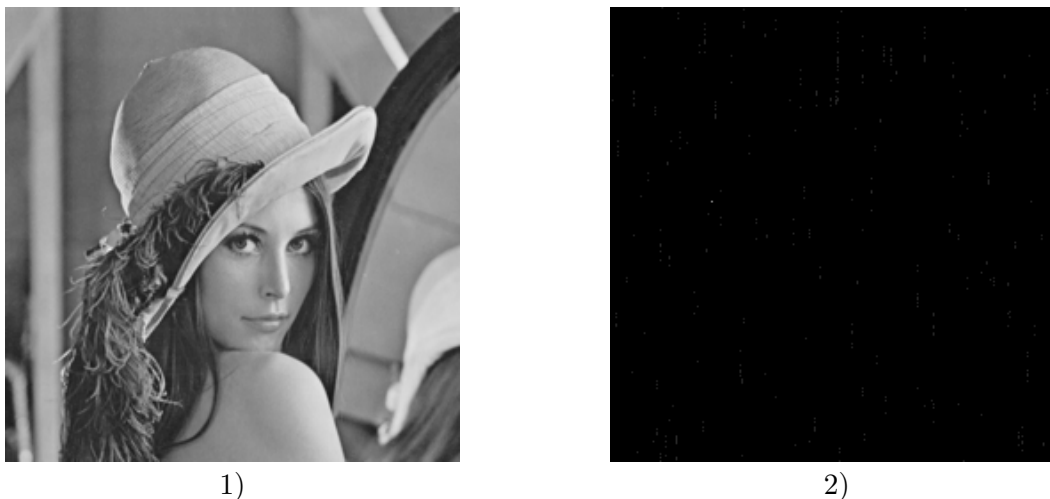


Figura 4.29: “Lena” resultante de la prueba 7 convolutiva, con RCPC. 1) Imagen recibida. 2) Errores de la imagen recibida.

En la Tabla 4.29, se presentan resumidos los resultados finales de las imágenes “Tux” y “Lena”, empleando en ambas la codificación obtenida de la prueba 7.

Imagen	“Tux”	“Lena”
E_b/N_0 [dB]	10	10
Tasa de código promedio	0.314	0.314
Longitud de restricción promedio	6	6
$PSNR_{UEP}$ [dB]	60.490	70.683
$SSIM_{UEP}$	0.9992	0.9999
BER_{UEP}	8.24E-04	1.63E-04
% Ahorro en ejecución	44	44
% Ahorro en redundancia	47	47

Tabla 4.29: Comparación de los resultados de las imágenes “Tux” y “Lena” para códigos convolucionales, con RCPC con UEP.

Como se puede observar, los códigos convolucionales con RCPC de la prueba 7, favorecieron el resultado con “Lena” en términos de sus valores de calidad, con respecto a la imagen “Tux”, demostrando que es posible emplear el diseño resultante del esquema UEP, empleando códigos convolucionales en otras imágenes monocromáticas, con mayor cantidad de textura, de tamaño 256x256, con 8 bits de profundidad.

Capítulo 5

Implementación y análisis de resultados en el entorno físico con USRP2

En este capítulo, se describe el desarrollo experimental que sigue la metodología descrita en el capítulo 3 de este trabajo, con ayuda del concepto SDR haciendo uso de USRP2, con el propósito validar su comportamiento en un sistema de comunicaciones real. Los resultados estimados de la metodología, serán representados mediante tablas, gráficas e imágenes. Cabe mencionar que en el USRP2 será manipulado por medio del software GNU Radio, en el que solo se realiza la etapa de la transmisión/recepción de la imagen, y en Matlab, se realizará la etapa de codificación/decodificación de la imagen.

5.1 Pruebas físicas de la estrategia UEP en la plataforma de radios definidos por software USRP2.

Para tener las partes del transmisor-receptor el sistema de comunicaciones empleado en el USRP2, fue instalado en dos computadoras, ambas con el sistema operativo Ubuntu 14.04 LTS en el ambiente de GNU Radio Companion 3.7.9.3. Esto permitió elegir una computadora para ser el transmisor, y la otra para ser el receptor. Cada USRP2 está conectado a la computadora mediante el cable Gigabit Ethernet, y está provisto de una tarjeta hija RFX1800, y una antena VERT900, que soporta un rango de frecuencias de 1710 a 1990 MHz.

Todos los experimentos fueron desarrollados con una frecuencia de 1.81 GHz, que se encuentra en el rango de frecuencias que acepta el USRP2, y la antena VERT900. Esta frecuencia de operación se determinó por medio de un barrido de ancho de banda, desde 1.5

a 2 GHz, con pasos de 100 MHz, apoyándose en el analizador de espectros propio del radio (comando `uhd_fft` – en la terminal de Ubuntu).

Las especificaciones técnicas que se establecieron en el desarrollo del sistema en el USRP2, se lograron a partir de los siguientes datos: la tasa de muestreo empleada en este sistema (para la transmisión y recepción) fue de 100 Mb/s (máxima tasa que acepta el receptor del USRP2) dividida entre el factor de interpolación/decimación de 1024, lo cual dio como resultado 97.6562 Kb/s; por otro lado, debido a que solo se trabajó con la potencia de salida del USRP2 de 15 dBm (señalada en su hoja de especificaciones), la ganancia fue de 0 dB [42].

En la Figura 5.1, se presenta el esquemático que constituye el transmisor del sistema, y en la Figura 5.4 el correspondiente receptor. De estos, se tienen diversos bloques operativos, mismos que se describen a continuación.

- **File Source.** Archivo creado desde Matlab para transmitir la imagen procesada.
- **Throttle.** Limita la tasa de transmisión del sistema; para este caso, se usó un factor de 1/8 de la tasa establecida, considerando las 8 capas binarias transmitidas, y cuyo resultado fue de 12.207 [kb/s].
- **Packet Encoder.** Es el encargado de indicar al receptor el comienzo de la transmisión. Envía un preámbulo de 16 bits, después un código de acceso conformado de 64 bits (controla que no se propague la salida si existen errores), seguido de la longitud de los datos a transmitir, y finalmente los datos a transmitir. Para este caso, se fijó 2 muestras por símbolo, y un bit por símbolo el cual se empleará en la modulación.
- **GFSK Mod.** Es el tipo de modulación digital que se emplea en este experimento, cuya teoría se explicó en el capítulo 2 del presente trabajo. Los parámetros requeridos en este bloque, se expresan en las siguientes ecuaciones⁶:

$$\text{sensibilidad} = \frac{(\pi * \text{índice de modulación})}{\text{muestras por símbolo}} \quad (4.1)$$

el índice de modulación está dado por la ecuación 4.2

$$\text{índice de modulación} = \frac{(\Delta_f)}{\text{Tasa de baudios}/2} = \frac{(\Delta_f)}{(1/2)R_s} \quad (4.2)$$

Donde Δ_f es la desviación de frecuencia y R_s la tasa de símbolo. Considerando la desviación de frecuencia como la mitad de la frecuencia del símbolo, por lo tanto, el índice de modulación es uno; aplicando el valor de índice de modulación, y

⁶ <http://gnuradio.org/redmine/projects/gnuradio/wiki/SignalProcessing>

considerando que las muestras por símbolo son 2, da como resultado una sensibilidad de 1.570

- **Multiply Const.** Es un factor que multiplica el valor de la amplitud de la señal modulada, en nuestra experimentación, se fijó de 0.1 para todas las pruebas que se llevaron a cabo. Esto debido a que cuando los radios se encontraban a una distancia no mayor a 1 metro, la señal se saturaba en el instante en que no se aplicaba dicho factor.

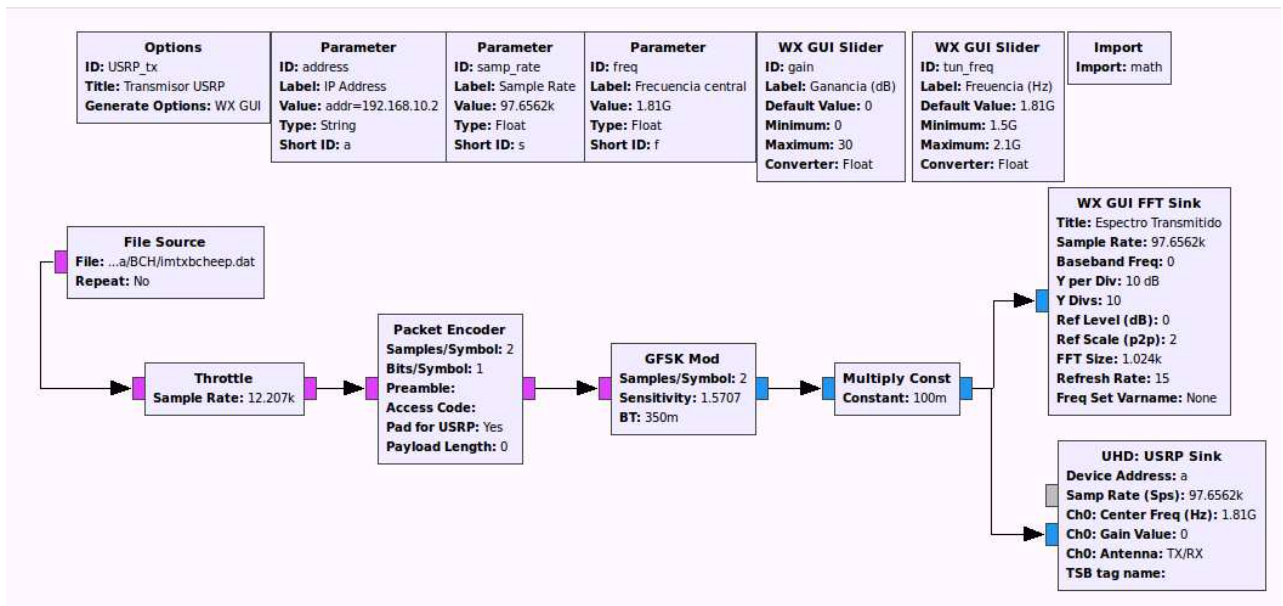


Figura 5.1: Diseño del Transmisor para USRP2.

- **UHD: USRP Sink.** Bloque que define al radio USRP2 para transmisión, se fijan los parámetros antes mencionados: dirección IP para la comunicación entre la computadora, y el USRP2 (192.168.10.2), tasa de muestreo (97.6562 KHz), frecuencia central (1.81 GHz), puerto de antena a utilizar (TX/RX).
- **WX GUI FFT Sink.** Representa gráficamente el espectro de la señal transmitida en banda base. Para este experimento, el espectro transmitido se puede observar en la Figura 5.2.

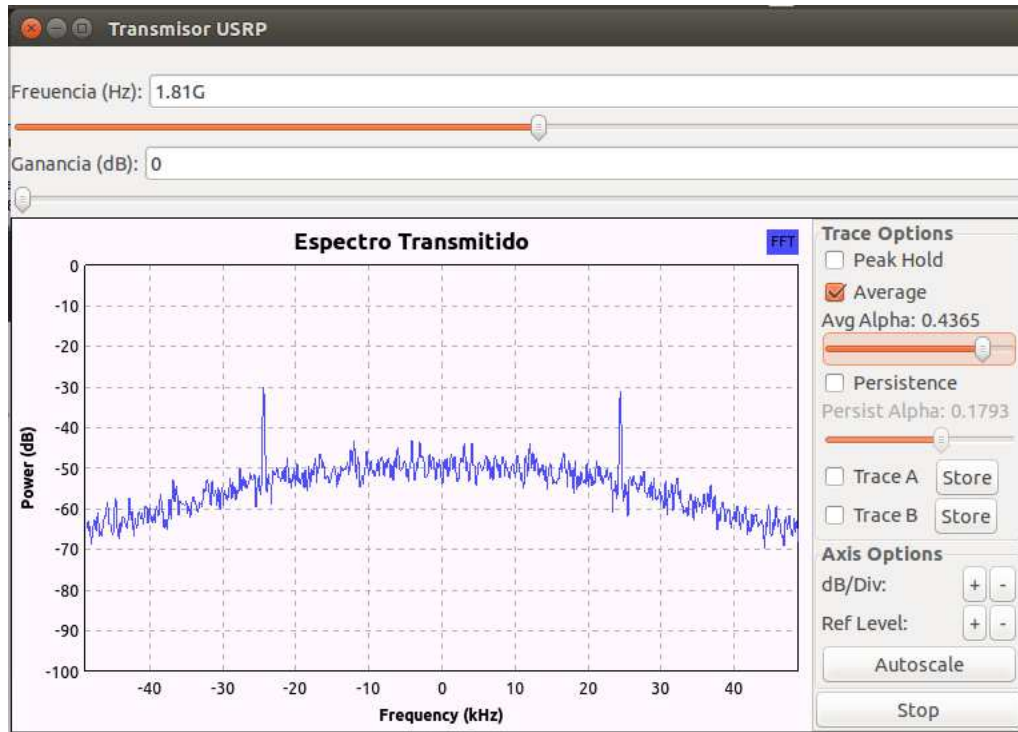


Figura 5.2: Espectro transmitido en banda base.

- **UHD: USRP Source.** Bloque que define al radio USRP2 para recepción, se fijan los mismos parámetros que para el bloque de la transmisión: dirección IP para la comunicación con la computadora, y el USRP2 (192.168.10.2), tasa de muestreo, frecuencia central, puerto de antena a utilizar (para este caso es RX2).
- **WX GUI FFT Sink.** Representa gráficamente el espectro de la señal recibida en banda base. Para este experimento, el espectro recibido se puede observar en la Figura 5.3.
- **GFSK Demod.** Representa la etapa de la demodulación GFSK; por razones prácticas los parámetros del demodulador se mantuvieron iguales a los predeterminados del bloque.
- **Packet Decoder.** La función que realiza este bloque es, al instante de recibir la información, solo ve el código de acceso con el número de bits erróneos disponibles. Una vez sincronizado, lee la cabecera para obtener la longitud de la información recibida, extrae los datos recibidos, y a su vez, los entrega a la salida.
- **File Sink.** Es el archivo con el contenido recibido, mismo que posteriormente se procesará en Matlab, para desplegar la imagen recibida junto con el cálculo de los valores de calidad.

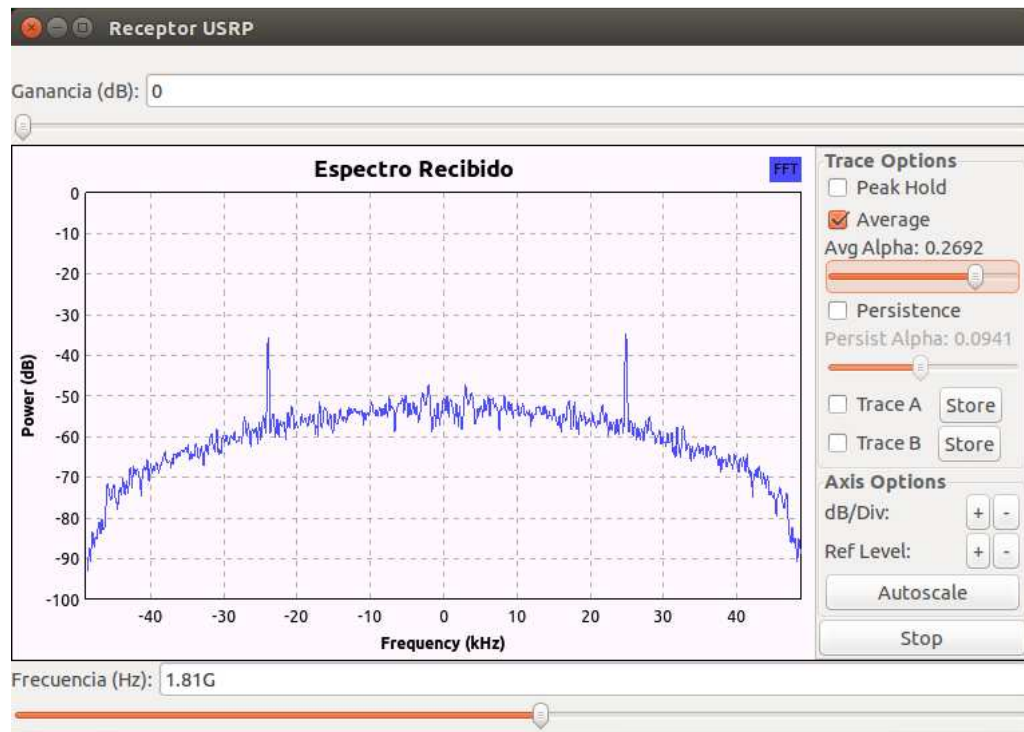


Figura 5.3: Espectro recibido en banda base.

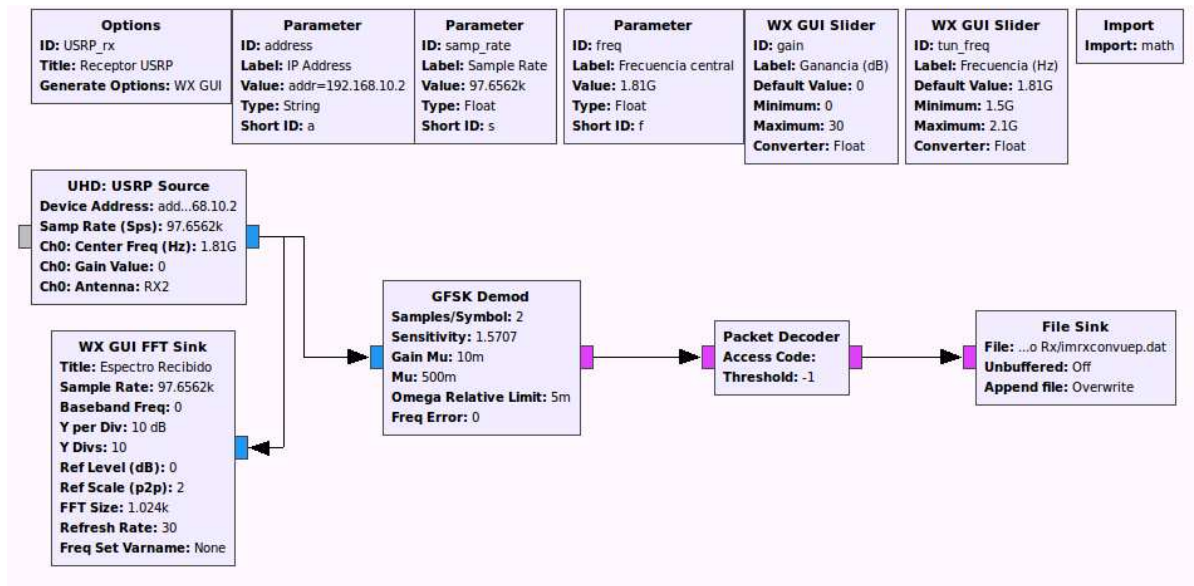


Figura 5.4: Diseño del Receptor para USRP2.

5.1.1 Desarrollo y resultado experimental para comparación de los casos: sin codificación, y con códigos BCH⁷, empleando EEP- UEP

En esta sección, se presentan las pruebas más representativas, siguiendo la metodología propuesta del capítulo 3, sección 3.2.2, la cuales se realizaron para la codificación con códigos BCH, con Matlab - GNU Radio, y hardware de SDR USRP2. Por compatibilidad con las pruebas de simulación en computadora, se siguieron usando las imágenes “Tux” y “Lena”, de tamaño 128x128 pixeles, y 8 bits de profundidad. De esta manera se ha podido validar dicha metodología en un sistema de comunicaciones físico, para diferentes imágenes, empleando codificación de bloque BCH.

Para realizar el experimento en el escenario real descrito en esta sección, se usaron los códigos BCH provenientes de la Tabla 5.1 para el esquema EEP, y de la Tabla 5.2 para el esquema UEP. Haciendo variar la razón señal a ruido del receptor por medio de diversas distancias de separación entre los dos USRP2 (Tx/Rx), cuyo rango empieza desde 0.5 [m], hasta 5 [m].

Capa	n	k	t	Redundancia
1-8	63	10	13	53

Tabla 5.1: Código BCH escogido para el desarrollo experimental con EEP en la plataforma USRP2.

Capa	n	k	T	Redundancia
1	15	5	3	10
2	31	11	5	20
3	31	6	7	25
4	31	6	7	25
5	63	18	10	45
6	63	18	10	45
7	63	18	10	45
8	63	18	10	45
			Promedio	32

Tabla 5.2: Códigos BCH usados para el desarrollo experimental con UEP en la plataforma USRP2.

⁷ Los presentes códigos BCH fueron obtenidos de [25].

5.1.1.1 *Resultados del desarrollo experimental con códigos BCH, para la imagen “Tux”.*

En este apartado, se presentan los resultados obtenidos con los códigos BCH en los esquemas: Sin codificación, – EEP, – UEP, aplicados a la imagen “Tux” sobre la plataforma SDR USRP2. Se hicieron las mediciones de: los valores de calidad para diferentes distancias entre los USRP2 (tablas y gráficas), y las comparaciones de las imágenes recibidas (sin codificar y con UEP) para una cierta distancia seleccionada, a partir de los valores ideales de calidad resultantes del esquema EEP.

De la Tabla 5.3, se puede observar que para una distancia de 3 [m], se logró obtener el caso ideal de los parámetros de calidad en el esquema EEP. Entonces a partir de ello, se emplearon los códigos en el esquema UEP descritos en la Tabla 5.2, con lo que se lograron obtener los siguientes valores de calidad para una distancia = 3 [m]: PSNR = 47.39 [dB], SSIM = 0.9977 y BER = 2.77E-04. Esto nos indica que en estas condiciones de operación, la imagen no fue afectada por el canal inalámbrico, con respecto a los valores del sistema sin codificación. Mientras que para diferentes valores de distancias entre los radios, se pueden apreciar la posición relativa de las curvas, tal como se vieron en los apartados 4.2.1 y 4.2.2 de este trabajo, a diferencia que, para distancias menores, los resultados de los valores de calidad son ideales, con respecto a los obtenidos para distancias mayores. La razón se debe a que, la distancia entre los USRP2 transmisor y receptor, representa el canal inalámbrico de comunicaciones, el cual presenta diversos efectos naturales difíciles de controlar, que provocan errores en la imagen recibida, con respecto a la imagen original.

En la Figura 5.6, se presenta la comparativa de las imágenes recibidas de “Tux” con su respectiva imagen de errores residuales, obtenidas usando una distancia entre los dos equipos USRP2 de 3 [m], sin codificación, y con codificación BCH y esquema UEP.

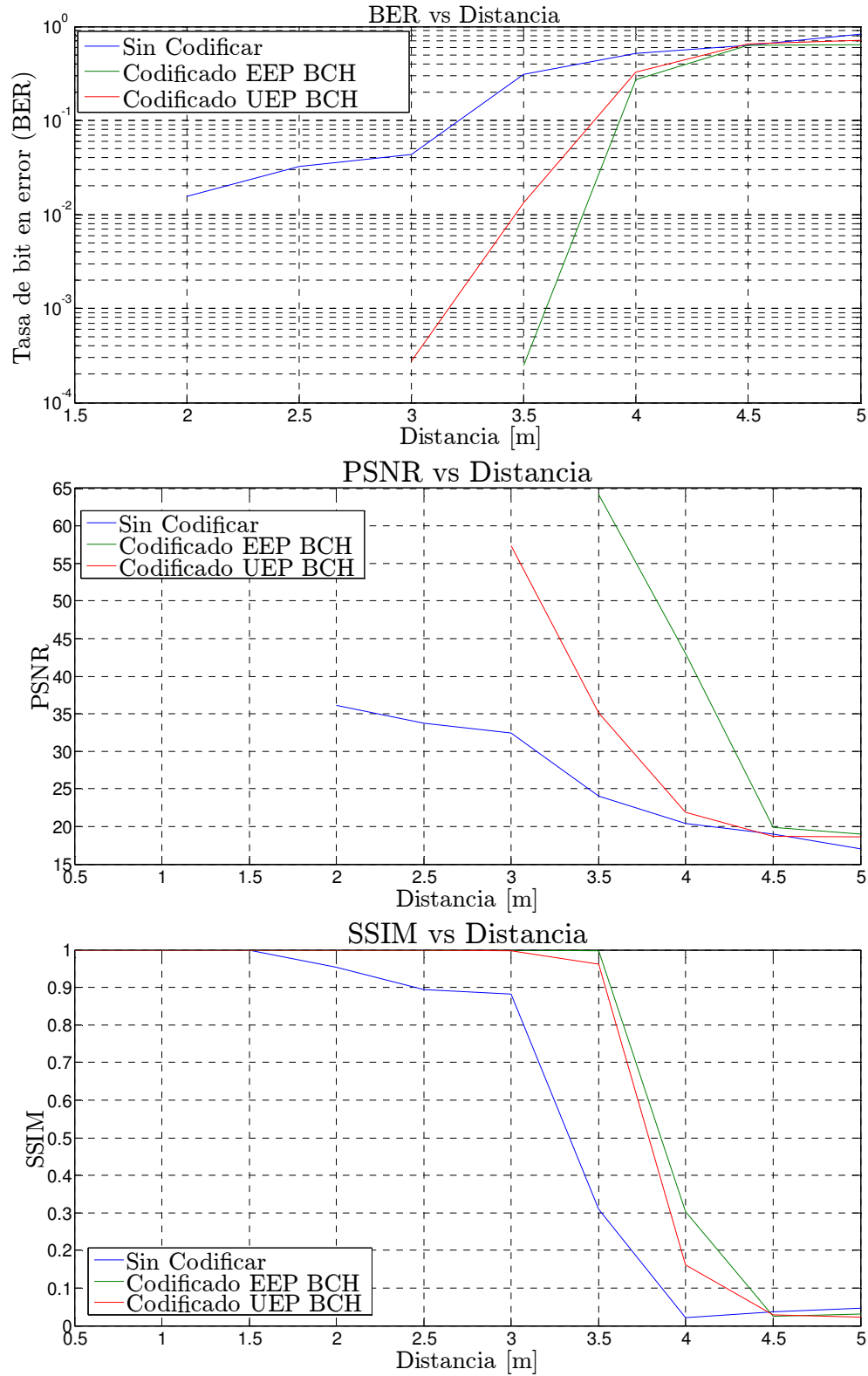


Figura 5.5: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación en BCH obtenida de “Tux”, empleando el USRP2.

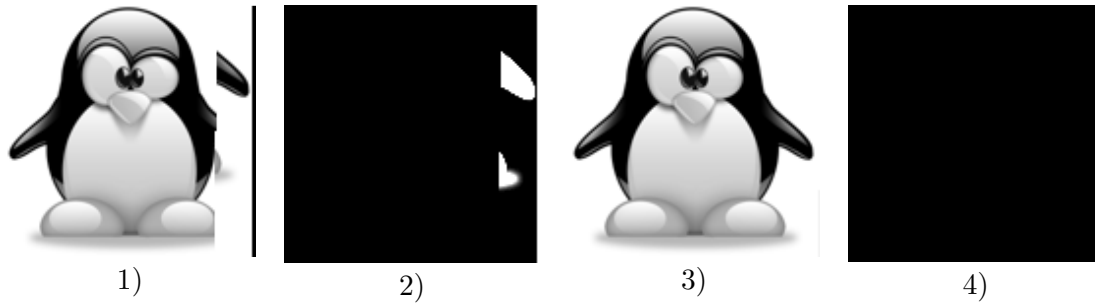


Figura 5.6: “Tux” resultante con 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación BCH con UEP. 4) Errores residuales con codificación BCH, con UEP.

5.1.1.2 *Resultados del desarrollo experimental con códigos BCH para la imagen “Lena”.*

A manera de comparación, en este apartado, se presentan los resultados obtenidos con los códigos BCH, usados en los esquemas: Sin codificación, – EEP, - UEP, aplicados a la imagen “Lena” sobre la plataforma SDR USRP2. Haciendo las medidas de: los valores de calidad para diferentes distancias entre los USRP2 (tablas y gráficas), y las comparaciones de las imágenes recibidas (sin codificar y con UEP), para una cierta distancia entre transmisor y receptor, seleccionada a partir de los valores ideales de calidad resultantes del esquema EEP.

De la Figura 5.7, se tiene que para una distancia de 3 [m], se logró obtener el caso ideal de los parámetros de calidad en el esquema EEP. Entonces a partir de ello, se emplearon los códigos en el esquema UEP dados en la Tabla 5.2, logrando obtener los siguientes valores de calidad para una distancia = 3 [m]: PSNR = 52.8520 [dB], SSIM = 0.9997 y BER = 1.21E-04. Estos valores nos permiten constatar que la imagen no fue afectada por el canal inalámbrico, con respecto a los valores del sistema sin codificación, al igual que para la imagen “Tux”.

En la Figura 5.8, se presentan la comparativa de las imágenes recibidas de “Lena” con su respectiva imagen de errores residuales, correspondientes a una distancia entre los dos USRP2 de 3 [m], sin codificación, y con codificación BCH y esquema UEP.

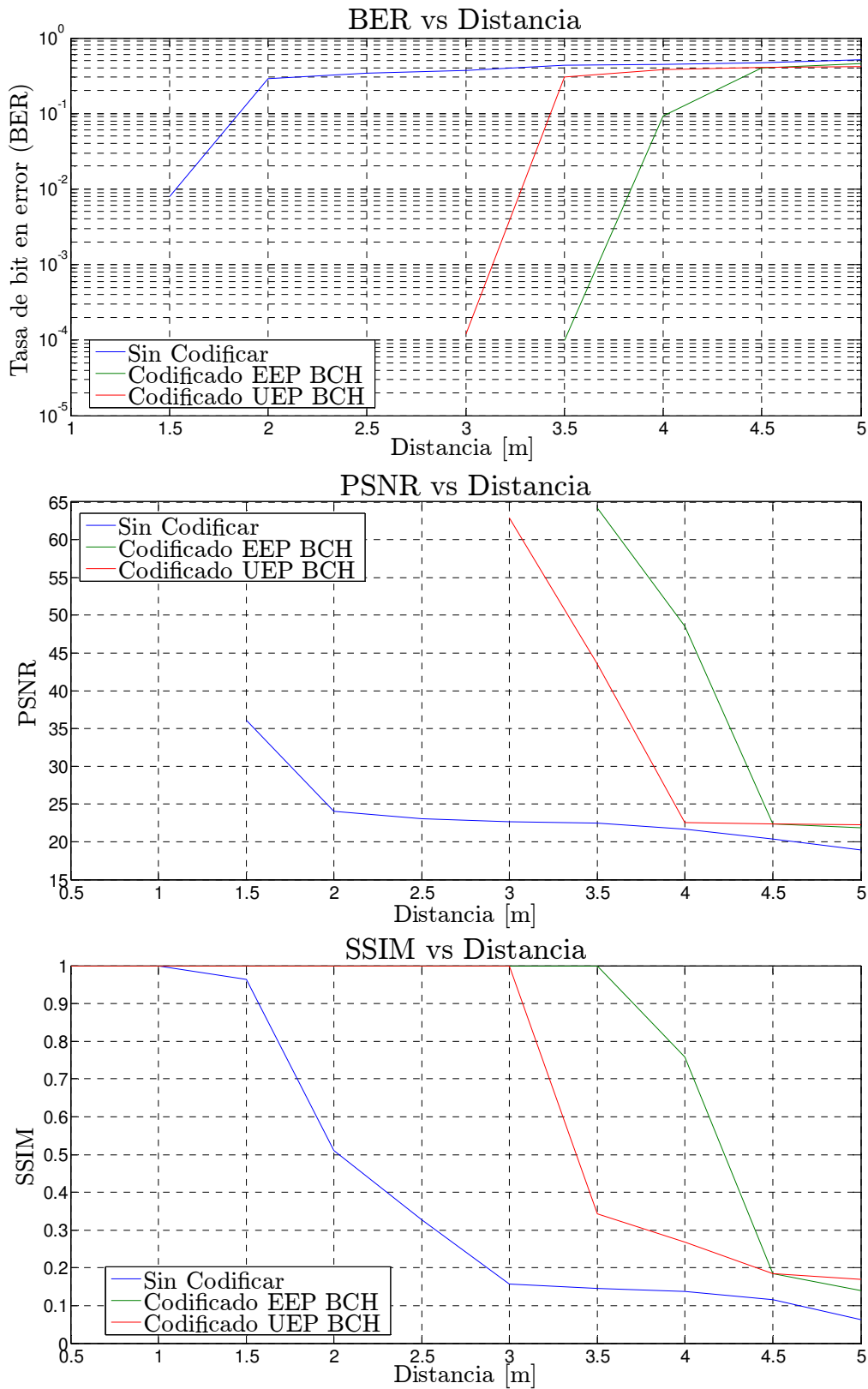


Figura 5.7: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación en BCH obtenida de “Lena”, empleando el USRP2..

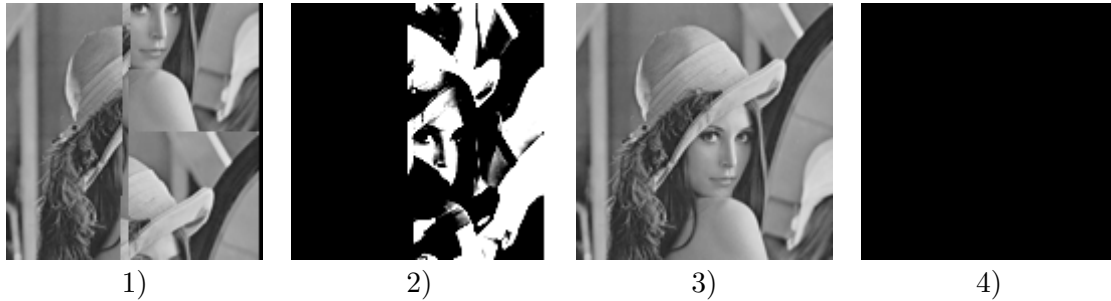


Figura 5.8: “Lena” resultante de 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación BCH con UEP. 4) Errores con codificación BCH con UEP.

5.1.1.3 Análisis de los resultados usando los códigos BCH con UEP

En la presente sección, se realiza una comparación de resultados obtenidos de las secciones previas con respecto a las imágenes: “Tux” y “Lena”.

Para ambas imágenes, el valor de la redundancia promedio con UEP disminuyó 40 % con respecto a la redundancia con EEP. En la Tabla 5.3, se muestra el tamaño, y el tiempo de la transmisión/recepción de los datos, donde se aprecia la mejora en tiempos de la transmisión usando la codificación UEP, con respecto a la codificación EEP, hasta en un 20 %.

Tipo de esquema	Tamaño del archivo [KB]	Tiempos de transmisión/recepción [s]
Sin codificación	256	21
BCH EEP	1614	132.21
BCH UEP	1323	108.38

Tabla 5.3: Tiempos de tx/rx para diferentes esquemas con codificación BCH mediante el USRP2, obtenidos de “Tux” y “Lena”.

Imagen	“Tux”	“Lena”
Distancia [m]	3	3
PSNR _{UEP} [dB]	57.39	62.8520
SSIM _{UEP}	0.9977	0.9997
BER _{UEP}	2.77E-04	1.21E-4
% Ahorro tiempo de transmisión	20	20
% Redundancia	40	40

Tabla 5.4: Comparación de resultados de las imágenes “Tux” y “Lena” empleando códigos BCH, con USRP2.

Como se puede observar en las gráficas de “Tux” (Figura 5.5), y “Lena” (Figura 5.7), a pesar de ser imágenes que conservan las mismas características entre sí (tamaño, formato),

por el contenido de textura diferente, el efecto que causó el canal de comunicación afectó de manera también diferente a cada una de ellas. Sin embargo, ambas imágenes se pudieron recibir en una distancia de 3 [m] para el esquema EEP, usando el mismo código. También en dichas gráficas, se puede observar que, a menor distancia entre los USRP2, como es de esperarse, mayor es el valor de PSNR y SSIM, mientras que el valor de BER disminuye. Esto nos permite constatar que entre más cerca estén los radios, mejoran los valores de calidad de las imágenes, en término de los criterios ideales.

Examinando los resultados de PSNR tanto para “Tux” y “Lena”, se puede apreciar que son considerablemente “bajos” con respecto a lo que se había obtenido en las secciones anteriores, donde el límite permitido más bajo fue de 60 [dB]. No obstante, en las Figuras 5.6 y 5.8, se muestra que visualmente, las imágenes correspondientes a “Tux” y “Lena”, no sufrieron graves alteraciones a esa distancia con respecto a la imagen original, por lo que se puede concluir que ambas imágenes recibidas, y los códigos usados en ellas son óptimos para su uso.

5.1.2 Desarrollo y resultado experimental para comparación de los casos: sin codificación, y con códigos convolucionales, empleando EEP- UEP

De manera similar a lo realizado con los códigos BCH, en esta sección, se presentan las pruebas más representativas para la codificación convolucional, siguiendo la metodología propuesta del capítulo 3 sección 3.2.2, con Matlab - GNU Radio, y hardware del equipo SDR USRP2. Por coherencia con las pruebas anteriores se siguen usando las imágenes “Tux” y “Lena”, con las mismas características.

La experimentación se realizó con el escenario físico descrito en la sección 4.3, usando los códigos convolucionales provenientes de la Tabla 5.5 para el esquema EEP, y de la Tabla 5.8 para el esquema UEP. Para cambiar la razón señal a ruido del receptor, se consideran diferentes distancias de separación entre los dos equipos USRP2 (Tx/Rx), cuyo rango de cobertura empieza desde 0.5 [m], hasta 5 [m]. Cabe mencionar que para experimentar con el esquema RCPC, las matrices de perforación usadas, provienen de [35], [36], y [37].

Capa	R	K	d_f	t	Code
1-8	1/6	8	34	16	253 375 331 235 313 357

Tabla 5.5: Código convolucional escogido para el desarrollo experimental con EEP en el USRP2.

Código Madre						Código RCPC		
Capa	R	K	d_f	t	Code	Matriz	RCPC	df
1	1/2	6	8	3	53 75	1;0;1;1;1;1;1 1;1;1;1;1;1;1;1;	4/7	4
2	1/3	6	13	6	47 53 75	1;1;1;1;1;1;1;1; 1;0;0;0;1;0;0;0 1;1;1;1;1;1;1;1;	4/9	10
3	1/3	6	13	6	47 53 75	1;1;1;1;1;1;1;1; 1;1;0;0;1;1;0;0	2/5	11
4	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
5	1/4	6	18	8	51 55 67 77	N/A	N/A	N/A
6	1/5	5	20	9	37 27 33 25 35	N/A	N/A	N/A
7	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
8	1/5	8	28	13	257 233 323 271 357	N/A	N/A	N/A
Tasa de código promedio							0.314	

Tabla 5.6: Códigos convolucionales con RCPC usados en el desarrollo experimental con UEP, en el USRP2.

5.1.2.1 Resultados del desarrollo experimental con códigos convolucionales para la imagen "Tux".

De la Figura 5.9, se tiene que para una distancia de 3 [m], se alcanzó el caso ideal de los parámetros de calidad en el esquema EEP. A partir de ello, se emplearon los códigos dados en la Tabla 5.6 para el esquema UEP, logrando obtener los siguientes valores de calidad para una distancia = 3 [m]: PSNR = 42.8164 [dB], SSIM = 0.9840 y BER = 2.30E-03, además de constatar que se sigue respetando el orden de aparición de las curvas de los diferentes esquemas, tal como se tuvo en los apartados 4.2.1 y 4.2.2 de este trabajo. De igual forma, se observa que, de acuerdo al rango de distancia, de 3, a 5 [m], el desempeño del sistema decae, con respecto al rango de distancias de 0.5 a 3 [m], en donde se logra una ganancia en los valores de calidad, con respecto al sistema sin codificación.

En la Figuras 5.10, se presenta la comparativa de las imágenes recibidas de "Tux", con su respectiva imagen de errores, obtenidas a partir de una distancia entre los dos USRP2 de 3 [m], sin codificación, y con codificación UEP convolucional, con RCPC.

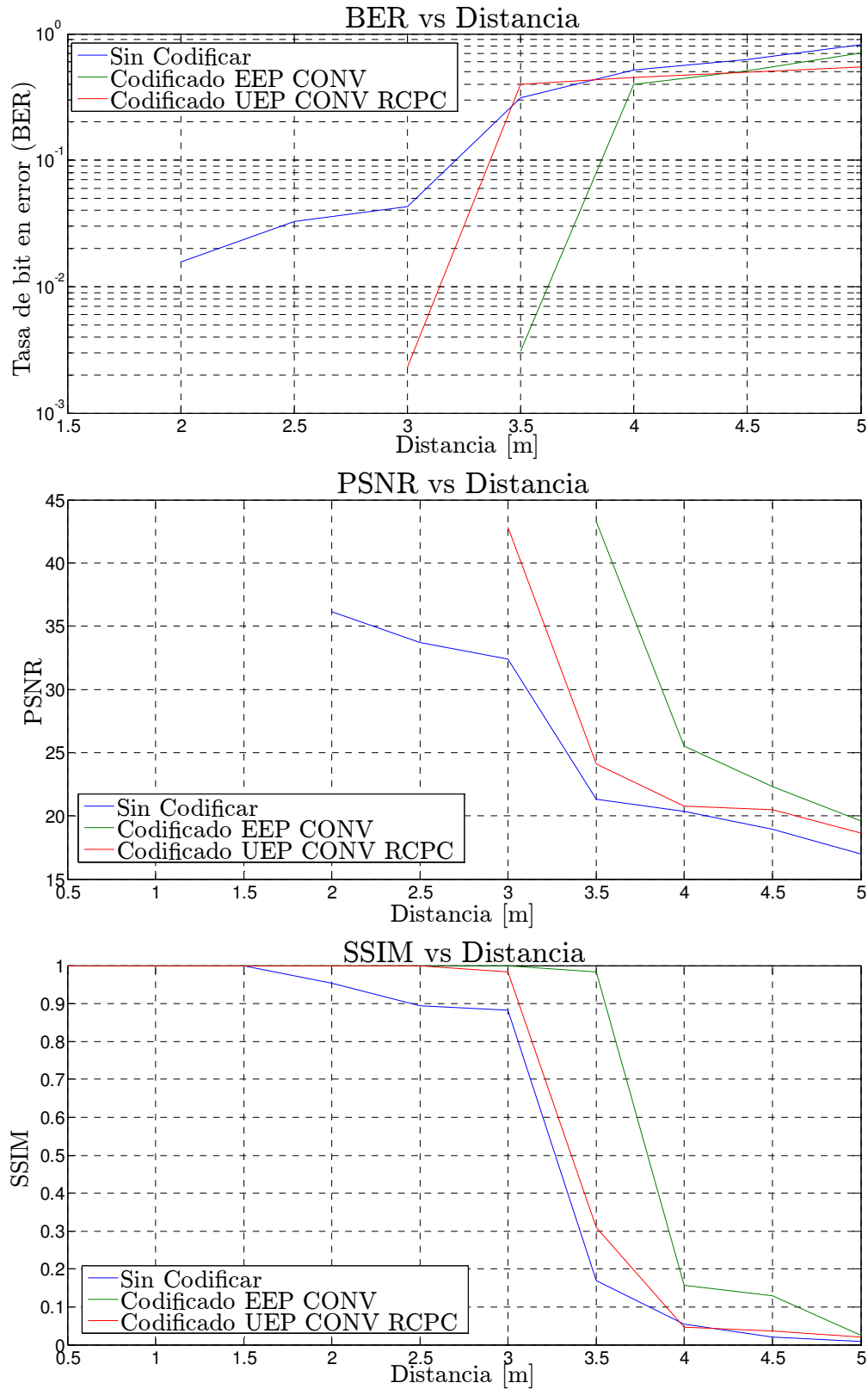


Figura 5.9: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación convolucional RCPC obtenidas de “Tux”, con el uso de USRP2.

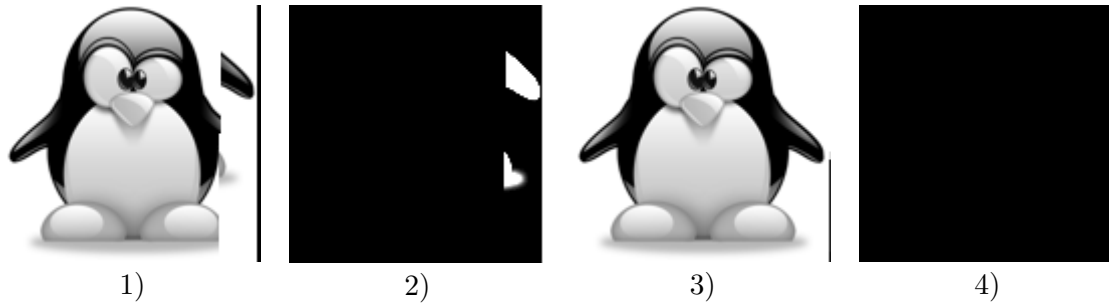


Figura 5.10: “Tux” resultante con 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación convolucional y RCPC para UEP. 4) Errores residuales con codificación convolucional y RCPC para UEP.

5.1.2.2 *Resultados del desarrollo experimental con códigos convolucionales para la imagen “Lena”.*

Los siguientes resultados corresponden a la experimentación con la imagen “Lena”, a la cual se le han aplicado los mismos códigos de las Tablas 5.5 y 5.6.

De la Figura 5.11, se tiene que para una distancia de 3 [m], se alcanzó el caso ideal de los valores de calidad en el esquema EEP. A partir de ello, se emplearon los códigos en el esquema UEP de la Tabla 4.56, lográndose obtener los siguientes valores de calidad: PSNR = 42.8164 [dB], SSIM = 0.9883 y BER = 1.20E-03. Además, se puede apreciar las curvas de los valores de calidad de la imagen, con un desempeño similar al obtenido con la imagen “Tux”, con códigos convolucionales.

En la Figura 5.12, se presenta la comparativa de las imágenes recibidas de “Lena” con su respectiva imagen de errores, obtenidas a partir de una distancia entre los dos USRP2 de 3 [m], sin codificación, y con codificación UEP convolucional, con RCPC.

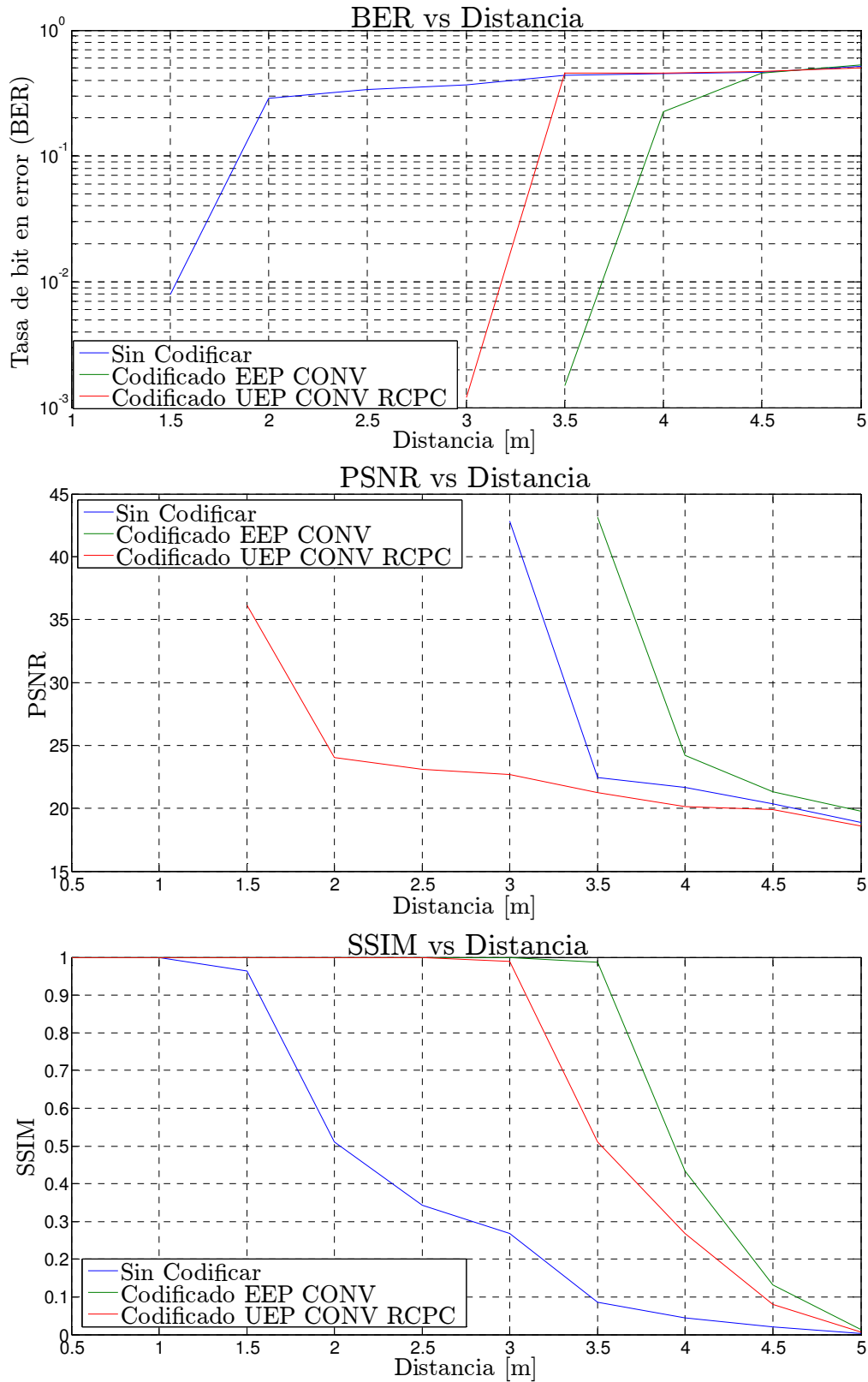


Figura 5.11: Comparación de BER, PSNR, y SSIM para distintos esquemas con codificación convolucional RCPC obtenidas de “Lena”, con el uso de USRP2.

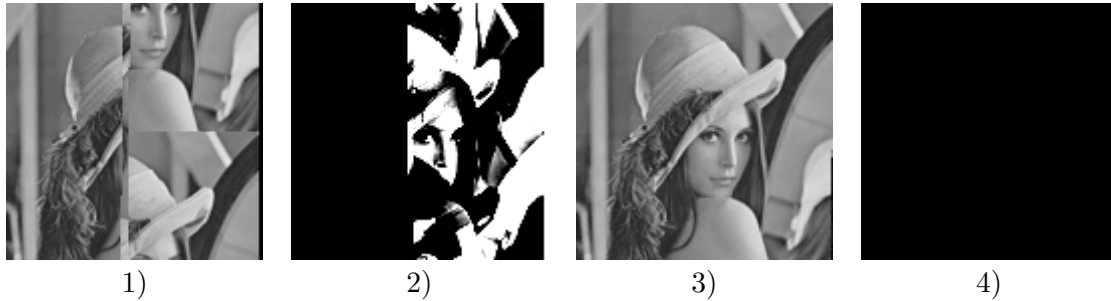


Figura 5.12: “Lena” resultante con 3 [m] de distancia. 1) Imagen recibida sin codificación. 2) Errores sin codificación. 3) Imagen recibida con codificación convolucional y RCPC, con UEP. 4) Errores residuales con codificación convolucional y RCPC, con UEP.

5.1.2.3 Análisis de los resultados usando los códigos convoluciones, con UEP.

En la presente sección, se realiza una comparación de resultados obtenidos de las secciones 5.1.1, y 5.1.2, con las imágenes: “Tux” y “Lena”.

Para ambas imágenes, el valor de la redundancia (tasa) del código promedio con UEP, disminuyó en un 47 %, con respecto a la redundancia (tasa) del código con EEP. En la Tabla 5.11, se muestra el tamaño, y el tiempo de la transmisión/recepción de los datos, se observa una mejora en tiempos de la transmisión de la codificación UEP, con respecto a la codificación EEP, de hasta un 16 %.

Tipo de esquema	Tamaño del archivo [KB]	Tiempos de transmisión/recepción [s]
Sin codificación	256	20.97
Convolucional EEP	1536	125.82
Convolucional RCPC UEP	1280	104.85

Tabla 5.7: Tiempos de tx/rx para diferentes esquemas con codificación convolucional mediante el USRP2, obtenidos de “Tux” y “Lena”.

Imagen	“Tux”	“Lena”
Distancia [m]	3	3
PSNR _{UEP} [dB]	42.8164	42.8194
SSIM _{UEP}	0.9840	0.9883
BER _{UEP}	2.03E-03	1.20E-3
% Ahorro tiempo de transmisión	16	16
% Ahorro en redundancia	47	47

Tabla 5.8: Comparación de resultados de las imágenes “Tux” y “Lena” empleando códigos convolucionales con UEP, mediante la plataforma USRP2.

Como se puede observar en las gráficas de “Tux” (Figura 5.9) y “Lena” (Figura 5.11), a pesar de ser imágenes con los mismos parámetros (tamaño, formato), debido al diferente contenido en textura, el efecto que causó el canal de comunicación fue diferente en cada una de ellas.

Ahora bien, examinando los resultados de PSNR tanto para “Tux”, como para “Lena”, se puede ver que son considerablemente “bajos” con respecto a lo que se había obtenido en las secciones anteriores, donde el límite permitido más bajo fue de 60 [dB]. No obstante, en las Figuras 5.10 y 5.12, se observa que visualmente, las imágenes correspondientes a “Tux” y “Lena”, no sufrieron graves alteraciones a esa distancia con respecto a la imagen original, por lo que se puede concluir que ambas imágenes recibidas, y los códigos usados en ellas son adecuados para su uso.

5.2 Análisis de resultados finales.

En esta última sección, con el propósito de reafirmar el buen funcionamiento de los códigos correctores de errores estudiados en este trabajo (BCH y convolucionales), y subrayar el correcto desempeño de la metodología UEP propuesta, se realiza el análisis de resultados, resumiendo los análisis desarrollados en las secciones 4.2.1.1, 4.2.2.1 del capítulo 4, y 5.1.1.3, 5.1.2.3 de este capítulo, tomando en cuenta los escenarios simulados en computadora, y físicos implementados.

A continuación, en la Tabla 5.9, se muestra el concentrado de resultados finales de cada sección, considerando el escenario simulado en computadora y físico. Se puede concluir que la metodología propuesta es funcional, debido a que, en todos los casos, permitió reducir el tiempo de transmisión de un equipo USRP2, a otro. Además, también permitió disminuir el tiempo de ejecución computacional con Matlab.

Escenario	Simulado por computadora	Real por USRP2	Simulado por computadora	Real por USRP2
Imagen	Tux	Tux	Tux	Tux
Tipo de códigos	BCH	BCH	Convolucional	Convolucional
E_b/N_0 [dB]	10	N/A	10	N/A
Distancia [m]	N/A	3	N/A	3
PSNRUEP [dB]	69.935	47.39	60.49	42.8164
SSIMUEP	0.9998	0.9977	0.9992	0.984
BERUEP	1.01E-04	2.77E-04	8.24E-04	2.03E-03

Escenario	Simulado por computadora	Real por USRP2	Simulado por computadora	Real por USRP2
% Ahorro en tiempo de ejecución en Matlab	24	N/A	44	N/A
% Ahorro en tiempo de Tx/Rx	N/A	20	N/A	16
% Redundancia	24	40	47	47

Tabla 5.9: Concentrado de resultados para diferentes escenarios y códigos, obtenidos con "Tux".

El análisis relativo al escenario simulado, permite constatar que, el sistema de comunicaciones basado en códigos convolucionales con RCPC, disminuyó casi en un 50% la redundancia promedio en el esquema UEP, con respecto a la requerida en el esquema EEP. En el tiempo de ejecución entre ambos esquemas, también hubo una reducción del mismo orden. Por otra parte, el sistema de comunicaciones basado en códigos BCH con el esquema UEP, logró reducir en casi un 25% el tiempo de ejecución, y redundancia promedio, con respecto al esquema EEP.

Adicionalmente, conforme a lo desarrollado en el escenario real, la redundancia promedio relativa a los códigos BCH con UEP, se redujo en un 40%, y en el caso de códigos convolucionales, casi en un 50%, ambos con respecto al esquema EEP. Sin embargo, a diferencia de lo que se observó en el escenario de simulación, el ahorro en tiempo de transmisión, fue menos de la mitad del porcentaje obtenido con la redundancia para ambos sistemas codificados, con respecto al obtenido en EEP, es decir, para los códigos BCH con UEP fue de 20%, y con códigos convolucionales con UEP fue de 16%. Esta diferencia se explica por la tasa de muestreo resultante, la cual fue de 12.207 [kb/s], además, también depende del tamaño de la imagen transmitida (con la redundancia agregada, que aumenta el ancho de banda). De tal modo que el tiempo de transmisión es calculado a partir de la tasa de muestreo, y del tamaño de la imagen transmitida, lo cual hace que el ahorro en el tiempo de transmisión sea variable. Por otra parte, otro punto para analizar en el escenario físico fue que, los valores de calidad obtenidos en el esquema UEP, para ambos sistemas (códigos BCH y convolucionales), con respecto a los valores mínimos establecidos en la metodología propuesta (presentada en el capítulo 3), fueron menores, principalmente el valor de PSNR; no obstante, los valores de SSIM y BER se inclinan hacia dichos valores establecidos, para ambos sistemas. También se puede comentar que, el resultado de acuerdo a la calidad visual, fue muy bueno, ya que los errores residuales no se alcanzan a percibir por el ojo humano, en ambas imágenes recibidas.

Con lo antes expuesto, se evidencia que los códigos correctores de errores (BCH y convolucionales), seleccionados conforme a las diversas pruebas realizadas en el presente capítulo, y empleados en los sistemas de comunicaciones basados en UEP, lograron corregir la mayor parte de los errores de la imagen recibida, lo cual se ve reflejado en los valores de calidad de la imagen, así como también de forma visual, permitiendo un ahorro en redundancia, en tiempo de ejecución, y de transmisión, con respecto al sistema basado en EEP.

Capítulo 6

Conclusiones y trabajo a futuro

6.1 Conclusiones

De acuerdo a la metodología de diseño UEP propuesta en este trabajo de tesis, se probaron dos sistemas, uno basado en códigos de bloque BCH, cuyos parámetros escogidos después de las pruebas realizadas están en la Tabla 4.10, y otro basado en códigos convolucionales, cuyos escogidos al final de las pruebas están en la Tabla 4.42. Estos sistemas fueron aplicados al sistema de comunicaciones propuesto en el capítulo 4, con el objetivo de estudiar su desempeño en términos de BER, SNR y SSIM, por medio de simulaciones en Matlab. De lo anterior, se obtuvo que el desempeño de ambos sistemas UEP en términos de BER fue de $1E-04$, lo cual representa una disminución del BER de $1E-02$ con respecto al sistema sin codificación de canal; con esto se concluyó la funcionalidad de la metodología UEP propuesta.

Por otra parte, se corroboró que el tiempo de ejecución es directamente proporcional a la redundancia agregada a la fuente de datos. Con esto en mente, en el sistema UEP basado en códigos convolucionales, se empleó el concepto de RCPC, del que se obtuvo una disminución de redundancia de casi el 50%, con respecto a la redundancia del esquema EEP, y un ahorro en el tiempo de ejecución del 44%, sin deteriorar la calidad en términos de BER, PSNR y SSIM de la imagen recibida. Por el contrario, con el esquema UEP basado con códigos BCH, se obtuvo un ahorro en el tiempo de ejecución de casi el 25%, en relación al esquema EEP, porcentaje menor al medido con los códigos convolucionales, esto se debe a que con los códigos BCH no se logró reducir más la redundancia promedio, porque a su vez disminuía la calidad visual de la imagen recibida. No obstante, con los códigos BCH se superó a los códigos convolucionales en términos de los valores de calidad, para la misma razón señal a ruido E_b/N_0 .

Con respecto a la validación de la metodología UEP en un escenario de comunicación físico real, también se propusieron dos esquemas UEP, uno basado en códigos BCH, y otros basado en códigos convolucionales, con el objetivo de estudiar su desempeño utilizando los equipos de radio. De manera que, de acuerdo a los resultados resumidos de la Tabla 4.61, el ahorro en tiempo de la transmisión y recepción de la imagen en el esquema UEP, no fue cercano al ahorro de redundancia del código como en los casos simulados. Esta diferencia se explica porque el tiempo de transmisión depende de la tasa de muestreo de los USRP2, la cual se mantuvo en 12.207 [kb/s]. Sin embargo, se logró reducir el tiempo de transmisión en un 20%, con respecto al esquema EEP. Por otro lado, los valores de calidad en el esquema UEP se mantuvieron por encima de los obtenidos sin codificación, con las mismas distancias entre los equipos USRP2, confirmando el buen desempeño obtenido en el sistema simulado.

6.2 Trabajo a futuro

Con base a lo realizado durante este trabajo de tesis, se propone el siguiente trabajo a futuro, con el objetivo de darle continuidad a la línea de investigación:

- El sistema de comunicaciones basado en SDRs implementado en este trabajo, no pudo realizar procesamiento en tiempo real, debido a que Matlab no posee “scripts” que permitieran lograr la sincronía entre los paquetes enviados, y recibidos en el sistema. Además, dentro de la gama de funciones de GNU Radio, no se encuentra disponible el bloque de los códigos BCH, de modo que se tuvo la necesidad de realizar las pruebas por separado en Matlab y GNU Radio. Con Matlab se efectuaron las etapas de la codificación y decodificación de la imagen, mientras que con GNU Radio, se realizó la etapa de la transmisión/recepción de la imagen con el uso de los equipos USRP2. El área de oportunidad para que el sistema de comunicaciones propuesto funcione en tiempo real, sería aprovechar las bondades que posee Matlab con el tratamiento de imágenes y, así mismo, desarrollar “scripts” en Matlab que involucren la sincronía del paquete en la transmisión/recepción con los equipos USRP2.
- Implementar la etapa de codificación de fuente al sistema de comunicaciones propuesto, con la finalidad de comparar su funcionamiento con respecto a los resultados obtenidos de los codificadores de canal empleados en este trabajo, además, estimar nuevas alternativas de diseño UEP.
- Una última propuesta, más ambiciosa, es diseñar un esquema UEP usando turbo códigos, con el propósito de estudiar su desempeño con relación a un sistema sin codificación, en términos de BER y redundancia agregada.

Referencias

- [1] Goldsmith, A. (2005). Wireless communications. *Cambridge university press*.
- [2] Shannon C. E. (1948). A Mathematical Theory of Communication. *Bell Syst. Tech.*, 379-423 (Part I), 623-656 (Part II).
- [3] Anderson, J. B., & Mohan, S. (2012). Source and channel coding: an algorithmic approach (Vol. 150). *Springer Science & Business Media*.
- [4] Borade, S., Nakiboglu, B., & Zheng, L. (2009). Unequal error protection: An information-theoretic perspective. *IEEE Transactions on Information Theory*, 55(12).
- [5] Masnick B. and Wolf J. (1967). On Linear Unequal Error Protection Codes. *IEEE PTransactions on Information Theory*, vol. IT-13, No. 4, 600-607, October 1967.
- [6] Fowdur, T.P., Soyjaudah, K.M.S. (2010). Wireless JPEG Image Transmission Using Multiple Diversity Combining and Unequal Error Protection. *Springer Science+Business Media*, 275-294.
- [7] Qian MAO, Chuan QIN. (2013). A Novel Turbo Unequal Error Protection Scheme for Image Steganography. *Informatica*, vol. 24, No. 4, 561-576, August 2013.
- [8] Qian MAO, Chuan QIN and Xinpeng Zhang. (2010). Image Watermarking Scheme With Unequal Protection Capability Based on Error Correcting Codes. *Journal of multimedia*, vol. 5, No. 5, 427-433, October 2010.
- [9] Xu L., Wang L., Hong S., Wu H. (2014). Results on Radiography Image Transmission with Unequal Error Protection Using Protograph Double LDPC Codes. *IEEE*, 1-4, April 2014.

-
- [10] Khalid A., Khan E., Adebisi B., Honary B., Khan S.U. (2014). Image transmission using unequal error protected multi-fold turbo codes over a two-user power-line binary adder channel. *IET Image Processing*, vol. 9, 395-404, September 2014.
- [11] Iqbal M. I., Zepernick H. (2011). A framework for error protection of region of interest coded images and videos. *Signal Processing: Image Communication*. (26), 236-249, March 2011.
- [12] V. Chande, N. Farvardin, Progressive transmission of images over memoryless noisy channels, *IEEE Journal on Selected Areas in Communications* 18 (6) (2000) 850–860.
- [13] Yuan D., Li Z., Sui A. and Ning J. (2000). Research on unequal error protection with punctured convolutional codes in image transmission system over mobile channels. *IEEE Wireless Communications and Networking Conference*, vol. 3, 1253–1257.
- [14] Condo C., Masera G., Montuschi P. (2015). Unequal Error Protection of Memories in LDPC Decoders. *IEEE Transactions on Computers*, vol. 64, No. 10, 2981-2992, October 2015.
- [15] El-Hajjar M., Nguyen Q. A., Maunder R.G. and Ng S. X. (2014). Demonstrating the Practical Challenges of Wireless Communications Using USRP. *IEEE Communications Magazine*, 195-201, May 2014.
- [16] Fajkus M., Mikulec M., Voznak M., Tomis M., Fazio P. (2014). Speech Quality Measurement of GSM Infrastructure Built on USRP N210 and OpenBTS Project. *Advances in Electrical and Electronic Engineering*, vol. 12, no. 4, 341-346.
- [17] Raschellà A., Umberto A. (2016). Implementation of Cognitive Radio Networks to evaluate spectrum management strategies in real-time. *Computer Communications* 79 (2016) 37-52.
- [18] Tarek, K., Elshabrawy, T., Ashour, M., & Hammad, H. (2016, April). An SDR platform for performance evaluation of OFDM-based indoor DAS solutions. *In 2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)* (pp. 1-6). IEEE.

-
- [19] Hussain, A., Glazunov, A. A., Einarsson, B. P., & Kildal, P. S. (2016). Antenna Measurements in Reverberation Chamber Using USRP. *IEEE Transactions on Antennas and Propagation*, 64(3), 1152-1157.
- [20] Van, D. N., Anh, T. N., Ngoc, T. T., Nguyen, V. D., Jeon, B., & Nguyen, T. H. (2014, January). A real-time COFDM transmission system based on the GNU radio: USRP N210 platform. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication* (p. 61). ACM.
- [21] Lackey, R. I., & Upmal, D. W. (1995). Speakeasy: the military software radio. *IEEE Communications Magazine*, 33(5), 56-61.
- [22] Mitola, J. (1993). Software radios: Survey, critical evaluation and future directions. *IEEE Aerospace and Electronic Systems Magazine*, 8(4), 25-36.
- [23] Mitola, J. (1999). Cognitive radio for flexible mobile multimedia communications. In *Mobile Multimedia Communications, 1999. (MoMuC'99) 1999 IEEE International Workshop on* (pp. 3-10). IEEE.
- [24] Couch, L. W. L. W. (2008). *Sistemas de comunicación digitales y analógicos*. Pearson Educación.
- [25] Sklar, B. (2001). *Digital communications (Vol. 2)*. NJ: Prentice Hall.
- [26] Tomasi, W. (2003). *Sistemas de comunicaciones electrónicas*. Pearson educación.
- [27] Gerez, H. (2013). *Implementation of Digital Signal Processing: Some Background on GFSK Modulation*.
- [28] Haran, O., Ilan, H., & Kaufmann, Y. (2002). U.S. Patent No. 6,438,176. Washington, DC: U.S. Patent and Trademark Office.
- [29] Twu, D. C., & Ku, S. Y. (2002). U.S. Patent Application No. 10/174,877.
- [30] Morelos-Zaragoza, R. H. (2006). *The art of error correcting coding*. John Wiley & Sons.

- [31] Lin, S., & Costello, D. J. (2004). Error control coding. Pearson Education India.
- [32] Waterston, J. W., Wooten, C., Bennett, W., & Welch, T. B. (2000). Development and Implementation of an Adaptive Error Correction Coding Scheme for a Full Duplex Communications Channel. In *Wireless Personal Communications* (pp. 229-237). Springer US.
- [33] Peterson, W. W., & Weldon, E. J. (1972). Error-correcting codes. MIT press.
- [34] Cain, J., Clark, G. C., & Geist, J. (1979). Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding (Corresp.). *IEEE Transactions on Information Theory*, 25(1), 97-100.
- [35] Hagenauer, J. (1988). Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE transactions on communications*, 36(4), 389-400.
- [36] Liu, H., Ma, H., El Zarki, M., & Gupta, S. (1997). Error control schemes for networks: An overview. *Mobile Networks and Applications*, 2(2), 167-182.
- [37] Lee, L. C. (1994). New rate-compatible punctured convolutional codes for Viterbi decoding. *IEEE Transactions on Communications*, 42(12), 3073-3079.
- [38] Shahhosseini, S., & Hosseini, S. A. (2010, December). A visual base image coding using unequal error protection in high error rate channels. In *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on* (pp. 537-542). IEEE.
- [39] Grayver, E. (2012). Implementing software defined radio. Springer Science & Business Media.
- [40] Pinar, I., Murillo J. (2011). Laboratorio de Comunicaciones Digitales Radio Definida por Software. Universidad de Sevilla.
- [41] Ettus Research. (2016). USRP N210. 26-08-2016, de National Instruments Company Sitio web: <https://www.ettus.com/product/details/UN210-KIT>.

- [42] Ettus Research. (2016). USRP N200/N210, Networked series. 26-08-2016, de National Instruments Company Sitio web: https://www.ettus.com/content/files/07495_Ettus_N200-210_DS_Flyer_HR_1.pdf.
- [43] Ettus Research. (2016). SDR Software. 26-08-2016, de National Instruments Company Sitio web: <https://www.ettus.com/sdr-software>.
- [44] Olivieri, S. J., Aarestad, J., Pollard, L. H., Wyglinski, A. M., Kief, C., & Erwin, R. S. (2012, June). Modular FPGA-based software defined radio for CubeSats. In 2012 IEEE International Conference on Communications (ICC) (pp. 3229-3233). IEEE.
- [45] Ettus Research. Selecting an RF Daughterboard. 29-08-2016, de National Instruments Company Sitio web: https://www.ettus.com/content/files/kb/Selecting_an_RF_Daughterboard.pdf.
- [46] Ettus Research. Selecting a USRP Device. 30-08-2016, de National Instruments Company Sitio web: https://www.ettus.com/content/files/kb/application_note_selecting_a_usrp.pdf.
- [47] Redmine. (2013). What is GNU Radio and why do I want it?. 30-08-2016, de GNURadio Sitio web: <http://gnuradio.org/redmine/projects/gnuradio/wiki/WhatIsGR>.
- [48] Amado Gutiérrez Gómez. (2014). Diseño del nivel físico de un radio OFDM para comunicación digital con una plataforma software defined radio (SDR) (tesis de maestría). Universidad Autónoma Metropolitana-Iztapalapa, México.
- [49] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600-612.
- [50] Zanella, A. (2007). Analysis of the packet reception statistics of Bluetooth v2+ EDR in fading channels.
- [51] Frenger, P., Orten, P., & Ottosson, T. (1999). Convolutional codes with optimum distance spectrum. *IEEE Communications Letters*, 3(11), 317-319.

- [52] Daut, D., Modestino, J., & Wismer, L. (1982). New short constraint length convolutional code constructions for selected rational rates (Corresp.). *IEEE Transactions on Information Theory*, 28(5), 794-800.