

GRADO EN INGENIERÍA EN TECNOLOGÍA DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

*ANÁLISIS E IMPLEMENTACIÓN
DE UNA NUEVA CODIFICACIÓN
LDPC EN SISTEMAS DE
COMUNICACIONES POR SATÉLITE*

Alumno: Maseda García, Jon

Director: Angueira Buceta, Pablo

Codirector: Ordiales Basterrechea, Juan Luis

Curso: 2017-2018

Fecha: 29 de junio de 2018

ÍNDICE

1) Lista de tablas, figuras y acrónimos	5
1.1) Tablas	5
1.2) Figuras	6
1.3) Acrónimos	7
2) Resumen	8
3) Introducción	10
4) Contexto	11
5) Objetivos y Alcance	13
5.1) Objetivos	13
5.2) Alcance.....	13
6) Beneficios	15
6.1) Beneficios técnicos.....	15
6.2) Beneficios económicos.....	15
6.3) Beneficios sociales.....	16
7) Estado del arte	17
7.1) Códigos LDPC	17
7.2) Estándar DVB-S2.....	21
7.2.1) Modos de funcionamiento.....	21
7.2.2) Arquitectura de DVB-S2.....	22
7.3) Estándar 5G - NR	24
8) Metodología	27
8.1) Familias de códigos LDPC propuestos en 5G	27
8.2) Codificación del modelo de referencia DVB-S2 en MATLAB.....	28
8.2.1) Cadena simple.....	29
8.2.2) Primeras simulaciones	32
8.2.3) Mejora del código.....	33
8.3) Codificación del código candidato seleccionado en MATLAB	36

8.4) Integración del nuevo módulo LDPC en el modelo de referencia DVB-S2.....	39
8.5) Caso práctico: simulación y resultados obtenidos.....	40
9) Planificación	43
9.1) Equipo de trabajo.....	43
9.2) Fases del trabajo	43
9.3) Diagrama de Gantt.....	47
10) Presupuesto y costes.....	48
10.1) Horas internas	48
10.2) Amortizaciones	48
10.3) Gastos totales	49
11) Análisis de riesgos	50
11.1) Identificación y evaluación de los riesgos	50
11.2) Planes de contingencia.....	51
12) Conclusiones y visión de futuro.....	53
Referencias	55
Anexo 1: Código MATLAB.....	58
1.1) Parámetros BCH.....	58
1.2) Bloques de la arquitectura de DVB-S2.....	59
1.3) Cadena DVB-S2 simple.....	61
1.4) Cadena DVB-S2 mejorada	63
1.5) Nueva codificación LDPC	66
1.6) Integración del nuevo módulo LDPC en DVB-S2.....	66
Anexo 2. Resultados adicionales.....	72
Anexo 3: Rendimiento ideal en DVB-S2.....	74

1) LISTA DE TABLAS, FIGURAS Y ACRÓNIMOS

1.1) Tablas

Tabla 1. Características de DVB-S2.....	24
Tabla 2. Características de los códigos LDPC en 5G NR.....	28
Tabla 3. Parámetros del codificador BCH.....	30
Tabla 4. Parámetros del entrelazador.....	31
Tabla 5. Parámetros de las primeras simulaciones.....	32
Tabla 6. Resultados de las primeras simulaciones.....	33
Tabla 7. Parámetros de la simulación con código mejorado.....	34
Tabla 8. Resultados de la segunda simulación.....	34
Tabla 9. Parámetros del nuevo entrelazador.....	40
Tabla 10. Casos de estudio.....	40
Tabla 11. Valores de E_s/N_0 en QPSK 1/2.....	41
Tabla 12. Valores de E_s/N_0 en QPSK 3/4.....	41
Tabla 13. Equipo de trabajo.....	43
Tabla 14. PT0. Coordinación, gestión y supervisión del proyecto.....	43
Tabla 15. PT1. Planificación del proyecto.....	43
Tabla 16. PT2. Estudio del estado del arte.....	44
Tabla 17. PT3. Profundización en el programa MATLAB.....	44
Tabla 18. PT4. Desarrollo de la aplicación.....	45
Tabla 19. PT5. Simulación y obtención de curvas de rendimiento.....	45
Tabla 20. PT6. Informes finales.....	45
Tabla 21. Entregables.....	46
Tabla 22. Hitos.....	46
Tabla 23. Horas internas.....	48
Tabla 24. Amortizaciones.....	49
Tabla 25. Presupuesto completo.....	49
Tabla 26. Parámetros simulación 8PSK.....	72
Tabla 27. Parámetros simulación 16APSK.....	72
Tabla 28. Parámetros simulación 32APSK.....	73

1.2) Figuras

Figura 1: Sistema de distribución de TV en Europa en 2017 (Fuente: [23]).....	12
Figura 2. Cadena de codificación y corrección FEC.....	17
Figura 3. Bloque LDPC	17
Figura 4. Ejemplo de matriz \mathbf{H}	19
Figura 5. Ejemplo de grafo bipartido	19
Figura 6. Esquema de la cadena de transmisión de DVB-S2	22
Figura 7. Formato de la trama en banda base (BBFRAME).....	22
Figura 8. Constelación de las modulaciones QPSK, 8PSK, 16APSK y 32APSK.....	23
Figura 9. Plan de desarrollo de 5G NR (Fuente: [19]).....	25
Figura 10. Estructura de la matriz LDPC en 5G NR (Fuente: [13]).....	26
Figura 11. Cadena de transmisión DVB-S2 completa.....	29
Figura 12. Detalle del entorno de trabajo de MATLAB para la primera simulación....	32
Figura 13. Relación BER vs. E_s/N_0	35
Figura 14. Constelación QPSK en el transmisor.....	35
Figura 15. Constelación QPSK en el receptor	35
Figura 16. Proceso de codificación y decodificación del código candidato	37
Figura 17. Arquitectura del nuevo sistema creado.....	39
Figura 18. Relación BER vs. E_s/N_0 para QPSK 1/2 en los dos entornos.....	41
Figura 19. Relación BER vs. E_s/N_0 para QPSK 1/3 en los dos entornos.....	42
Figura 20. Diagrama de Gantt	47
Figura 21. Matriz de riesgos.....	51
Figura 22. Relación BER vs. E_s/N_0 (8PSK)	72
Figura 23. Relación BER vs. E_s/N_0 (16PSK)	73
Figura 24. Relación BER vs. E_s/N_0 (32APSK).....	73

1.3) Acrónimos

3G	Third Generation
3GPP	3 rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
8PSK	8 Phase Shift Keying
16APSK	16 Amplitude and Phase Shift Keying
32APSK	32 Amplitude and Phase Shift Keying
ACM	Adaptative Coding Modulation
AWGN	Additive White Gaussian Noise
BB	Base Band
BCH	Bose–Chaudhuri–Hocquenghem
BER	Bit Error Rate
BP	Believe Propagation
CCM	Constant Coding and Modulation
CRC	Cyclic Redundancy Check
DVB	Digital Video Broadcasting
DVB-S	DVB by Satellite
DVB-S2	DVB-S – Second Generation
DVB-T2	DVB – Terrestrial 2
DTH	Direct To Home
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
GPU	Graphics Processing Unit
ISI	Intersymbol Interference
LDPC	Low-Parity-Check-Codes
LTE	Long Term Evolution
MIT	Massachusetts Institute of Technology
MPEG	Moving Pictures Expert Group
MPEG-TS	MPEG Transport Stream
NR	New Radio
PER	Packet Error Rate
PL	Physical Layer
QPSK	Quadrature Phase Shift Keying
SES	Sociedad Europea de Satélites
VCM	Variable Coding and Modulation

2) RESUMEN

Las transmisiones vía satélite constituyen hoy en día una de las principales fuentes de recepción de televisión digital en Europa. Los estándares definidos necesitan de una constante evolución para implementar las nuevas técnicas que permitan mejorar tanto la eficiencia como la calidad de recepción. En este sentido, el proceso de codificación y decodificación de las señales constituye la etapa más importante de la comunicación, ante la necesidad de proteger y corregir los eventuales errores producidos por el ruido radioeléctrico.

El objetivo final de este proyecto es analizar alternativas y proponer una solución que mejore el bloque de codificación LDPC (*Low-Parity-Check-Codes*) en sistemas de comunicaciones por satélite, teniendo en cuenta las mejoras introducidas por 5G (*Fifth Generation*) y tomando como referencia de trabajo el sistema DVB-S2 (*Digital Video Broadcasting by Satellite – Second Generation*). Para ello, se desarrolla un programa bajo MATLAB que permite simular la nueva solución propuesta y se analizan los resultados obtenidos de cara a evaluar la idoneidad de la misma.

Satelite bidezko transmisioak European telebista digitalaren iturri nagusienetarako bat dira gaur egun. Definitutako estandarrek etengabeko eboluzioa beharrezkoa dute bai efizientzia eta bai seinalearen hartze kalitatea hobetuko dituzten teknikak ezartzeko helburuarekin. Ildo horretan, seinaleen kodetze- eta deskodetze prozesua komunikazioaren urratsik garrantzitsua da, zarata erradioelektrikoak sortutako erroreez babesteko beharra dela eta.

Proiektu honen helburua alternatibak bilatzea eta satelite bidezko komunikazio sistemetan LDPC (*Low-Parity-Check-Codes*) kodifikazio-blokea hobetuko duen soluzio bat proposatzea da, 5G (*Fifth Generation*)-k ekarritako hobekuntzak kontuan hartuz eta DVB-S2 (*Digital Video Broadcasting by Satellite – Second Generation*) sistema hartuz lan-erreferentzia gisa. Horretarako, proposatutako soluzio berria simulatzeko programa bat garatuko da MATLAB bidez; eta lortutako emaitzak aztertuko dira honen egokitasuna ebaluatzeko helburuaz.

Satellite transmissions are currently one of the main sources of reception of digital television in Europe. The defined standards need a constant evolution to take into consideration new techniques that improve the efficiency and quality of reception. On this matter, the process of coding and decoding signals represents the most important stage of a communications system as it is necessary to protect and correct errors produced by interferences.

The ultimate goal of this project is to analyze and propose a new solution that improves the LDPC (*Low-Parity-Check-Codes*) coding block in satellite communications systems taking the standard DVB-S2 (*Digital Video Broadcasting by Satellite – Second Generation*) and the improvements introduced by 5G (*Fifth Generation*) as a reference. To do this, a program is developed in MATLAB to simulate the new solution. Furthermore, a complete analysis of the results is done in order to evaluate its efficiency.

Les transmissions par satellite constituent aujourd’hui une des principales sources de réception de télévision numérique en Europe. Les standards définis nécessitent d’une évolution constante pour intégrer les nouvelles techniques qui permettent améliorer l’efficacité et la qualité de réception. Dans ce sens, le processus de codage et de décodage des signaux constitue l’étape la plus importante dans une chaîne de communication, en vue du besoin de protéger et de corriger les éventuelles erreurs produites par le bruit radioélectrique.

L’objectif final de ce projet est d’analyser des alternatives et de proposer une solution qui puisse faire améliorer le bloc de codage LDPC (*Low-Parity-Check-Codes*) dans des systèmes de communications par satellite, en tenant en compte les améliorations introduites par la 5G (*Fifth Generation*) et en utilisant le standard DVB-S2 (*Digital Video Broadcasting by Satellite – Second Generation*) comme système de travail. Pour cela, un programme est développé sous MATLAB pour simuler la nouvelle solution proposée et permettre analyser les résultats obtenus afin d’évaluer son adéquation.

3) INTRODUCCIÓN

A lo largo de este documento, se presenta la memoria del proyecto de Fin de Grado cuya finalidad consiste en el análisis e implementación de una nueva codificación LDPC en un sistema de comunicaciones por satélite, tomando como referencia de trabajo el estándar DVB-S2.

La realización del proyecto se ha llevado a cabo de manera conjunta con el grupo de investigación TSR (Tratamiento de la Señal y Radiocomunicaciones) de la Escuela de Ingeniería de Bilbao.

En primer lugar se realizará un estudio teórico de las áreas de conocimiento necesarias para acotar el problema y lograr, posteriormente, el diseño de la solución planteada. Cabe destacar el análisis de los códigos LDPC, del estándar DVB-S2 y de las familias de códigos propuestas para el nuevo estándar 5G *New Radio*.

A continuación, se detallará el proceso de desarrollo realizado con la herramienta MATLAB de la nueva codificación LDPC y su posterior integración en el modelo de referencia DVB-S2. Asimismo, se simulará un caso práctico en dos escenarios diferenciados: por una parte, se tomará exclusivamente el estándar DVB-S2 original y, por otra parte, el nuevo modelo creado. Será la base para analizar los resultados obtenidos en cada escenario y así evaluar la idoneidad de la solución planteada.

Asimismo, se detallarán los beneficios que aporta el proyecto en distintos contextos. Se proporcionará la información requerida para valorar la viabilidad del proyecto indicando su organización y planificación, así como el presupuesto económico realizado y su análisis de riesgos.

Finalmente, este documento contará con varios anexos que incluirán el código de las funciones realizadas en MATLAB, así como de los resultados obtenidos en las simulaciones realizadas y el modelo teórico ideal de rendimiento en DVB-S2.

4) CONTEXTO

Los códigos LDPC surgen, por primera vez, en 1960 tras el desarrollo de la tesis doctoral de R. Gallager en el MIT (*Massachusetts Institute of Technology*). [4] Su gran ventaja radica en que permiten realizar comunicaciones con eficiencias muy cercanas al límite establecido por Shannon. [7] No obstante, debido a la complejidad excesiva de su decodificación, no se utilizaron como una técnica de protección de datos hasta la década de los años 90.

La primera introducción de los códigos LDPC se realizó en la segunda generación del estándar de transmisión de televisión digital vía satélite (DVB-S2). Su uso permitió contribuir al aumento de la eficiencia en la transmisión de televisión por satélite en un 30% respecto a la generación anterior (DVB-S). Sin embargo, su implementación data de 2003 y la codificación utilizada puede, en algunos casos, considerarse ineficiente si se tiene en cuenta la constante evolución de los códigos LDPC.

Por otra parte, en el sector de la telefonía móvil, se descartó la utilización de este tipo de codificación en 3G y 4G en detrimento del uso de turbo-códigos. La razón de ello radicaba en que, si bien el uso LDPC permitía mejorar la eficiencia de las transmisiones, los retardos introducidos por los algoritmos iterativos en el proceso de decodificación eran demasiado importantes como para ser considerados. Los turbo-códigos, por su parte, permitían decodificar los paquetes de datos transmitidos de manera rápida, reduciendo así el retardo en la transmisión de paquetes de datos.

El nuevo estándar de telefonía móvil 5G, definido por 3GPP en su versión NR (*New Radio*) en octubre de 2017 [19], contempla, por primera vez, el uso de unas nuevas familias de códigos LDPC en este tipo de transmisiones. La motivación principal de este cambio surge por la necesidad de incrementar el ancho de banda de transmisión respecto a generaciones anteriores hasta los 20 Gbps.

En este proyecto se combina por un lado la codificación LDPC utilizada en 5G NR y por otro, el estándar DVB-S2, con el objetivo de crear un nuevo bloque de codificación que recoja las nuevas familias de códigos propuestos y evaluar su implementación en una cadena de transmisión por satélite DTH (*Direct to Home*).

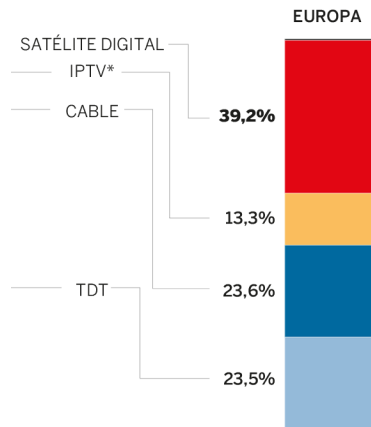


Figura 1: Sistema de distribución de TV en Europa en 2017 (Fuente: [23])

En efecto, hay que comentar que, según el estudio *Satellite Monitors 2017* [23] de la Sociedad Europea de Satélites [Fig. 1], el satélite constituye, hoy en día, la principal fuente de recepción de la televisión en Europa con un 39% de uso. Esto muestra la relevancia que tienen las transmisiones DTH en la población europea y justifica la necesidad de valorar nuevas propuestas que permitan mejorar la eficiencia de dichas transmisiones.

5) OBJETIVOS Y ALCANCE

En este apartado, se resumen el objetivo principal y los secundarios, así como el alcance de este trabajo.

5.1) Objetivos

El objetivo principal de este proyecto es analizar alternativas y proponer un modelo que mejore el bloque de codificación de canal en sistemas de comunicaciones por satélite, tomando como referencia de trabajo el sistema DVB-S2.

Para alcanzar el objetivo principal, se han definido varios objetivos secundarios que se detallan a continuación:

- Analizar el estado del arte de los códigos LDPC, con especial atención a la familia de códigos actualmente en estudio en 3GPP.
- Evaluar la compatibilidad de la estructura de la capa física de DVB-S2.
- Definir las especificaciones técnicas que impone cualquier modificación del bloque de codificación.
- Evaluar el rendimiento de un conjunto limitado de códigos y configuraciones de códigos a través de una implementación en MATLAB.
- Integrar el módulo implementado en el modelo de referencia de MATLAB para DVB-S2.
- Evaluar la posible mejora de rendimiento del nuevo modelo de codificación respecto de los estándares actuales DVB-S2.

5.2) Alcance

El alcance final de este proyecto consiste en obtener un programa realizado en MATLAB que permita simular de forma fiable una cadena típica de transmisión DTH con un nuevo modelo de codificación propuesto y evaluar su eficiencia.

Para ello, se detallará el proceso de desarrollo realizado con la herramienta MATLAB tanto del modelo de referencia DVB-S2 como del código candidato seleccionado. Además, se considerará un caso práctico y se analizarán los resultados obtenidos para evaluar la idoneidad de la solución planteada.

6) BENEFICIOS

En esta sección, se describen los beneficios que aporta este proyecto desde el punto de vista técnico, económico y social.

6.1) Beneficios técnicos

El beneficio técnico principal de la solución que se plantea radica en el diseño de un programa que permite simular una nueva cadena de transmisión en comunicaciones por satélite. Destaca, en este sentido, la creación de un nuevo modelo de transmisión no definido hasta la fecha que permite introducir las últimas evoluciones desarrolladas en el ámbito de los códigos LDPC y, más en particular, de la nueva codificación propuesta por 3GPP para el estándar 5G NR.

El programa desarrollado permite obtener el rendimiento de la solución propuesta y, además, evaluar y comparar su rendimiento respecto al modelo utilizado en la actualidad. De esta manera, es posible comprobar la idoneidad de la misma y las mejoras que aporta dentro de los casos prácticos que se detallan en esta memoria y plantear en un futuro nuevos casos de estudio ampliando la herramienta creada.

Asimismo, es preciso comentar que las simulaciones desarrolladas se basan en un modelo de propagación que tiene en cuenta la presencia de ruido radioeléctrico, tal y como ocurriría en un entorno real. Por este motivo, este proyecto podrá ser completado más adelante con un estudio que permitiese implementar de forma física la nueva codificación propuesta, a nivel de hardware, en un bloque de transmisión/receptor. De esta forma, sería posible considerar un escenario real que permitiese validar la nueva codificación.

6.2) Beneficios económicos

Los beneficios económicos de este proyecto se obtienen principalmente en la eventual implementación de la solución propuesta en la vida real. En efecto, una mejora en la eficiencia de las transmisiones de comunicaciones vía satélite conllevaría una reducción del consumo eléctrico empleado en cada transmisor manteniendo la misma calidad de recepción. Esto permitiría una reducción de costes en el mantenimiento de la

infraestructura de los satélites existentes que son utilizados en la actualidad para las comunicaciones DTH.

6.3) Beneficios sociales

El impacto directo de este proyecto es sin duda sobre la sociedad en su conjunto. Como se ha comentado, según la SES [23], la transmisión por satélite representa en la actualidad la principal fuente de recepción de televisión digital en Europa.

Por este motivo, uno de los beneficios directos de modificar el bloque de codificación en este tipo de transmisiones es la obtención de una posible mejora en la calidad de recepción de las mismas. De esta manera, el usuario podría seguir visualizando correctamente la televisión a pesar de encontrarse en un entorno en el que el ruido radioeléctrico produjese errores e interferencias en la comunicación.

7) ESTADO DEL ARTE

En este apartado se realiza un estudio teórico de las áreas de conocimiento necesarias para acotar el problema y lograr, posteriormente, el diseño de la solución planteada. En este sentido, se detalla el principio de funcionamiento de los códigos LDPC, de los estándares DVB-S2 y 5G NR.

7.1) Códigos LDPC

Como se ha comentado en apartados anteriores, los códigos LDPC surgen por primera vez en 1960 tras el desarrollo de la tesis doctoral de R. Gallager en el MIT. [4] Se trata de códigos correctores de errores de tipo FEC (*Forward Error Correction*) que se basan en reglas de paridad simples dentro de grandes bloques de datos. Su objetivo es proteger la información a transmitir contra las pérdidas producidas en el canal de transmisión para poder, posteriormente, recuperarla en recepción.

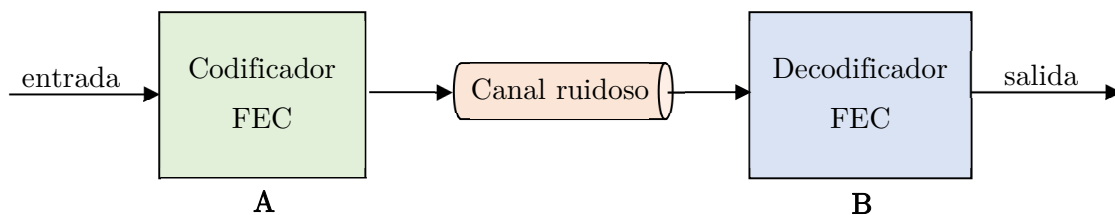


Figura 2. Cadena de codificación y corrección FEC.

Así, en el extremo A (transmisión) se generan unos datos de protección (paridad) que constituyen una información redundante añadida a la original y que, en caso de error, permite reconstruirla o, en su defecto, conocer la existencia de dicho error. El bloque de información codificado tiene un tamaño de N_{LDPC} bits y está compuesto por K_{LDPC} bits de información al que se añaden $N_{LDPC} - K_{LDPC}$ bits de paridad, tal y como se esquematiza en la siguiente figura:

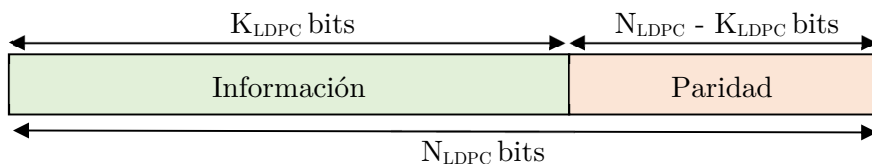


Figura 3. Bloque LDPC

Los códigos LDPC son considerados lineales dado que cada palabra codificada $c = [c_0, c_1, \dots, c_{n-1}]$ puede obtenerse realizando el producto entre el vector de la palabra original $u = [u_0, u_1, \dots, u_{k-1}]$ y la matriz generadora \mathbf{G} de dimensiones $(K_{LDPC} \times N_{LDPC})$.

$$c = u \cdot G$$

Además, los códigos LDPC pueden ser definidos como sistemáticos ya que, como puede comprobarse en la figura [Fig. 3], cada palabra codificada se obtiene anexando la información de paridad a la palabra original. Cada bit de paridad está, de esta manera, asociado a una ecuación de comprobación de paridad. Por tanto, la matriz generadora \mathbf{G} puede definirse según la siguiente forma:

$$G = [I \mid P]$$

Donde \mathbf{I} representa la matriz de identidad de dimensiones $(K_{LDPC} \times K_{LDPC})$ y \mathbf{P} , de dimensiones $K_{LDPC} \times (N_{LDPC} - K_{LDPC})$ representa el sistema de ecuaciones de comprobación de paridad.

Asimismo, un código LDPC puede ser definido por la matriz de comprobación de paridad \mathbf{H} , de tal forma que cada palabra codificada c debe satisfacer la ecuación:

$$H \cdot c^T = 0, \text{ donde } H = [P^T \mid B]$$

La matriz de comprobación de paridad \mathbf{H} constituye un elemento esencial en el proceso de codificación LDPC. En este caso, al tratarse de códigos binarios, los elementos en \mathbf{H} sólo pueden tomar dos valores: 0 y 1.

Esta matriz \mathbf{H} puede representarse en la forma de un grafo bipartido de Tanner. En un lado del grafo, se sitúan los bits de un bloque LDPC (de tamaño N_{LDPC}) que representan los nodos variables del sistema V , mientras que, en el otro lado del mismo, se sitúan los nodos de comprobación C (de tamaño $N_{LDPC} - K_{LDPC}$).

Por ejemplo, para el caso de un código con $N_{LDPC} = 7$ y $N_{LDPC} - K_{LDPC} = 3$, \mathbf{H} podría ser de la forma:

$$\begin{array}{c}
 \overbrace{\hspace{10em}}^{N_{LDPC} = 7 \text{ bits}} \\
 H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{array}{l} \overbrace{\hspace{1em}}^{N_{LDPC} - K_{LDPC} = 3 \text{ bits}} \\ \underbrace{\hspace{1em}} \\ \underbrace{\hspace{1em}} \end{array} \\
 \underbrace{\hspace{3em}}_{\mathbf{P}^T} \quad \underbrace{\hspace{3em}}_{\mathbf{B}}
 \end{array}$$

Figura 4. Ejemplo de matriz \mathbf{H}

Las columnas de la matriz \mathbf{H} equivalen a cada nodo variable V y las filas corresponden a cada nodo de comprobación C . Los valores a “1” indican una conexión entre ambos nodos, que puede representarse como una línea del grafo. De esta forma, y para el ejemplo descrito en la figura [Fig. 4], el grafo de Tanner asociado sería el siguiente:

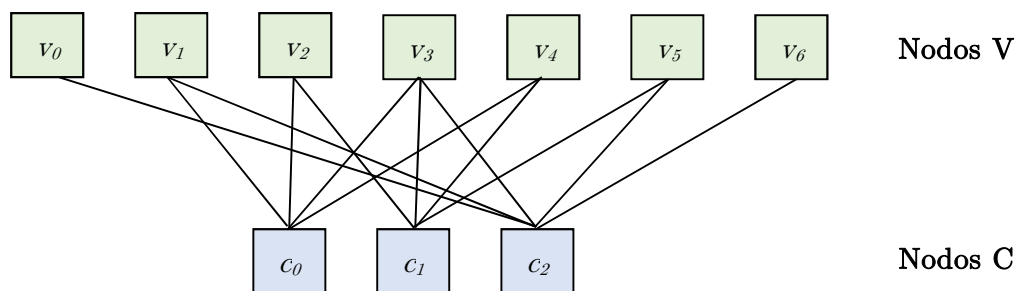


Figura 5. Ejemplo de grafo bipartido

Las matrices definidas en los estándares de DVB siguen las siguientes reglas:

- La submatriz (\mathbf{B}) tiene que ser de forma cuadrada y tener, al menos, un “1” en cada línea.
- De igual manera, en la submatriz (\mathbf{B}), no puede haber ningún “1” a la derecha del primer “1” de cada columna.

Los bloques LDPC pueden ser codificados siguiendo las mismas técnicas que para los otros códigos de bloque, y la complejidad de la codificación dependerá de la técnica utilizada. La propuesta realizada por Gallager consiste en aplicar dichos algoritmos, pero en códigos de gran longitud (más de 10.000 bits por bloque).

Si se considera, de nuevo, el ejemplo anterior, en el que $N_{LDPC} = 7$ y $K_{LDPC} = 4$, se puede codificar la palabra $u = [1 \ 1 \ 0 \ 0]$ empleando la expresión $c = u \cdot G$. Para ello, hay que obtener en primer lugar la matriz generadora \mathbf{G} a partir de \mathbf{H} . Puede deducirse, por tanto, lo siguiente:

$$P^T = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \Rightarrow P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \Rightarrow G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

De tal forma que la palabra codificada será:

$$c = [1 \ 1 \ 0 \ 0] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = \underline{[1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]}$$

El proceso de decodificación en este caso sería sencillo, pues bastaría con comprobar que la palabra código satisface la ecuación $H \cdot c^T = 0$.

$$H \cdot c^T = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \underline{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}$$

Esta técnica de comprobación de errores mediante comparación directa de la palabra código recibida sólo es posible utilizarla cuando K_{LDPC} es pequeño. En el caso de los códigos LDPC, es impensable su uso pues el tamaño de los bloques es elevado (más de 10.000 bits por lo general) por lo que se hace necesaria la utilización de algoritmos de tipo paso-de-mensaje para su decodificación.

En el grafo de Tanner, los nodos V están interconectados entre sí de manera indirecta (se puede ir de un nodo hacia otro a través de los nodos C), como se puede observar en la figura [Fig. 5]. Los algoritmos de tipo paso-de-mensaje son iterativos, de manera que se envían mensajes entre los nodos V y C hasta que se realiza la comprobación de cada nodo.

Esta fue una de las razones por las que se abandonaron los códigos LDPC dado que la realización de las iteraciones era demasiado compleja y requería de un tiempo considerable para la comprobación de cada bloque LDPC. Por este motivo, en la década de los 90 se definió la técnica *Believe Propagation* (BP) [1] que permite realizar una decodificación sencilla.

La técnica BP consiste en obtener los nodos con valores incorrectos e intentar corregirlos. Para lograrlo, se calcula su valor más probable en comparación con los otros nodos del grafo. Se realizan únicamente tres iteraciones en este caso, por lo que su implementación es más sencilla.

7.2) Estándar DVB-S2

El estándar DVB-S2 fue ratificado por la ETSI en 2005 y constituye la segunda generación del estándar de transmisión digital vía satélite DVB-S. [17] Los objetivos que fundamentaron su creación pueden resumirse en los siguientes puntos:

- Prestación de transmisión optimizada en cuanto a rendimiento
- Flexibilidad total de formatos de entrada
- Complejidad del decodificador aceptable
- Compatibilidad con el estándar anterior (DVB-S)

7.2.1) Modos de funcionamiento

DVB-S2 puede operar en tres modos distintos: CCM, VCM y ACM.

El primer modo de funcionamiento, que ya ofrecía DVB-S, es el de codificación y modulación constante CCM (*Constant Coding and Modulation*). La modulación utilizada es fija y la codificación del canal se mantiene constante a lo largo del tiempo, dotando de un nivel de protección fijo al sistema.

El segundo modo de funcionamiento es el de codificación y modulación variable VCM (*Variable Coding and Modulation*). Este modo puede utilizarse para proveer diferentes servicios en la misma portadora, dotando a cada uno de ellos de su propia modulación y nivel de protección frente a errores.

El último modo de funcionamiento, ACM (*Adaptive Coding Modulation*) extiende el modo VCM utilizando un canal de retorno.

7.2.2) Arquitectura de DVB-S2

Resulta preciso comentar de manera detallada la arquitectura de DVB-S2 pues ésta constituye la referencia en la que se ha basado este proyecto.

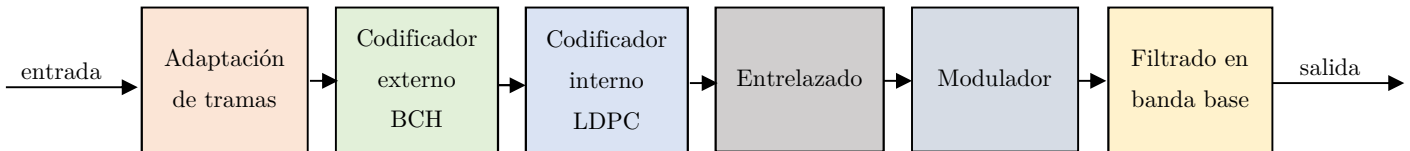


Figura 6. Esquema de la cadena de transmisión de DVB-S2

En primer lugar, se genera la trama en banda base (BBFRAME), en la que se ha encapsulado los datos de entrada junto a una cabecera de señalización (BBHEADER) para permitir la correcta sincronización y configuración del receptor. Son tramas de longitud constante por lo que se introducirán tantos ceros como sea necesario en caso de paquetes más reducidos (*padding*).

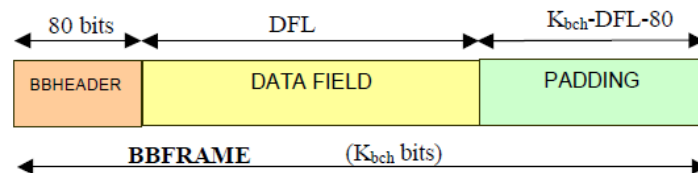


Figura 7. Formato de la trama en banda base (BBFRAME)

La detección y corrección de errores consiste una de las etapas más importantes en la cadena de transmisión de cualquier estándar. Si el objetivo principal de DVB-S2 es aumentar la eficiencia a la vez que se dota de robustez al sistema, resulta primordial tener la capacidad de detectar y corregir el mayor número de errores posible todo ello manteniendo un rendimiento útil elevado.

En este caso, se concatena un código externo BCH (*Bose-Chaudhuri-Hocquenghem*) con un código interno LDPC. Los códigos BCH forman parte de los códigos de corrección cíclicos. La particularidad de estos códigos radica en el control preciso del número de bits corregibles por el código, la capacidad de corrección de múltiples errores en una misma palabra código y su decodificación sencilla mediante el uso de síndromes.

Además, se aplica un entrelazado para dispersar posibles ráfagas de error en distintas tramas posibilitando así su corrección.

En el proceso de mapeado, se han incluido tres nuevos tipos de modulaciones (8PSK, 16APSK, 32APSK) respecto a la modulación QPSK ya existente en DVB-S.

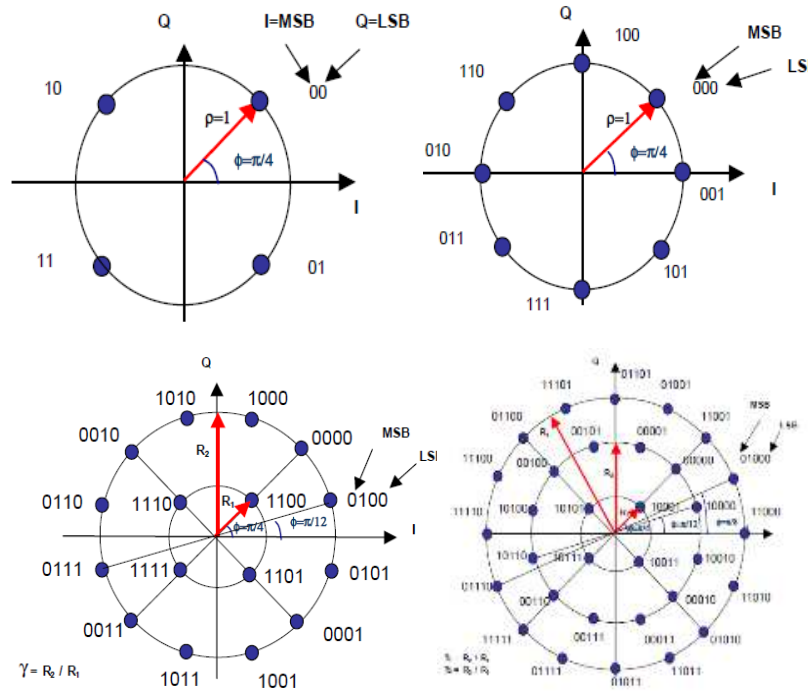


Figura 8. Constelación de las modulaciones QPSK, 8PSK, 16APSK y 32APSK

No obstante, aumentar la eficiencia espectral y por tanto, la constelación utilizada, implica reducir el área de decisión de cada símbolo. Para el ejemplo de 32APSK, el área de decisión puede llegar a ser de $\pi/12$ frente a $\pi/2$ en QPSK (6 veces menor).

A continuación, se encapsulan los datos en tramas de nivel físico (PLFRAME). Hay que comentar que, cuando no hay ninguna trama para transmitir, se introducen tramas de relleno (*Dummy*) para garantizar un flujo constante de bits en salida. Asimismo, y al igual que en el caso de las tramas en banda base, se introduce una cabecera (PLHEADER) para la correcta configuración y sincronización del receptor. La aleatorización de los símbolos modulados permite repartir de una forma más equitativa la energía en cada punto de la constelación.

Por último, se realiza el filtrado en coseno alzado de señal generada de manera a limitarla en banda y reducir así la interferencia entre símbolos (ISI). De nuevo, para

aumentar la eficiencia, DVB-S2 introduce dos nuevos factores de roll-off ($\alpha = 0.20$ y 0.25) respecto a DVB-S ($\alpha = 0.35$).

En la siguiente tabla se resumen las características de las transmisiones en DVB-S2:

Tabla 1. Características de DVB-S2

	DVB-S2
Códec de vídeo	MPEG-2, MPEG-4
Códec de audio	AAC, AC-2, MPEG1-L2
Multiplexación	MPEG-TS, ATM, IP...
Tamaño de los paquetes	Variable
Código externo	BCH variable
Código interno	LDPC
<i>Code rate</i>	1/4, 1/3, 2/5, 1/2, 3/5, 2/3, 3/4, 4/5, 5/6, 8/9 y 9/10
Ancho de banda del transpondedor	Variable
Roll-off	0.20, 0.25, 0.35 (coseno alzado)
Esquema de modulación	QPSK, 8PSK, 16APSK o 32APSK

Asimismo, cabe destacar que las tramas de DVB-S2 pueden ser de dos tamaños diferentes: cortas (16200 bits) o largas (64800 bits). Por lo general, en el caso de transmisiones DTH, el tamaño de las tramas utilizado es el largo pues permite enviar mayor información útil por trama.

7.3) Estándar 5G - NR

El estándar 5G NR, sucesor del actual 4G, define la quinta generación de tecnologías utilizadas en telefonía móvil. Entre los objetivos que persigue esta nueva tecnología, destacan:

- Tasa de transferencia de datos pico de 20 Gbps
- Latencia en las transmisiones de 1 ms (en el 4G actual, se sitúa en torno a los 50 ms)

- Conexión de hasta 10^6 dispositivos por km^2
- Mejora en la eficiencia espectral respecto a la tecnología anterior (entre 3 y 4 veces superior).
- Eficiencia energética similar a 4G
- Trama física (PL) con diseño flexible para dar soporte a multitud de aplicaciones, algunas de ellas no definidas en la actualidad.

Se debe señalar la reciente aparición de este nuevo estándar. En efecto, 3GPP definió el plan de desarrollo de 5G en el último trimestre del año 2016 y, en octubre de 2017, se publicaron, por primera vez, las especificaciones de la versión *Stand Alone* (incompatible con el actual estándar LTE). Por otra parte, hay que comentar que se prevé que la versión *Non Stand Alone* se publique en el segundo semestre del año 2018, de forma que el primer uso comercial de esta tecnología pueda realizarse en 2020.

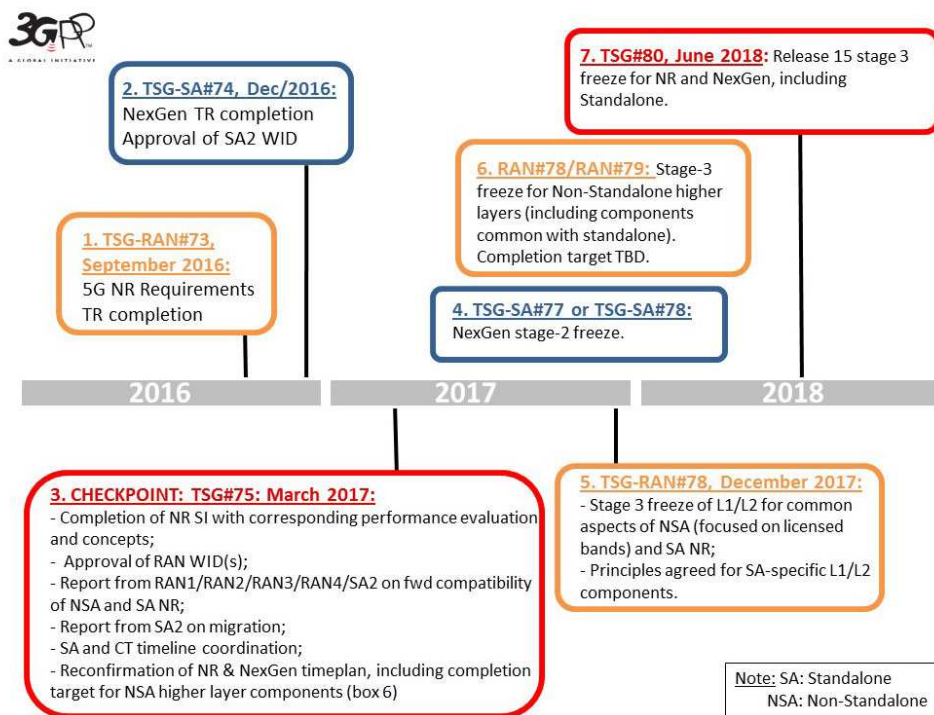


Figura 9. Plan de desarrollo de 5G NR (Fuente: [19])

Como se ha comentado, 5G NR contempla, por primera vez, el uso de los códigos LDPC como técnica de codificación. La última propuesta, y en la que se basa este proyecto, está definida en la versión 15.1.1 publicada por 3GPP, en abril de 2018 [19].

En este caso, la codificación difiere de la comentada en el apartado 7.1). En efecto, en lugar de manejar matrices de paridad \mathbf{H} de dimensiones considerables al tratar con bloques de datos de gran tamaño, se toman como referencia dos matrices de base, denominadas BG1 y BG2, en las que cada elemento representa una matriz de extensión de dimensiones $Z \times Z$. Esto permite almacenar de manera compacta una matriz que sirve de referencia para varios tamaños de código.

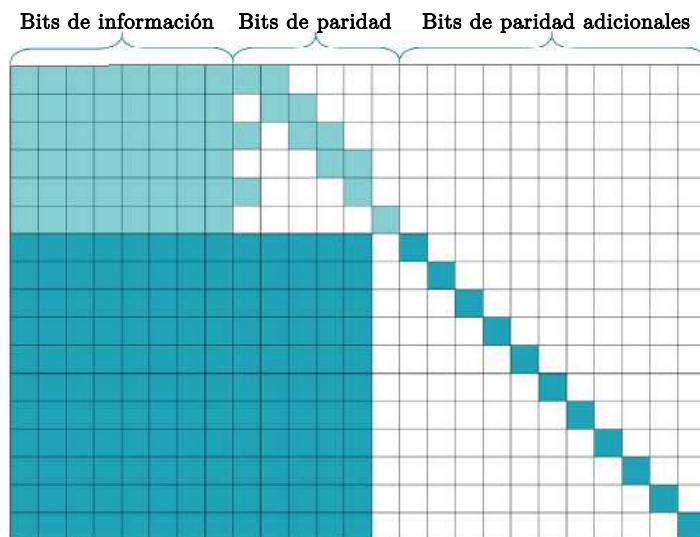


Figura 10. Estructura de la matriz LDPC en 5G NR (Fuente: [13])

La figura [Fig. 10] representa la estructura de la matriz LDPC utilizada en 5G NR. La parte superior izquierda, en fondo claro, define un *code rate* alto (entre $2/3$ y $8/9$). Asimismo, se pueden generar bits adicionales de paridad extendiendo la matriz de base, incluyendo nuevas columnas y filas (marcadas en fondo oscuro, en la parte inferior izquierda). Esto permite la transmisión en los casos en los que se utilicen *code rates* bajos o para generar nuevos bits de paridad en aplicaciones que lo requieran.

Por otra parte, 5G NR utiliza un tipo de codificación distinta para señalización: los códigos polares. Su creación data de 2008 y se trata de los primeros códigos capaces de llegar al límite establecido por Shannon. [7] Son óptimos en bloques de control y señalización dado que proporcionan un gran rendimiento en estos casos, al tratarse de bloques de datos pequeños.

8) METODOLOGÍA

La fase práctica del proyecto consiste en el diseño de una nueva herramienta en MATLAB que permite simular una cadena de transmisión de televisión digital vía satélite basada en DVB-S2 y que integra la nueva codificación LDPC propuesta en 5G NR.

En este apartado, se detalla la metodología seguida para la creación de la herramienta según el orden cronológico en el que se ha ido produciendo el desarrollo de la misma. Se comienza por justificar el código LDPC elegido entre los propuestos en 5G. A continuación, se detalla la codificación y simulación tanto del modelo de referencia DVB-S2 como del código candidato seleccionado. Finalmente, se integra la nueva codificación en la cadena de transmisión definida y se analiza su rendimiento en un caso práctico determinado.

8.1) Familias de códigos LDPC propuestos en 5G

Como se ha comentado en el apartado 7.3), en el estándar 5G NR se han definido nuevas familias de códigos LDPC diferentes a las definidas en los estándares anteriores en los que se utiliza este tipo de codificación (DVB-S2, DVB-T2, ...).

En este caso, teniendo en cuenta que 5G NR ha sido diseñado para abarcar aplicaciones de todo tipo, es necesario que la codificación sea compatible tanto para bloques de datos de tamaño pequeño (de alrededor de 100 bits) como de gran tamaño (de alrededor de 8000 bits). Además, también debe cumplir los requerimientos de 3GPP en cuanto a ancho de banda, de tal forma que la misma codificación se adapte a *code rates* distintos para lograr un ancho de banda de media de 5 Gbps en condiciones óptimas (*code rate* 8/9) como para garantizar 100 Mbps (*code rate* 1/5) en modo *extended coverage*.

3GPP definió dos matrices de base, BG1 y BG2 en las que cada elemento representa una matriz de extensión de dimensiones $Z \times Z$. [19] De esta manera, la construcción de la matriz de paridad \mathbf{H} necesaria para la codificación de los paquetes de datos se calcula sustituyendo cada elemento de la matriz base generadora (BG1 o BG2) por las matrices de extensión $Z \times Z$.

Las características de uso de una matriz u otra depende de la longitud del paquete de datos a codificar: BG1 será idónea para bloques de tamaño considerable, con $500 \leq$

$K_{LDPC} \leq 8448$, y para *code rates* elevados ($1/3 \leq R \leq 8/9$), mientras que BG2 se utilizará para paquetes de tamaño más reducido, con $40 \leq K_{LDPC} \leq 2560$, y para *code rates* bajos ($1/5 \leq R \leq 2/3$).

Tabla 2. Características de los códigos LDPC en 5G NR

	BG1	BG2
Tamaño de bloque	$500 \leq K_{LDPC} \leq 8448$	$40 \leq K_{LDPC} \leq 2560$
<i>Code rate</i>	$1/3 \leq R \leq 8/9$	$(1/5 \leq R \leq 2/3)$.
Dimensiones de la matriz base	46 x 68	42 x 52
Valores de Z	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 22, 24, 26, 28, 30, 32, 36, 40, 44, 48, 52, 56, 60, 64, 72, 80, 88, 96, 104, 112, 120, 128, 144, 160, 176, 192, 208, 224, 240, 256, 288, 320, 352, 384	

Por su parte, el tamaño de los bloques a la entrada del codificador LDPC definido en el estándar DVB-S2 para el modo normal, es de $16.200 \leq K_{LDPC} \leq 58.200$ bits y los *code rates* empleados varían entre $1/4$ y $9/10$ (ver Tabla 1). Teniendo en cuenta ambos parámetros, el código seleccionado entre los propuestos es el definido por la matriz **BG1**. Hay que comentar que se ve necesario adaptar los paquetes de datos de DVB-S2 para soportar el nuevo código. Esta situación se analizará y comentará en el apartado 8.3).

8.2) Codificación del modelo de referencia DVB-S2 en MATLAB

A partir de este apartado, se detalla el proceso de diseño e implementación que se ha llevado a cabo para el desarrollo del programa creado utilizando la herramienta MATLAB.

En primer lugar, se realiza la codificación del modelo de referencia DVB-S2. Para ello es necesario considerar la cadena de transmisión definida en el apartado 7.2.2) completa, es decir incluyendo el transmisor, un canal de comunicación y el receptor. El diagrama de bloques de la cadena de comunicación considerando estos elementos sería la siguiente:

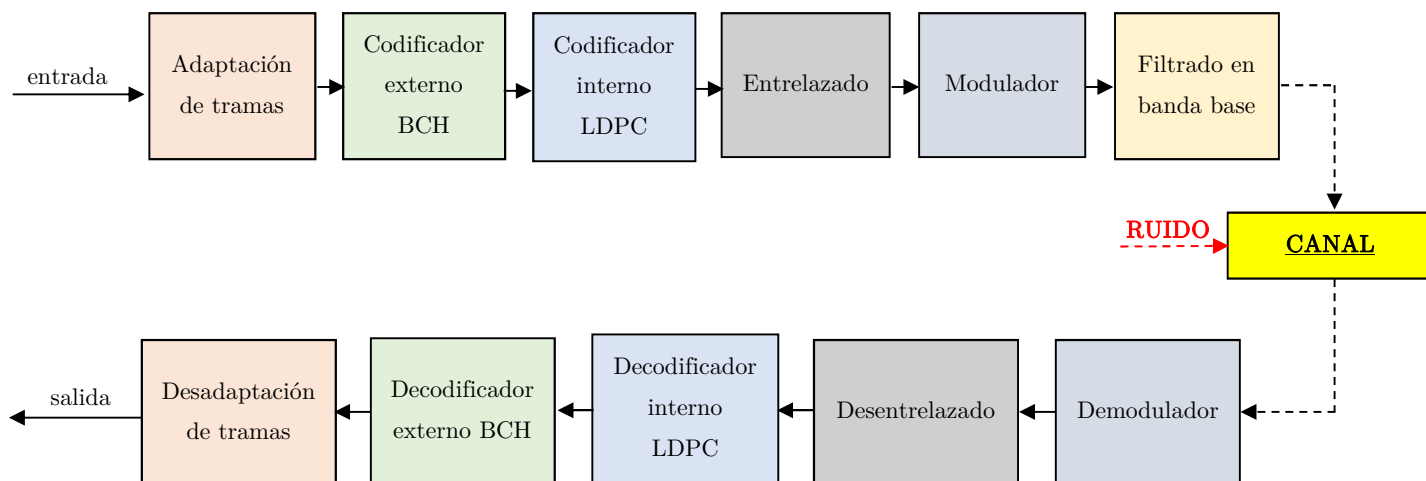


Figura 11. Cadena de transmisión DVB-S2 completa

8.2.1) Cadena simple

Inicialmente, se codifica una cadena de transmisión simple, que se amplía a medida que se verifica su correcto funcionamiento. En este caso, la primera cadena simulada sólo admite un tipo de modulación y *code rate* para un valor fijo de relación energía por símbolo respecto a la densidad de potencia de ruido (E_s/N_0).

La primera parte de este desarrollo consiste en definir todos los bloques de la cadena de transmisión [Figura 11]. Para ello, se utilizan las librerías proporcionadas por MATLAB y, en concreto, de la *Communications System Toolbox*.

- Fuente binaria:

La fuente utilizada en todas las simulaciones se basa en el modo de funcionamiento CCM de DVB-S2, en el que la modulación y la codificación se mantienen constantes a lo largo de las mismas. Asimismo, el tamaño de los paquetes generados se corresponde con el de los paquetes MPEG-TS (188 bytes) utilizados para la transmisión de televisión digital en sistemas DTH.

Para simplificar la generación de bits, se ha optado por utilizar la función *randi* que genera arrays pseudoaleatorios de “0” y “1” de tamaño 1504 bits cada uno (correspondiente al tamaño de un paquete MPEG-TS).

- Codificador BCH:

El codificador BCH utilizado necesita de los tamaños de datos de entrada K_{BCH} , salida N_{BCH} y del polinomio primitivo para su funcionamiento. Para ello, se ha definido una función *getbchparameters* que devuelve dicha información para cada tipo de *code rate* tal y como se detalla en la siguiente tabla:

Tabla 3. Parámetros del codificador BCH

<i>Code rate</i>	K_{BCH}	N_{BCH}	t
1/4	16.008	16.200	12
1/3	21.408	21.600	12
2/5	25.728	25.920	12
1/2	32.208	32.400	12
3/5	38.688	38.880	12
2/3	43.040	43.200	10
3/4	48.408	48.600	12
4/5	51.648	51.840	12
5/6	53.840	54.000	10
8/9	57.472	57.600	8
9/10	58.192	58.320	8

- Codificador LDPC:

Para la obtención de la matriz de paridad \mathbf{H} asociada a la codificación LDPC bastará con utilizar la función *dvbs2ldpc* ya definida en MATLAB y proporcionar el *code rate* asociado.

- Entrelazador:

En el caso de utilizar las modulaciones 8PSK, 16APSK y 32APSK es necesario entrelazar la salida del codificador LDPC de manera que los datos se escriben en diferentes registros por columnas y se leen por filas. El número de columnas y filas viene detallado en la siguiente tabla:

Tabla 4. Parámetros del entrelazador

Modulación	Número de filas	Número de columnas
8PSK	21.600	3
16APSK	16.200	4
32APSK	12.960	5

- Modulador

Para modular la señal siguiendo los esquemas de modulación definidos en la figura [Figura 8] es preciso determinar la constelación de cada una de las modulaciones utilizadas.

En el caso de modulaciones simples como QPSK o 8PSK, la obtención de la constelación y del mapeado de cada símbolo se obtendrá de manera clásica, esto es, indicando los valores teóricos de la posición de cada símbolo dentro de la modulación.

En el caso de modulaciones más complejas como 16APSK o 32APSK, la obtención de la constelación y del mapeado se hará llamando a la función *dvbsapskmod* indicando la modulación y *code rate* utilizados. Hay que señalar que la función necesita del *code rate* en formato texto, por lo que es necesario adaptar el valor del mismo al requerido.

- Canal:

El canal definido para las simulaciones es de tipo AWGN, es decir, en el que se ha añadido un ruido aditivo blanco de tipo gaussiano. La varianza del ruido empleada en este canal viene dada por la relación:

$$\sigma^2 = \frac{1}{10^{\frac{E_s}{N_0} (dB)}}$$

Donde E_s/N_0 , es fijado por el usuario al comienzo de la simulación.

- Receptor:

En cuanto al receptor, los bloques de decodificación y demodulación son muy parecidos a los detallados anteriormente para el emisor, por lo que se ha decidido omitir su explicación de cara a evitar saturar la memoria de este proyecto.

El detalle del código utilizado para la definición de los bloques que forman la cadena de DVB-S2 está disponible en el anexo 1.2).

8.2.2) Primeras simulaciones

Para poder realizar las primeras simulaciones, se han conectado entre sí los bloques definidos en el apartado anterior mediante el comando *step*. De esta manera, la salida de la fuente binaria está conectada a la entrada del codificador BCH. A su vez, la salida del codificador BCH está conectada a la entrada del codificador LDPC. Esta acción se repite hasta completar todos los bloques que forman la arquitectura de DVB-S2.

En este caso, se ha considerado el caso más sencillo posible. Es decir, cuando únicamente se está transmitiendo un único paquete MPEG-TS para un valor fijo de *code rate* y E_s/N_0 .

Tabla 5. Parámetros de las primeras simulaciones

<i>Code rate</i>	1/2, 2/3, 3/4, 5/6
E_s/N_0	-1 dB
Número de tramas	1
Modulación	QPSK, 8PSK, 16APSK, 32APSK

Tras realizar la primera simulación, se ha comprobado en el entorno de trabajo de MATLAB que tanto el formato como el tamaño de las tramas eran correctos y se correspondían a los indicados en las tablas 3 y 4.















 numPaquetesPorTramaBB	32
 numTramas	1
 offsetFase	0.7854
 Qy	[1;-1;-1]
 salidaCanal	32400x1 complex double
 salidaCodificadorBCH	48600x1 logical
 salidaCodificadorLDPC	64800x1 logical
 salidaDecodificadorBCH	48408x1 logical
 salidaDecodificadorLDPC	48600x1 logical
 salidaDemodulador	64800x1 double
 salidaDesentrelazador	64800x1 double
 salidaEntrelazador	64800x1 logical
 salidaModulador	32400x1 complex double
 simbolosRx	1x32400 complex double
 simbolosTx	1x32400 complex double
 tipoModulacion	'QPSK'
 tramaBBRx	48408x1 logical
 tramaBBTx	48408x1 logical

Figura 12. Detalle del entorno de trabajo de MATLAB para la primera simulación

Asimismo, se ha calculado el BER (*Bit Error Rate*), para cada una de las cuatro modulaciones utilizadas, según la expresión:

$$BER = \frac{\text{núm. bits erróneos}}{\text{núm. bits totales}}$$

Los resultados de estas primeras simulaciones están detallados en la siguiente tabla:

Tabla 6. Resultados de las primeras simulaciones

Modulación			QPSK	8PSK	16APSK	32APSK
Code rate	1/2	BER	0.1758	0.2765	-	-
	2/3		0.1838	0.2826	0.3253	-
	3/4		0.1868	0.2826	0.3350	0.3395
	5/6		0.1879	0.3167	0.3495	0.3479

Puede comprobarse que los resultados obtenidos coinciden con el comportamiento esperado teóricamente. En efecto, a medida que se aumenta el orden del tipo de modulación utilizado, disminuye el órgano de decisión entre símbolos, por lo que el decodificador tendrá mayor dificultad para corregir y decodificar la trama recibida. Ocurrirá el mismo fenómeno al aumentar el valor del *code rate*, ya que la redundancia introducida en el proceso de codificación será menor, provocando errores en las tramas recibidas. Asimismo, puede que sorprenda la obtención de niveles de BER tan elevados. En este caso, está totalmente justificado dado que se ha escogido un valor de E_s/N_0 muy bajo (de -1 dB) para las primeras simulaciones.

8.2.3) Mejora del código

Una vez comprobado el correcto funcionamiento de la cadena simple de DVB-S2 codificada, resulta primordial introducir ciertos elementos de mejora que permitan comparar los resultados obtenidos y realizar representaciones gráficas.

Una de las primeras mejoras introducidas en el código ha sido incluir la posibilidad de transmitir varias tramas, para poder analizar los resultados reales de una transmisión DTH. Su implementación se fundamenta en la introducción de un bucle *for* que repita la transmisión para cada una de las tramas. Del mismo modo, se ha añadido la posibilidad de obtener resultados para varios valores de E_s/N_0 dados, así como para distintos valores de *code rate*.

Asimismo, se pueden realizar representaciones gráficas de la constelación transmitida respecto a la recibida, además de la relación entre los distintos valores de BER y E_s/N_0 .

Por ejemplo, si se considera ahora la siguiente simulación:

Tabla 7. Parámetros de la simulación con código mejorado

<i>Code rate</i>	1/2, 2/3, 4/5, 9/10
E_s/N_0	-1 a 6 dB (en pasos de 0.5 dB)
Número de tramas	20
Modulación	QPSK

Los resultados obtenidos para esta segunda simulación realizada con el código mejorado se trasladan a la siguiente tabla:

Tabla 8. Resultados de la segunda simulación

E_s/N_0		-1	-0.5	0	0.5	1	1.5	2	2.5	
<i>Code rate</i>	1/2	BER	0,1732	0,1487	0,1277	0,0945	0	0	0	
	2/3		0,187	0,1689	0,157	0,1414	0,1237	0,1031	0,0841	0,0515
	4/5		0,187	0,1734	0,159	0,1468	0,1314	0,1179	0,1008	0,0907
	9/10		0,1884	0,1729	0,158	0,1447	0,1324	0,1177	0,1034	0,0907

E_s/N_0		3	3.5	4	4.5	5	5.5	6
<i>Code rate</i>	1/2	BER	0	0	0	0	0	0
	2/3		0	0	0	0	0	0
	4/5		0,0759	0,0617	0,0439	0	0	0
	9/10		0,0791	0,066	0,0537	0,0453	0,0365	0,0257

Además, la relación entre el BER y E_s/N_0 puede observarse de una manera más clara en la siguiente gráfica:

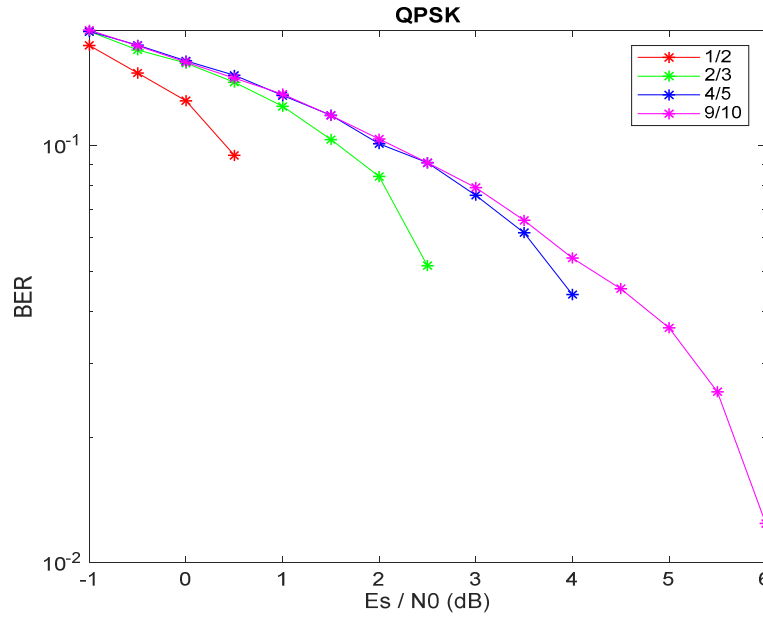


Figura 13. Relación BER vs. E_s/N_0

La gráfica obtenida coincide con el modelo teórico esperado. En efecto, a medida que el valor del *code rate* es más pequeño, la codificación es más robusta ante los errores producidos por el ruido radioeléctrico, de manera que el valor del BER se reduce también.

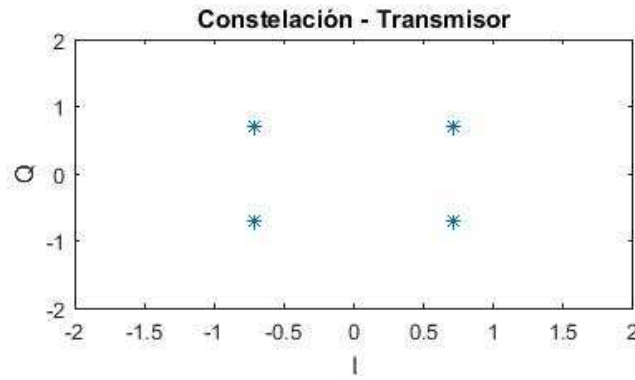


Figura 14. Constelación QPSK en el transmisor

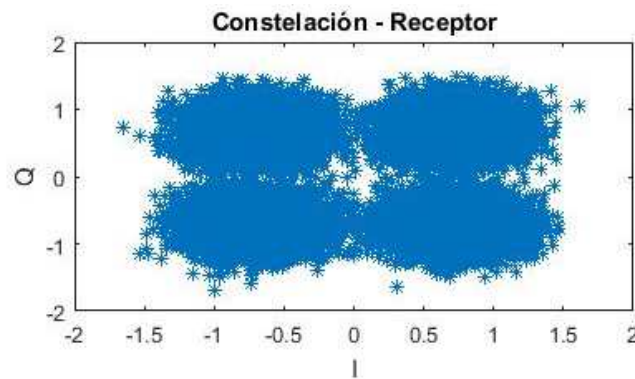


Figura 15. Constelación QPSK en el receptor

Si ahora se representa la constelación del transmisor [Fig. 14] y la correspondiente al receptor [Fig. 15] puede observarse el efecto que produce el ruido del canal en la transmisión. En efecto, los símbolos se dispersan de su posición original y, en algunas ocasiones, traspasan el órgano de decisión del receptor. A pesar de ello, la codificación utilizada permite recuperar y corregir estos errores en su mayoría.

El detalle del código mejorado utilizado para la simulación de la cadena de DVB-S2 está disponible en el anexo 1.4). Se incluye además los resultados obtenidos para las simulaciones realizadas para las demás modulaciones definidas en DVB-S2 [Tabla 1] en el anexo 2.

8.3) Codificación del código candidato seleccionado en MATLAB

Como se ha comentado en el apartado 8.1), el código candidato seleccionado entre las nuevas familias de códigos definidos por 3GPP en el estándar 5G NR, es el que se genera a partir de la matriz de base BG1. No obstante, estos códigos presentan una limitación técnica al establecer como tamaño máximo de bloque $K_{LDPC} = 8.448$ bits.

Si se tiene en consideración los parámetros de la codificación en BCH utilizada [Tabla 3] en la arquitectura de DVB-S2 [Fig. 11], el número de bits a la salida del codificador BCH es el mismo que el número de bits a la entrada del codificador LDPC. De tal forma que, $N_{BCH} = K_{LDPC}$. Es decir, el rango de valores K_{LDPC} que admite DVB-S2 comienza en 16.200 bits y puede llegar hasta los 58.320 bits, dependiendo del *code rate* utilizado. Por tanto, será necesario dividir las tramas utilizadas en un factor de, como mínimo 2 y, como máximo 7, para poder hacer uso de la codificación definida para 5G.

Asimismo, hay que comentar que es necesario realizar una adaptación del *code rate* para que sea válido con el código candidato. En efecto, el *code rate* asociado a la matriz BG1 es de 1/3, por lo que es necesario realizar dicha adaptación para su compatibilidad con otros valores y, en concreto, con los utilizados en DVB-S2.

En el proceso de codificación, hay que tener en cuenta que, para el cálculo de las matrices de extensión, se consideran dos escenarios:

- Si el elemento en BG1 o BG2, es un “0”, se reemplaza por una matriz nula de dimensiones $Z \times Z$.
- Si el elemento en BG1 o BG2, es un “1”, se reemplaza por una matriz de dimensiones $Z \times Z$ realizando permutaciones cíclicas circulares según las tablas definidas por 3GPP [19].

Por último, es necesario tener en consideración que el estándar 5G NR especifica la inclusión de un código de redundancia cíclica (CRC) de 24 bits anexo a las tramas transmitidas. Estos nuevos bits de paridad deben eliminarse en recepción para recuperar la trama original, lo que conlleva la inserción de una etapa nueva en el proceso de codificación.

En resumen, la codificación candidata puede esquematizarse según el siguiente diagrama de bloques:

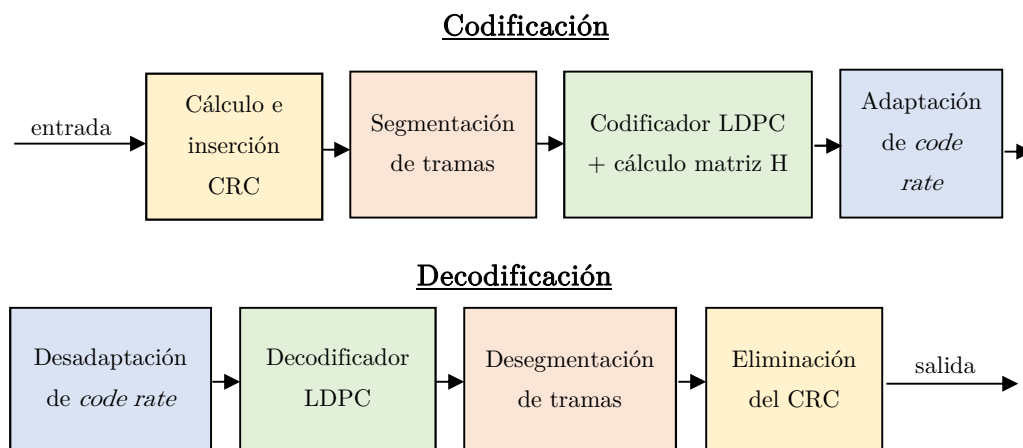


Figura 16. Proceso de codificación y decodificación del código candidato

Para implementar la nueva codificación en MATLAB, se utiliza como base las librerías de 5G NR definidas en la *Toolbox* de LTE. De esta manera, será más sencillo realizar la codificación y adaptarla, más adelante, al sistema de referencia DVB-S2.

Esta implementación se realiza, en primer lugar, de manera autónoma. Es decir, se generarán una cantidad de bits compatibles con los tamaños de K_{LDPC} a los que se les aplicará la nueva codificación LDPC y, en ausencia de canal, se decodificará directamente. De esta manera es posible evaluar los errores cometidos en el proceso y corregirlos antes de incorporarlos a la cadena de comunicación ya creada.

La generación de los bits de los paquetes de datos sigue el mismo mecanismo que en el apartado 8.2.1). Por su parte, el cálculo y la inserción del CRC se realiza utilizando la función *h5gCRCEncode*. Para segmentar las tramas en el tamaño aceptado por 5G NR, se llama a la función *h5gCodeBlockSegment* indicando la matriz base utilizada (BG1). Cuando las tramas no son múltiplo de 8448, se introducen varios “-1” de relleno, de manera que el codificador LDPC pueda ignorarlas.

Las tramas segmentadas se codifican una a una llamando a la función *h5gLDPCEncode*. El último paso consistiría en adaptar las tramas al *code rate* empleado en la simulación. Para ello, en esta ocasión, se indica como parámetros a la función *h5gRateMatchLDPC*, el tamaño de bits N_{LDPC} requerido a la salida.

El proceso de decodificación es similar al proceso de codificación aquí detallado, aunque en este caso, se realiza en orden inverso, como se puede observar en la figura [Fig. 16].

El detalle del código de la nueva codificación LDPC implementada está disponible en el anexo 1.5).

8.4) Integración del nuevo módulo LDPC en el modelo de referencia DVB-S2

La última fase del desarrollo del programa creado, consiste en la integración del nuevo módulo LDPC implementado en el modelo de referencia DVB-S2. Para ello, resulta conveniente volver a revisar la arquitectura de este estándar y realizar un nuevo esquema que permita visualizar el diagrama de bloques del sistema modificado.

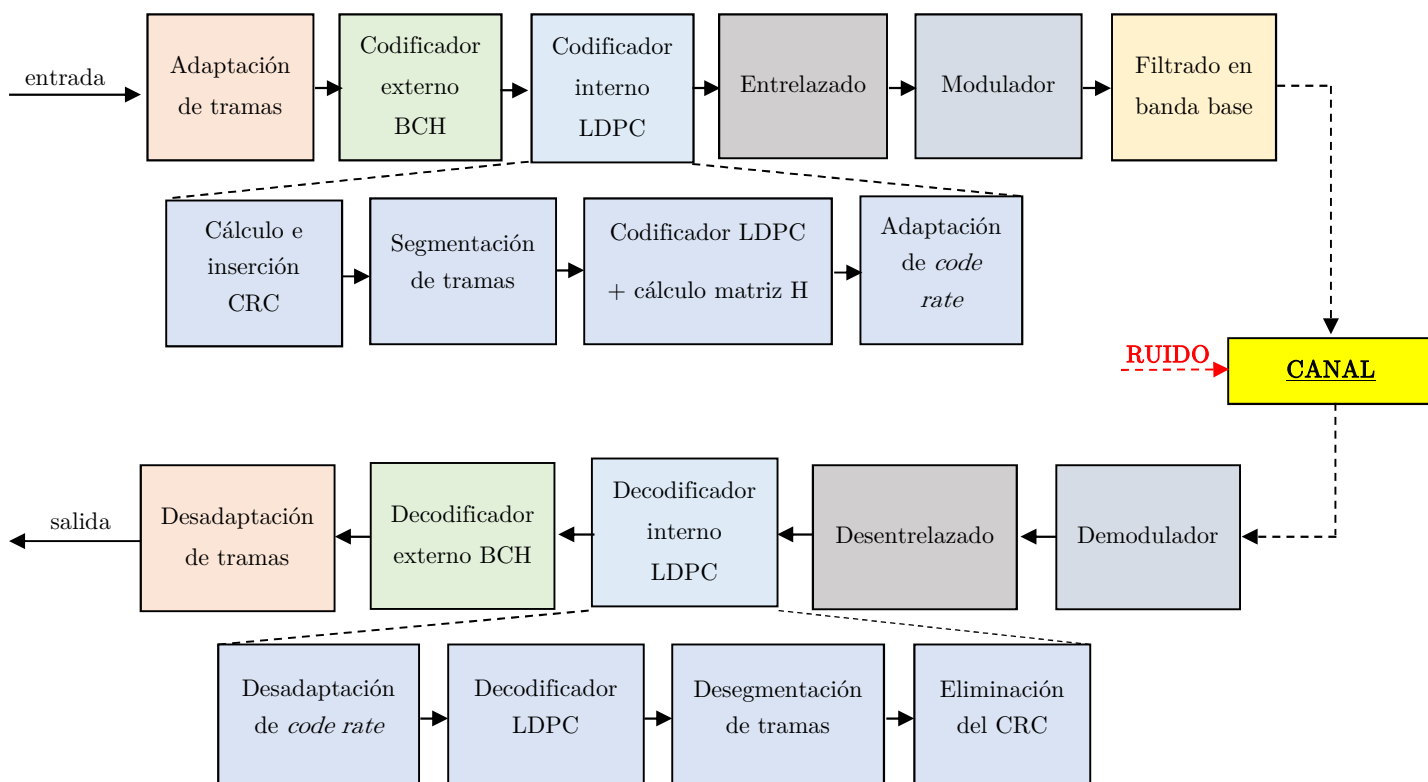


Figura 17. Arquitectura del nuevo sistema creado.

El nuevo módulo de codificación LDPC está compuesto por cuatro etapas que requieren de ciertas adaptaciones para su correcto funcionamiento dentro de DVB-S2. En primer lugar, y como se ha comentado en el apartado anterior 8.4), se produce una inserción de 24 nuevos bits de paridad al implementar el nuevo módulo. Es necesario verificar el resto de elementos de la cadena de transmisión al tener ahora tramas de tamaño $N_{LDPC} + 24$ bits a la salida del nuevo codificador.

En efecto, el entrelazador definido para DVB-S2 ya no resulta válido, dado que N_{LDPC} toma ahora un valor de 64824 bits, en lugar de 64800 bits. Los parámetros del nuevo entrelazador adaptado a esta circunstancia se detallan en la siguiente tabla:

Tabla 9. Parámetros del nuevo entrelazador

Modulación	Número de filas	Número de columnas
8PSK	21.608	3
16APSK	16.206	4
32APSK	12.964	5

Por otra parte, se ha comprobado que las funciones que realizan la adaptación del *code rate* (*h5gRateMatchLDPC* y *h5gRateRecoverLDPC*) incluyen el entrelazador utilizado en 5G NR. Si bien este último es útil para simular enlaces basados en esa tecnología, no resulta conveniente incluirlo en esta nueva cadena dado que se introducirían dos procesos de entrelazado en cascada. Por ese motivo, y para no causar nuevas fuentes de error que puedan mermar la calidad de transmisión, se ha decidido crear dos nuevas funciones (*h5gRateMatchLDPC_2* y *h5gRateRecoverLDPC_2*) idénticas a las originales pero eliminando el proceso de entrelazado de 5G NR.

Una vez comprobado el correcto funcionamiento de la cadena de transmisión que implementa el nuevo módulo de codificación LDPC, se ha procedido a definir un caso práctico que permite evaluar la solución planteada.

8.5) Caso práctico: simulación y resultados obtenidos

Para evaluar la idoneidad de la nueva codificación LDPC implementada, resulta primordial definir un caso práctico que permita evaluar su rendimiento y a su vez, poder compararlo en dos entornos diferenciados. Para ello, se toma exclusivamente por una parte el estándar DVB-S2 original y, por otra, el nuevo modelo creado.

Antes de fijar un caso práctico de simulación, se ha comprobado el comportamiento de la herramienta creada para diferentes tipos de modulaciones y *code rates*. Se ha observado que los tiempos de simulación y la precisión de los resultados que se obtenían eran óptimos en el caso de utilizar una modulación QPSK. De esta manera, se ha considerado establecer los siguientes casos de estudio:

Tabla 10. Casos de estudio

	Caso 1	Caso 2
Modulación	QPSK	
<i>Code rate</i>	1/2	1/3
Número de tramas	100	

Para definir el rango de valores que toma E_s/N_0 , se ha decidido simular en primer lugar el envío de una sola trama para un intervalo medianamente amplio (entre -4 y 7 dB). Con ello se logra acotar la zona de convergencia y definir una matriz de puntos no homogénea que incluya diferentes pasos de dB. De esta forma, se consiguen resoluciones altas en la zona que nos interesa (zona de convergencia) y resoluciones más bajas en las demás zonas. Esta decisión permite establecer un compromiso entre el número de tramas transmitidas y la resolución de las gráficas obtenidas al igual que permite limitar en cierto modo los tiempos de simulación que pueden llegar a ser de varias horas.

Las siguientes tablas y gráficas muestran los valores E_s/N_0 y los resultados obtenidos en cada uno de los casos de estudio.

Tabla 11. Valores de E_s/N_0 en QPSK 1/2

E_s/N_0	de -1 a 0.4 dB	de 0.4 a 1 dB
Pasos	0.1 dB	0.03 dB

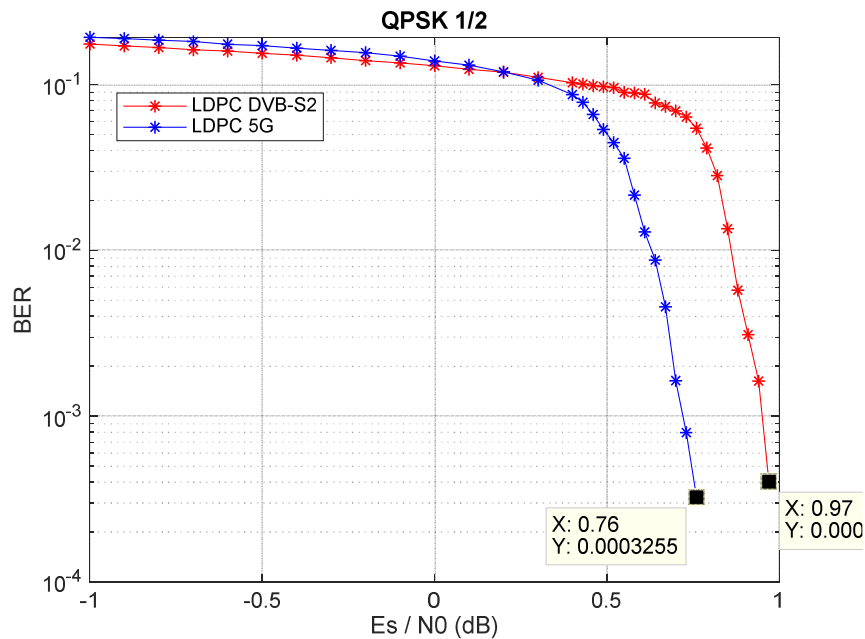


Figura 18. Relación BER vs. E_s/N_0 para QPSK 1/2 en los dos entornos

Tabla 12. Valores de E_s/N_0 en QPSK 3/4

E_s/N_0	de -4 a -2 dB	de -2 a -1.5 dB
Pasos	0.1 dB	0.03 dB

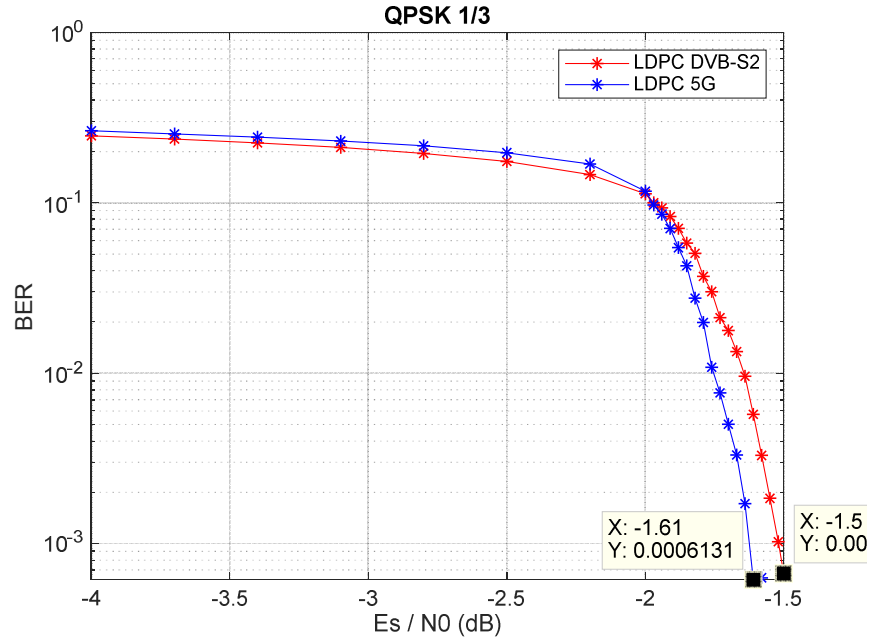


Figura 19. Relación BER vs. E_s/N_0 para QPSK 1/3 en los dos entornos

Con los resultados obtenidos de las simulaciones realizadas en los dos casos prácticos pueden establecerse las siguientes conclusiones:

En primer lugar, se puede observar que en el caso de DVB-S2, los valores de E_s/N_0 obtenidos cuando no hay errores (cuando el BER tiende a 0) coinciden prácticamente con los valores teóricos proporcionados en el anexo 3. Esto nos permite comprobar el correcto funcionamiento de la herramienta desarrollada para ese estándar.

Por otra parte, se observa que se obtiene una ganancia con el uso de los nuevos códigos LDPC definidos en el estándar 5G NR respecto a los existentes en DVB-S2. En el primer caso (QPSK 1/2), puede apreciarse que dicha ganancia es de aproximadamente 0.2 dB, coincidiendo con el valor teórico esperado. En el segundo caso (QPSK 1/3), la ganancia obtenida (0.11 dB) es algo inferior al comportamiento esperado teóricamente pero refleja de igual manera la mejora introducida por la implementación de la nueva codificación. En efecto, hay que comentar que, si bien es cierto que los nuevos códigos LPDC definidos en 5G introducen mejoras en el rendimiento, éste se ve disminuido parcialmente por la necesidad de segmentar los paquetes de datos para adaptarlos al estándar DVB-S2. No obstante, se consigue un buen rendimiento del nuevo código, mejorando a la codificación LDPC definida en DVB-S2.

9) PLANIFICACIÓN

La planificación constituye uno de los pasos previos más importantes en la realización de un proyecto. Permite establecer su rumbo a través de unos objetivos claros e incluso identificar las necesidades del proyecto en cada momento.

9.1) Equipo de trabajo

En este apartado, se especifican las personas que trabajan en este proyecto y su perfil.

Tabla 13. Equipo de trabajo

Nombre	Responsabilidad	Perfil
Pablo Angueira	Director del proyecto	Ingeniero senior
Juan Luis Ordiales	Codirector del proyecto	Ingeniero senior
Jon Montalbán	Ayudante del proyecto	Ingeniero senior
Jon Maseda	Realizador del proyecto	Ingeniero junior

9.2) Fases del trabajo

Tabla 14. PT0. Coordinación, gestión y supervisión del proyecto.

PT0. Coordinación, gestión y supervisión del proyecto
El director y codirector supervisan el trabajo realizado. Comienzo: 15/02/2018 Fin: 29/06/2018 Recursos humanos: Dos ingenieros senior (40 horas) Recursos técnicos: Un ordenador con la suite Microsoft Office (10 horas) y el software MATLAB instalado (5 horas)

Tabla 15. PT1. Planificación del proyecto.

PT1. Planificación del proyecto
El director y codirector describen las pautas a seguir para la realización del proyecto. PT1.1. Definición de los objetivos y alcance. Definir los objetivos del proyecto dejando claro cuál será su finalidad.

<p>PT1.2. Definición de las pautas del proyecto.</p> <p>Definir el plan de trabajo para lograr los objetivos marcados.</p> <p>Comienzo: 15/02/2018</p> <p>Fin: 09/03/2018</p> <p>Recursos humanos: Dos ingenieros senior (10 horas) y un ingeniero junior (10 horas)</p> <p>Recursos técnicos: Un ordenador con la suite Microsoft Office (10 horas)</p>

Tabla 16. PT2. Estudio del estado del arte

<p>PT2. Estudio del estado del arte</p> <p>Estudio y análisis de los conocimientos teóricos requeridos para la realización del proyecto.</p> <p>PT2.1. Códigos LDPC.</p> <p>Realización de un estudio del estado del arte de los códigos LDPC en la actualidad.</p> <p>PT2.2. Estándar DVB-S2.</p> <p>Estudio del estándar de transmisión DVB-S2 y comparativa de eficiencia respecto a otras tecnologías.</p> <p>PT2.3. Estándar 5G NR.</p> <p>Estudio del nuevo estándar definido para telefonía móvil 5G y, en particular, de las familias de códigos LDPC propuestos en 5G</p> <p>Comienzo: 12/03/2018</p> <p>Fin: 16/04/2018</p> <p>Recursos humanos: Un ingeniero junior (60 horas)</p> <p>Recursos técnicos: Un ordenador con la suite Microsoft Office (45 horas)</p>

Tabla 17. PT3. Profundización en el programa MATLAB

<p>PT3. Profundización en el programa MATLAB</p> <p>Profundización en el conocimiento del programa para su uso en el desarrollo de la solución al problema planteado.</p> <p>Comienzo: 17/04/2018</p> <p>Fin: 23/04/2018</p> <p>Recursos humanos: Un ingeniero junior (10 horas)</p> <p>Recursos técnicos: Un ordenador con el software MATLAB (10 horas)</p>
--

Tabla 18. PT4. Desarrollo de la aplicación

PT4. Desarrollo de la aplicación
<p>Este paquete de trabajo constituye la mayor carga del proyecto. Consiste en el desarrollo de la herramienta que permite dar solución al problema planteado.</p> <p>PT4.1. Puesta a punto del modelo de referencia DVB-S2 en MATLAB. Diseño e implementación del estándar DVB-S2 en MATLAB.</p> <p>PT4.2. Selección e implementación de un módulo de codificación LDPC candidato. Análisis de la familia de códigos LDPC propuestos en el nuevo estándar 5G NR e implementación del código escogido en MATLAB.</p> <p>PT4.3. Integración del módulo de codificación LDPC en el modelo de referencia DVB-S2. Desarrollo del nuevo modelo que alberga la nueva codificación desarrollada y comprobación de su correcto funcionamiento.</p> <p>Comienzo: 24/04/2018 Fin: 08/06/2018 Recursos humanos: Dos ingenieros senior (30 horas) y un ingeniero junior (150 horas) Recursos técnicos: Un ordenador con el software MATLAB (115 horas)</p>

Tabla 19. PT5. Simulación y obtención de curvas de rendimiento.

PT5. Simulación y obtención de curvas de rendimiento.
<p>Realización de las diferentes simulaciones para la obtención de resultados del caso práctico y, en particular, de las curvas de rendimiento.</p> <p>Comienzo: 11/06/2018 Fin: 25/06/2018 Recursos humanos: Dos ingenieros senior (10 horas) y un ingeniero junior (20 horas) Recursos técnicos: Un ordenador con el software MATLAB (15 horas)</p>

Tabla 20. PT6. Informes finales

PT6. Informes finales.
<p>Diseño y formato del documento de la memoria final del proyecto. Redacción y corrección</p> <p>Comienzo: 05/06/2018 Fin: 29/06/2018 Recursos humanos: Un ingeniero junior (50 horas) Recursos técnicos: Un ordenador con la suite Microsoft Office (50 horas)</p>

Tabla 21. Entregables

Entregables		
E1	Documento con la planificación del trabajo	09/03/2018
E2	Bibliografía referente al estado del arte	16/04/2018
E3	Código implementación de la aplicación desarrollada	08/06/2018
E4	Memoria final	29/06/2018

Tabla 22. Hitos

Hitos		
H1	Planificación finalizada	09/03/2018
H2	Estado del arte completado	16/04/2018
H3	Simulaciones finalizadas	25/06/2018
H4	Finalización del proyecto	29/06/2018

9.3) Diagrama de Gantt

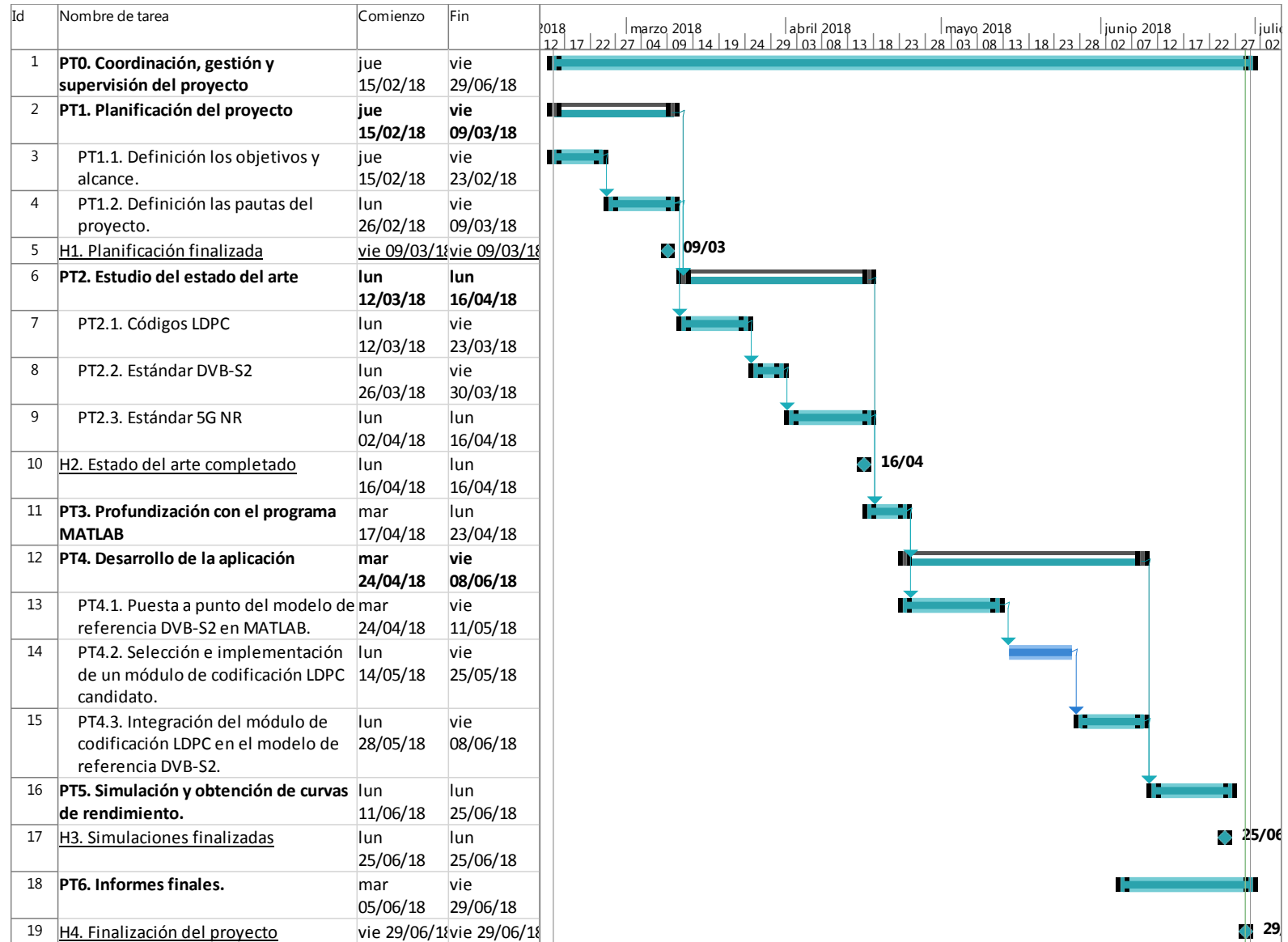


Figura 20. Diagrama de Gantt

10) PRESUPUESTO Y COSTES

En este apartado, se incluye un análisis de los costes y del presupuesto del proyecto. Estos gastos pueden dividirse en dos grupos: horas internas y amortizaciones.

10.1) Horas internas

En este apartado se incluyen los costes referentes a los participantes pertenecientes al equipo de trabajo del proyecto. Los costes indirectos (luz, agua, materiales de oficina...) ya se han incluido juntos con los costes directos de personal.

Tabla 23. Horas internas.

Puesto	Coste (€/h)	Número de horas (h)	Coste total (€)
Director del proyecto	80	50	4.000,00
Codirector del proyecto	70	20	1.400,00
Ayudante del proyecto	50	20	1.000,00
Realizador del proyecto	30	300	9.000,00
TOTAL			15.400,00

10.2) Amortizaciones

Para este apartado, se han tenido en cuenta el coste de uso del ordenador utilizado, así como de las licencias de Microsoft Office, de Matlab R2018a y de la *Toolbox* LTE utilizada.

Se ha tenido en cuenta que la vida útil del ordenador utilizado es de, aproximadamente, 36 meses. Suponiendo que se va a utilizar alrededor de 300 días al año durante 8 horas al día, su utilización sería de 7.200 horas.

Por otra parte, las licencias de software utilizado tienen una vigencia anual. Se ha considerado, en este caso, una utilización anual de 1.500 horas para cada programa.

Tabla 24. Amortizaciones.

Recurso	Coste inicial (€)	Vida útil (h)	Tiempo de uso (h)	Coste total (€)
Ordenador	1.000	7.200	300	41,67
Licencia Matlab	500	1.500	145	48,33
<i>Toolbox</i> LTE	300	1.500	145	29,00
Licencia Microsoft Office	100	1.500	105	7,00
TOTAL				126,00

10.3) Gastos totales

Por último, se presenta el presupuesto completo, en el que se tienen en cuenta los gastos mencionados en los apartados anteriores.

Tabla 25. Presupuesto completo

Concepto	Subtotal (€)
Horas internas	15.400,00
Amortizaciones	126,00
TOTAL	15.526,00

El coste final del proyecto *Análisis e implementación de una nueva codificación LDPC en sistemas de comunicaciones por satélite* asciende a un total de quince mil quinientos veintiséis euros.

En Bilbao, a 29 de junio de 2018.

Fdo. Jon Maseda García

11) ANÁLISIS DE RIESGOS

El objetivo de este análisis de riesgos consiste en identificar y evaluar los eventos que pueden impactar de forma negativa en el desarrollo del proyecto para poder anticipar soluciones. Permite estudiar las causas y las consecuencias directas e indirectas de circunstancias que pueden reducir el éxito y futuro del proyecto.

11.1) Identificación y evaluación de los riesgos

En este apartado, se identifican los riesgos que podrían afectar negativamente en el éxito del proyecto. Se clasifican en función del impacto que puedan tener, así como de la posibilidad de que suceda dicho riesgo. Se utilizan respectivamente las variables I y P en una escala de 1 (bajo) a 3 (alto).

- **Estándar de transmisión 5G NR relativamente nuevo:**

El estándar de transmisión 5G NR es relativamente nuevo en el mercado. La codificación LDPC que se analiza en este proyecto data de abril de 2018. La falta de modelos de simulación y de codificación de este estándar puede suponer un problema importante teniendo en cuenta las limitaciones existentes en cuanto a comprobación de la eficacia de la codificación implementada en MATLAB. La probabilidad de que este hecho ocurra es baja ($P = 1$), pero su impacto podría ser moderado, pudiendo afectar a la planificación del proyecto ($I = 2$).

- **Configuración errónea del código realizado:**

Si se configurasen los parámetros de la herramienta creada de forma errónea, provocaría retrasos en el desarrollo del proyecto. Por una parte, debido al tiempo que sería necesario para realizar una nueva simulación y por otra, debido al tiempo que se requiere para identificar el error y lanzar de nuevo la simulación. La probabilidad de que esto ocurra es alta ($P = 3$) dado que la herramienta creada en su versión inicial puede contener errores, pero su impacto es leve ($I = 1$) ya que se ha previsto el tiempo suficiente para la corrección de los errores de la misma.

- Pérdida de datos / Avería de equipos:

Este puede considerarse como el riesgo con mayor impacto debido a que la pérdida de datos de los códigos y funciones creados implicaría el tener que comenzar nuevamente el proyecto y, posiblemente, conllevaría a su fracaso por falta de tiempo. El impacto podría calificarse como alto ($I = 3$) y la probabilidad de que ocurra es moderada ($P = 2$).

En la siguiente figura, se representan los riesgos descritos en función de su probabilidad de aparición y su impacto en el desarrollo del proyecto.

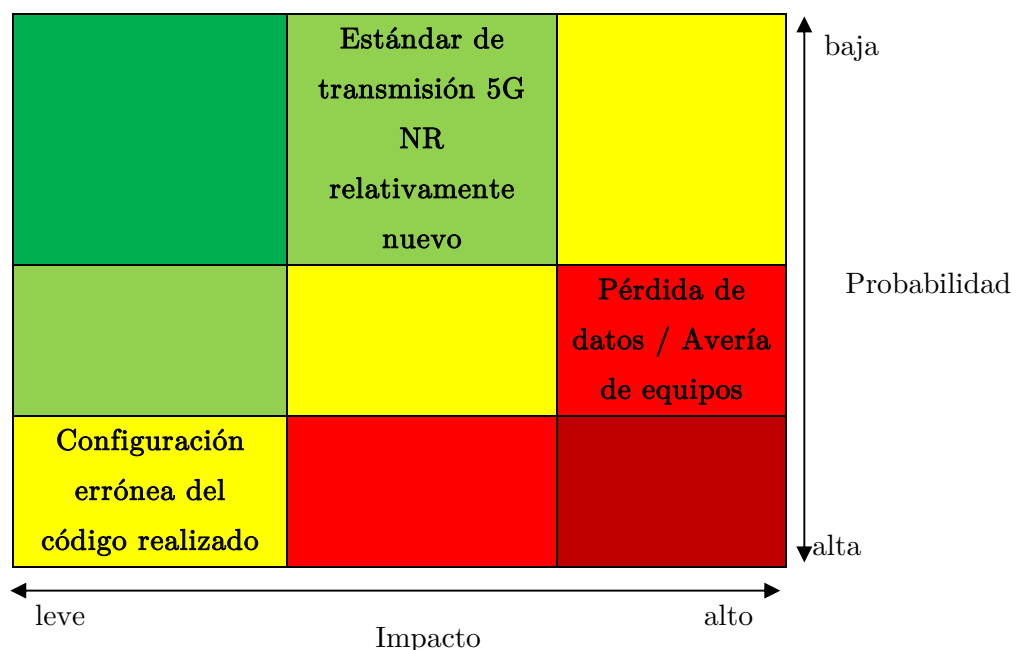


Figura 21. Matriz de riesgos

11.2) Planes de contingencia

Los planes de contingencia aquí descritos se centran en prevenir el riesgo, limitar la probabilidad de aparición, o directamente establecer un plan alternativo en el caso de que ocurra.

- Estándar de transmisión 5G NR relativamente nuevo:

En este caso, el único plan de contingencia posible es la constante actualización de la bibliografía que se consulta desde la realización del proyecto. Al tratarse de una tecnología reciente, la comunidad científica irá publicando artículos constantemente

durante el desarrollo del proyecto, por lo que resulta importante consultar regularmente las fuentes de información que analizan, cuestionan y proponen modelos para el nuevo estándar de transmisión.

- **Configuración errónea del código realizado:**

Un plan de contingencia posible radica en la previsión de disponer de un mayor tiempo para la implementación de la herramienta de simulación. De esta forma, es posible analizar y corregir los errores que presenta el programa creado y llegar a una versión final estable en los plazos marcados durante la planificación del proyecto.

- **Pérdida de datos / Avería de equipos:**

Para evitar que su impacto sea elevado, se deben realizar copias de seguridad con frecuencia de toda la información que se dispone del proyecto. Además, es recomendable, en este sentido, diversificar las fuentes de almacenamiento combinando, por ejemplo, un almacenamiento en la nube con el almacenamiento local en el ordenador de trabajo.

12) CONCLUSIONES Y VISIÓN DE FUTURO

En este proyecto se ha realizado en primer lugar un análisis de las nuevas familias de códigos definidas para el nuevo estándar 5G NR y del estado del arte del estándar DVB-S2 actualmente en uso para sistemas de comunicaciones por satélite DTH. Esto ha permitido evaluar la capa física de este sistema y los requerimientos necesarios para la implementación del módulo de codificación propuesto para 5G NR. Se ha verificado así la existencia de un nuevo código LDPC (BG1) que puede adaptarse para sustituir al actual definido en DVB-S2.

Se ha comprobado que modificar el módulo de codificación en un sistema de comunicaciones implica la realización de ciertas adaptaciones que permiten su correcto funcionamiento. En este caso, la principal limitación técnica encontrada radica en el pequeño valor del tamaño máximo de bloque aceptado por la nueva codificación en comparación con el tamaño de las tramas de DVB-S2. Se ha visto necesario segmentar y adaptar el resto de los bloques de la cadena de comunicación para lograr su implementación.

Se ha diseñado e implementado una nueva herramienta en MATLAB que permite simular una cadena de transmisión basada en DVB-S2 que sigue el modelo teórico esperado. Se ha desarrollado también una segunda herramienta que integra la nueva codificación LDPC propuesta en 5G NR. De esta forma, es posible evaluar y comparar el rendimiento de la solución planteada en estos dos escenarios para los distintos modos de funcionamiento de DVB-S2.

Se han identificado varios casos representativos. En los casos prácticos analizados, se ha obtenido una ganancia de entorno 0.2 dB con el uso de los nuevos códigos LDPC respecto a los ya existentes. En efecto, hay que comentar que, si bien es cierto que los nuevos códigos definidos en 5G aumentan el rendimiento debido a su desarrollo mejorado, éste se ve mermado por la necesidad de segmentar los paquetes de datos para adaptarlos al estándar DVB-S2.

Es importante señalar que una de las ventajas que produce la reducción del tamaño de los bloques radica en una mejora en cuanto a tiempo de procesado en las GPUs de los decodificadores al requerir menos memoria para realizar la decodificación.

Entre los futuros desarrollos de este proyecto, destaca la ampliación del estudio a nuevos casos prácticos que permitan extender los analizados en esta memoria. Otro desarrollo futuro consistiría en ampliar las funcionalidades de la herramienta creada para evaluar la idoneidad de la codificación propuesta en 5G NR en otro tipo de tecnologías como IEEE 802.11g o DVB-T2. Por último, también sería posible completar el desarrollo realizado en una maqueta totalmente real, es decir implementando la solución en un bloque transmisor/receptor físico que permitiese evaluar el sistema de comunicación en condiciones reales de transmisión.

REFERENCIAS

- [1] Franceschini, Ferrari y Raheli., *LDPC Coded Modulations*. Springer, 2009. ISBN 978-3-540-69455-7.
- [2] Miller, Vandome, y McBrewster., *Forward Error Correction*. VDM Publishing House Ltd., 2010. ISBN 978-6-130-80508-1.
- [3] Baldi, M. *QC-LDPC Code-Based Cryptography*. Spinger, 2014. ISBN 978-3-319-02555-1.
- [4] Gallager R. G., *Low – Density Parity Check Codes*. Cambridge, MA. MIT Press, 1963.
- [5] Andreadou, N. y Pavlidou, F-N. *Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) Codes for Deep Space and High Data Rate Applications*. Aristotle University of Thessaloniki. <https://ieeexplore.ieee.org/document/5286377>
- [6] Ullah, I. y Noor S. *Construction and Performance Evaluation of QC-LDPC Codes over Finite Fields*. Blekinge Institute of Technology, 2010. <http://www.diva-portal.org/smash/get/diva2:829786/FULLTEXT01.pdf>
- [7] Shannon C.E., *A Mathematical Theory of Communication*. Bell System Technical journal, July 1948.
- [8] Tahir B., Shwarz S. y Rupp M. *BER Comparison Between Convolutional, Turbo, LDPC, and Polar Codes*. Institute of Telecommunications Technische Universität (TU), 2017. https://publik.tuwien.ac.at/files/publik_262129.pdf
- [9] Kim H. *Coding and Modulation Techniques for High Spectral Efficiency Transmission in 5G and Satcom*. VTT Technical Research Centre of Finland, 2015. <http://ieeexplore.ieee.org/document/7362884>
- [10] Mihaylov Gr., Iliev T. B., Ivanova E. P., Stoyanov L. S., y Ihiev L., *Performance analysis of Low Density Parity Check Codes implemented in second generations of Digital Video Broadcasting standards*. University of

- Ruse, Department of Telecommunication, 2016.
<https://ieeexplore.ieee.org/document/7522195/>
- [11] Maunder R., *A Vision for 5G Channel Coding*. CTO AccelerComm, 2016.
<https://eprints.soton.ac.uk/401809/1/WhitePaper2.pdf>
- [12] Richardson T. y Kudekar S., *Design of Low-Density Parity Check Codes for 5G New Radio*. IEEE Communications Magazine (Volume: 56, Issue: 3), marzo de 2018. <https://ieeexplore.ieee.org/document/8316763/>
- [13] Zaidi A., Andersson M., Faxér S., Molés-Cases V.,y Wang Z. Designing for the future, The 5G NR Physical Layer. Ericsson Technology Review, 2017.
<https://www.ericsson.com/en/ericsson-technology-review/archive/2017/designing-for-the-future-the-5g-nr-physical-layer>
- [14] Gamage, H. *Waveforms and Channel Coding for 5G*. Faculty of Information Technology and Electrical Engineering. University of Honolulu, 2017.
<http://jultika.oulu.fi/files/nbnfioulu-201703161385.pdf>
- [15] ETSI EN 300 421 (V.1.1.2): “*Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services*”
http://www.etsi.org/deliver/etsi_en/300400_300499/300421/01.01.02_60/en_300421v010102p.pdf
- [16] ETSI EN 301 210: “*Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for Digital Satellite News Gathering (DSNG) and other contribution applications by satellite*”.
http://www.etsi.org/deliver/etsi_en/301200_301299/301210/01.01.01_60/en_301210v010101p.pdf
- [17] ETSI EN 302 307-1 (V.1.4.1): “*Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 1: DVB-S2*”
http://www.etsi.org/deliver/etsi_en/302300_302399/30230701/01.04.01_60/en_30230701v010401p.pdf

- [18] Martínez Rodrigue-Osorio R., Calvo Ramón M. “*Sistemas de comunicaciones por satélite. DVB-S y DVB-S2*”. <http://www.gr.ssr.upm.es/docencia/grado/csat/material/CSA08-8-DVB-S-S2-RCS.pdf>
- [19] 3rd Generation Partnership Project, *Technical Specification Group Radio Access Network; NR; Multiplexing and channel coding* (Release 15) TS 38.212 V15.1.1 www.3gpp.org/release-15
- [20] MathWorks, *DVB-S.2 Link, Including LDPC Coding*. MATLAB, 2018. https://es.mathworks.com/help/comm/examples/dvb-s-2-link-including-ldpc-coding.html?s_tid=gn_loc_drop
- [21] MathWorks, *Communications System Toolbox*. MATLAB, 2018. <https://es.mathworks.com/help/comm/>
- [22] MathWorks, *5G Library for LTE System Toolbox. User's Guide*. MATLAB, 2018.
- [23] SES. *Satellite Monitors, 2017*. <https://www.ses.com/ses-reach>

ANEXO 1: CÓDIGO MATLAB

Este anexo incluye el código desarrollado en MATLAB que permite realizar las simulaciones indicadas a lo largo de esta memoria.

1.1) Parámetros BCH

Esta función permite obtener los parámetros asociados al codificador externo BCH para poder crear su bloque posteriormente. Se basa en la tabla proporcionada por la ETSI en [17].

```
function [nBCH, kBCH, genBCH] = getbchparameters(R)

table5a = [1/4 16008 16200 12 64800
           1/3 21408 21600 12 64800
           2/5 25728 25920 12 64800
           1/2 32208 32400 12 64800
           3/5 38688 38880 12 64800
           2/3 43040 43200 10 64800
           3/4 48408 48600 12 64800
           4/5 51648 51840 12 64800
           5/6 53840 54000 10 64800
           8/9 57472 57600 8 64800
           9/10 58192 58320 8 64800];

rowidx = find(abs(table5a(:,1)-R)<.001);
kBCH = table5a(rowidx,2);
nBCH = table5a(rowidx,3);
tBCH = table5a(rowidx,4);

a8 = [1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 ...
      1 1 0 0 1 0 0 1 0 1 0 1 0 1 1 ...
      1 1 1 0 1 1 1 0 0 0 1 0 0 1 0 ...
      0 1 1 1 1 0 0 1 0 1 1 1 1 0 1 ...
      1 1 1 0 1 0 0 0 1 1 0 0 1 1 1 ...
      1 1 1 1 0 0 0 1 1 0 1 1 0 1 0 ...
      1 1 1 0 1 0 1 0 0 0 0 0 1 0 0 ...
      1 1 1 1 1 0 0 1 0 1 1 0 0 1 1 ...
      0 0 0 1 0 1 0 1 0 1 1];

a10 = [1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 ...
       0 1 0 0 0 0 1 1 0 0 1 1 1 0 1 ...
       1 0 1 1 1 1 1 1 1 0 0 0 0 1 0 ...
       1 0 1 0 0 0 1 1 0 0 1 1 0 0 0 ...
       1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 ...
       1 1 1 0 0 0 0 1 0 1 0 1 0 1 1 ...
       0 0 0 0 0 1 1 1 1 1 0 1 1 1 1 ...
       1 1 0 1 0 0 0 1 0 0 1 0 0 0 1 ...
       1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 ...
       0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 ...
       1 0 1 1 0 0 1 0 0 0 1];

a12 = [1 0 1 0 0 1 1 1 0 0 0 1 0 0 1 ...
       1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 ...
       0 1 1 1 0 0 0 0 1 0 0 0 1 0 1 ...
       1 1 0 0 0 1 0 1 0 0 0 1 0 0 0 ...
       1 1 1 0 0 0 1 0 1 0 0 0 0 1 1 ...
```

```

0 0 1 1 1 1 0 0 1 0 1 1 0 0 1 ...
1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 ...
0 0 1 1 0 1 0 1 0 0 0 0 1 0 0 ...
0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 ...
1 1 0 1 0 0 0 1 1 1 1 1 0 0 0 ...
1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 ...
0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 ...
1 0 1 0 1 1 1 1 0 0 1 1 1];

switch tBCH
case 8
    genBCH = a8;
case 10
    genBCH = a10;
case 12
    genBCH = a12;
end

```

1.2) Bloques de la arquitectura de DVB-S2

Esta función proporciona los bloques de la arquitectura de DVB-S2. En este caso, no se incluye el demodulador dado que se ha optado por incluirlo directamente en la simulación al depender de la varianza del canal de transmisión simulado.

```

%Bloques cadena DVB-S2
%Se incluyen:
%1. Fuente binaria
%2. Codificador externo BCH
%3. Codificador interno LDPC
%4. Entrelazado
%5. Modulado

%Canal AWGN y Demodulador añadido en la simulación

%9. Desentrelazado
%10. Decodificador interno LDPC
%11. Decodificador externo BCH

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               EMISOR                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               FUENTE                               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Paquete DVB-S2
numBytesPaquete = 188;
numBitsPaquete = 188*8; %1504

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CODIFICADOR EXTERNO BCH                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Parámetros
[nBCH, kBCH, BCHPolGen] = getbchparameters(codeRate);
desfasePrevioBCH = kBCH;
BCHPolPrimitivo = de2bi (65581, 'left-msb');
numPaquetesPorTramaBB = floor(kBCH/numBitsPaquete);
numBitsInfoPorPalabraCodigo = numPaquetesPorTramaBB*numBitsPaquete;

%Codificador BCH
codificadorBCH = comm.BCHEncoder('CodewordLength', nBCH, ...
    'MessageLength', kBCH, ...

```

```

'PrimitivePolynomialSource', 'Property', ...
'PrimitivePolynomial', BCHPolPrimitivo, ...
'GeneratorPolynomialSource', 'Property', ...
'GeneratorPolynomial', BCHPolGen, ...
'CheckGeneratorPolynomial', false);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% CODIFICADOR INTERNO LDPC %%%%%%%%%

%DVB-S2
%Parámetros
kLDPC = nBCH;
LDPCMatrizParidad = dvbs2ldpc(codeRate);
codificadorLDPC = comm.LDPCEncoder(LDPCMatrizParidad);

%DVB-S2
%Parámetros
numFilas = nLDPC;
filas = (1:numFilas).';

%No hay entrelazado en BPSK ni QPSK

if isequal (tipoModulacion, '8PSK')
    numCol = 3;
    filasTemp = reshape(filas, nLDPC/numCol, numCol).';
    if codeRate == 3/5 %Caso especial
        filasTemp = flipud(filasTemp); %Inverrrtir orden array
    end
    filas = filasTemp(:);

elseif isequal(tipoModulacion, '16APSK')
    numCol = 4;
    filasTemp = reshape(filas, nLDPC/numCol, numCol).';
    filas = filasTemp(:);

elseif isequal(tipoModulacion, '32APSK')
    numCol = 5;
    filasTemp = reshape(filas, nLDPC/numCol, numCol).';
    filas = filasTemp(:);

end

%Entrelazador
entrelazador = comm.BlockInterleaver(filas);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULADOR %%%%%%%%%

%Parametros
switch tipoModulacion

    case 'BPSK'
        Qy = [+1; -1];
        constelacion = complex (Qy);
        mapeadoSimbolo = [0 1];
        offsetFase = 0;

    case 'QPSK'
        Qy = [+1; +1; -1; -1];
        Iy = [+1; -1; +1; -1];
        constelacion = (Qy + 1i*Iy)/sqrt(2);
        mapeadoSimbolo = [0 2 3 1];
        offsetFase = pi/4;

    case '8PSK'
        A = sqrt(1/2);
        Qy = [+A +1 -1 -A 0 +A -A 0].';

```

```

Iy = [+A 0 0 -A 1 -A +A -1].';
constelacion = Qy + 1i*Iy;
mapeadoSimbolo = [1 0 4 6 2 3 7 5];
offsetFase = 0;

case '16APSK'
constelacion = dvbsapskmod((0:15)', 16, 's2', ...
    codeRate2, 'UnitAveragePower', true);
mapeadoSimbolo = [12 14 15 13 4 0 8 10 2 6 7 3 11 9 1 5];
offsetFase = [pi/4 pi/12];

case '32APSK'
constelacion = dvbsapskmod((0:31)', 32, 's2', ...
    codeRate2, 'UnitAveragePower', true);
mapeadoSimbolo = [15 13 29 31 14 6 7 5 4 12 28 20 21 23 22 30 ...
    11 10 2 3 1 0 8 9 25 24 16 17 19 18 26 27];
offsetFase = [pi/4 pi/12 pi/16];

otherwise
error(message(';Error en el tipo de modulaci3n!'));
end

M = length(constelacion);
numeroBitsPorSimbolo = log2(M);
numeroSimbolosPorPalabraCodigo = nLDPC/numeroBitsPorSimbolo;

%Modulador
modulador = comm.PSKModulator('ModulationOrder', 2^numeroBitsPorSimbolo, ...
    'BitInput', true, ...
    'PhaseOffset', offsetFase, ...
    'SymbolMapping', 'Custom', ...
    'CustomSymbolMapping', mapeadoSimbolo);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RECEPTOR %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DECODIFICADOR EXTERNO BCH %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
decodificadorBCH = comm.BCHDecoder('CodewordLength', nBCH, ...
    'MessageLength', kBCH, ...
    'PrimitivePolynomialSource', 'Property', ...
    'PrimitivePolynomial', BCHPolPrimitivo, ...
    'GeneratorPolynomialSource', 'Property', ...
    'GeneratorPolynomial', BCHPolGen, ...
    'CheckGeneratorPolynomial', false);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DECODIFICADOR INTERNO LDPC %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
decodificadorLDPC = comm.LDPCDecoder(LDPCMatrizParidad, ...
    'IterationTerminationCondition', 'Parity check satisfied', ...
    'MaximumIterationCount', maxIterLDPC, ...
    'NumIterationsOutputPort', true);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DESENTRELAZADOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
desentrelazador = comm.BlockDeinterleaver(filas);

```

1.3) Cadena DVB-S2 simple

El siguiente script permite simular una cadena DVB-S2 simple. Permite calcular el BER asociado a un valor fijo de E_s/N_0 proporcionado por el usuario para un 3nico tipo de modulaci3n y *code rate* indicado.

```

% cadenaSimple_DVBS2
%%% PRIMERA CADENA DVB-S2 SIMPLE %%%
% Jon Maseda García, 2018

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Inicialización parámetros %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Longitud tramas
nLDPC = 64800; %Long
%nLDPC = 16200; %Short

tipoModulacion = 'QPSK'; %Modulación
codeRate2 = '1/2'; %Formato texto
codeRate = str2num(codeRate2); %Code rate

EsNodB = -1; %Relación Es/No (en dB)
numTramas = 1; %Número de tramas a simular

maxIterLDPC = 50; %Número máximo de iteraciones LDPC

%Cargado de los bloques DVB-S2
bloquesDVBS2;

numBitErroneos = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CANAL %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Ruido
varianzaRuido = 1/(10^(EsNodB/10));
varianzaRuidoEstimada = varianzaRuido/(2*sin(pi/M));

%Canal
canal = comm.AWGNChannel('NoiseMethod', 'Variance',...
    'Variance', varianzaRuido);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DEMODULADOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
demodulador = comm.PSKDemodulator('ModulationOrder', 2^numeroBitsPorSimbolo,
...
    'BitOutput', true, ...
    'PhaseOffset', offsetFase, ...
    'SymbolMapping', 'Custom', ...
    'CustomSymbolMapping', mapeadoSimbolo, ...
    'DecisionMethod', 'Approximate log-likelihood ratio', ...
    'Variance', varianzaRuido);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SIMULACIÓN TRANSMISIÓN %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Trama BB
tramaBBTx = false(kBCH,1); %Array vacío tamaño kBCHx1

%EMISOR
bitsTx = randi([0 1], numBitsInfoPorPalabraCodigo, 1);
tramaBBTx(1:numBitsInfoPorPalabraCodigo)=logical(bitsTx);
salidaCodificadorBCH = step(codificadorBCH, tramaBBTx);
salidaCodificadorLDPC = step(codificadorLDPC, salidaCodificadorBCH);
salidaEntrelazador = step(entrelazador, salidaCodificadorLDPC);

if M == 4 || M == 8
    salidaModulador = step(modulador, salidaEntrelazador);
else
    salidaModulador = dvbsapskmod(salidaEntrelazador, M, 's2', ...

```

```

        codeRate2, 'InputType', 'bit', 'UnitAveragePower', true);
end

%CANAL
salidaCanal = step(canal, salidaModulador);

%RECEPTOR
if M == 4 || M == 8
    salidaDemodulador = step(demodulador, salidaCanal);
else
    salidaDemodulador = dvbsapskdemod(salidaCanal, M, 's2', ...
        codeRate2, 'OutputType', 'approx1lr', 'NoiseVar', ...
        varianzaRuido, 'UnitAveragePower', true);
end

salidaDesentrelazador = step(desentrelazador, salidaDemodulador);
[salidaDecodificadorLDPC, numIterLDPC] = step(decodificadorLDPC,
salidaDesentrelazador);
salidaDecodificadorBCH = step(decodificadorBCH, salidaDecodificadorLDPC);
tramaBBRx = salidaDecodificadorBCH;
bitsRx = salidaDecodificadorBCH(1:numBitsInfoPorPalabraCodigo);

%ERRORES
numBitErroneos = numBitErroneos + sum((bitsRx.')~=(bitsTx.'));
numBitsTotales = length(bitsRx)*numTramas;
%Cálculo BER
BER = numBitErroneos/(length(bitsRx)*numTramas)

```

1.4) Cadena DVB-S2 mejorada

El siguiente script retoma el del anterior anexo 1.3) incluyendo la posibilidad de realizar simulaciones simultáneas para distintos valores de *code rate* y E_s/N_0 proporcionados. Asimismo, permite realizar gráficas de las constelaciones correspondientes al emisor y receptor así como de la relación E_s/N_0 respecto al BER.

```

% cadenaMejorada_DVBS2
%%% CADENA DVB-S2 MEJORADA %%%
% Jon Maseda García, 2018

% Añadidas las siguientes funcionalidades
% 1. Posibilidad de simular para diferentes tramas
% 2. Posibilidad de simular para diferentes EsN0dB
% 3. Posibilidad de simular para diferentes Code Rates
% 3. Representación gráfica EsN0dB vs. BER
% 4. Representación gráfica constelación emisor vs. receptor

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Inicialización parámetros %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nLDPC = 64800; %Long
%nLDPC = 16200; %Short

tipoModulacion = 'QPSK'; %Modulación
codeRate2 = ["1/2", "2/3", "4/5", "9/10"]; %Formato texto

codeRate = [1/2 2/3 4/5 9/10]; %Code rate
EsNodB = -1:1:10; %Relación Es/No (en dB)
numTramas = 1; %Número de tramas a simular

```

```

maxIterLDPC = 50; %Número máximo de iteraciones LDPC

BER = ones(length(codeRate), length(EsNodB));
EsNo = ones(1, length(EsNodB));

%Repetición bucle para cada Code Rate
for k=1:length(codeRate)
    codeRate = str2num(codeRate2(k));

    %%Cargado de los bloques DVB-S2
    bloquesDVBS2;

    %Repetición bucle para cada EsNodB
    for i=1:length(EsNodB)
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %                               CANAL                               %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %Ruido
        varianzaRuido = 1/(10^(EsNodB(i)/10));
        varianzaRuidoEstimada = varianzaRuido/(2*sin(pi/M));

        %Canal
        canal = comm.AWGNChannel('NoiseMethod', 'Variance',...
            'Variance', varianzaRuido);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        DEMODULADOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        demodulador = comm.PSKDemodulator('ModulationOrder',
2^numeroBitsPorSimbolo, ...
        'BitOutput', true, ...
        'PhaseOffset', offsetFase, ...
        'SymbolMapping', 'Custom', ...
        'CustomSymbolMapping', mapeadoSimbolo, ...
        'DecisionMethod', 'Approximate log-likelihood ratio', ...
        'Variance', varianzaRuido);

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        %                               SIMULACIÓN TRANSMISIÓN           %
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

        %Inicialización número bits erróneos
        numBitErroneos = 0;
        %Trama BB
        tramaBBTx = false(kBCH,1); %Array vacío tamaño kBCHx1

        %Repetición bucle para cada una de las tramas
        for j=1:numTramas

            %EMISOR
            bitsTx = randi([0 1], numBitsInfoPorPalabraCodigo, 1);
            tramaBBTx(1:numBitsInfoPorPalabraCodigo)=logical(bitsTx);
            salidaCodificadorBCH = step(codificadorBCH, tramaBBTx);
            salidaCodificadorLDPC = step(codificadorLDPC,
salidaCodificadorBCH);
            salidaEntrelazador = step(entrelazador, salidaCodificadorLDPC);

            if M == 4 || M == 8
                salidaModulador = step(modulador, salidaEntrelazador);
            else
                salidaModulador = dvbsapskmod(salidaEntrelazador, M, 's2', ...
                    char(codeRate2(k)), 'InputType', 'bit',
'UnitAveragePower', true);
            end

            %CANAL
            salidaCanal = step(canal, salidaModulador);

```



```

        %RECEPTOR
        if M == 4 || M == 8
            salidaDemodulador = step(demodulador, salidaCanal);
        else
            salidaDemodulador = dvbsapskdemod(salidaCanal, M, 's2', ...
                char(codeRate2(k)), 'OutputType', 'approx1lr', 'NoiseVar',
...
                varianzaRuido, 'UnitAveragePower', true);
        end
        salidaDesentrelazador = step(desentrelazador, salidaDemodulador);
        [salidaDecodificadorLDPC, numIterLDPC] = step(decodificadorLDPC,
salidaDesentrelazador);
        salidaDecodificadorBCH = step(decodificadorBCH,
salidaDecodificadorLDPC);
        tramaBBRx = salidaDecodificadorBCH;
        bitsRx = salidaDecodificadorBCH(1:numBitsInfoPorPalabraCodigo);

        %ERRORES
        numBitErroneos = numBitErroneos + sum((bitsRx.')~=(bitsTx.'));
    end

    numBitsTotales = length(bitsRx)*numTramas;
    BER(k,i) = numBitErroneos/(length(bitsRx)*numTramas);
end

end

%%% GRÁFICAS %%%
%Constelación MODULADOR/DEMULADOR
figure(2);

subplot(2,1,1);
simbolosTx = salidaModulador.';
plot(real(simbolosTx), imag(simbolosTx), '*');
title('Constelación - Transmisor');
ylabel('Q');
xlabel('I');
axis([-2 +2 -2 +2]);

subplot(2,1,2)
simbolosRx = salidaCanal.';
plot(real(simbolosRx), imag(simbolosRx), '*');
title('Constelación - Receptor');
ylabel('Q');
xlabel('I');
axis([-2 +2 -2 +2]);

%Curvas BER respecto EsNo (dB)
figure(1);
semilogy(EsNodB, BER(1,:), 'r-*');
xlabel('Es / N0 (dB)');
ylabel('BER');
hold on
semilogy(EsNodB, BER(2,:), 'g-*');
hold on
semilogy(EsNodB, BER(3,:), 'b-*');
hold on
semilogy(EsNodB, BER(4,:), 'm-*');
legend('1/2', '2/3', '4/5', '9/10');

```

1.5) Nueva codificación LDPC

El siguiente script permite implementar la nueva codificación LDPC propuesta para 5G en MATLAB. Para su desarrollo, se han tenido en cuenta las consideraciones indicadas en el apartado 8.3).

```
% implementacionLDPC_5G
%%% IMPLEMENTACIÓN DEL MÓDULO LDPC DEFINIDO EN 5G
%%% AUTONOMO %%%

%Parámetros
%Se toma como ejemplo QPSK CR=3/4
numBits=48600; %valor de kLDPC
nLDPC = 64800;

%Parámetros codificador LDPC
nbg = 1; %Matriz de base BG1
chs.NBG = 1;
chs.rv = 0; %
chs.Modulation = 'QPSK'; %Modulación
chs.NLayers = 1; %No hay reenvío
maxIterLDPC = 50; %Número máximo de iteraciones

bitsTx = randi([0 1], numBits, 1);

%CODIFICACION LDPC
salidaCRC = h5gCRCEncode(bitsTx,'24B');
salidaSegmentada = h5gCodeBlockSegment(salidaCRC,nbg);
salidaCodificadorLDPC = h5gLDPCEncode(salidaSegmentada,nbg);
salidaCodificadorLDPCAdaptada =
h5gRateMatchLDPC(salidaCodificadorLDPC,nLDPC+24, chs.rv, chs);

%AUSENCIA DE CANAL

%DECODIFICACION LDPC
%Soft bits
recibidoSoft = double(1-2*salidaCodificadorLDPCAdaptada); %Cambio a bits soft
para entrada en decodificador
salidaDecodificadorDesadaptada =
h5gRateRecoverLDPC(recibidoSoft,numBits,chs.rv,chs);
salidaDecodificadorLDPC = h5gLDPCDecode(salidaDecodificadorDesadaptada, nbg,
maxIterLDPC);
[salidaDesegmentada,err] =
h5gCodeBlockDesegment(salidaDecodificadorLDPC,nbg,numBits+24);
%Eliminación del CRC
salidaDesegmentada2 = salidaDesegmentada_5G(1:numBits);

numBitErroneos = sum((salidaDesegmentada2.')~=(bitsTx.'));
```

1.6) Integración del nuevo módulo LDPC en DVB-S2

En este script se detallan las modificaciones efectuadas en la cadena DVB-S2 creada para poder integrar el nuevo módulo LDPC. Para ello, se crea en primer lugar una función similar a la del punto 1.2) que permite definir los bloques de DVB-S2 adaptados al módulo LDPC del estándar 5G.

```

%Bloques cadena DVB-S2 adaptados para la implementación del módulo LDPC 5G
%Se incluyen:
%1. Codificador externo BCH
%2. Entrelazado
%3. Modulado
%4. Demodulador
%5. Desentrelazado
%6. Decodificador externo BCH

%Canal AWGN y codificador-decodificador LDPC añadidos en la simulación

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               EMISOR
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               FUENTE
%                               %%%%%%%%%%%%%%%
%Paquete DVB-S2
numBytesPaquete = 188;
numBitsPaquete = 188*8; %1504

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               CODIFICADOR EXTERNO BCH
%                               %%%%%%%%%%%%%%%
%Parámetros
[nBCH, kBCH, BCHPolGen] = getbchparameters(codeRate);
desfasePrevioBCH = kBCH;
BCHPolPrimitivo = de2bi (65581, 'left-msb');
numPaquetesPorTramaBB = floor(kBCH/numBitsPaquete);
numBitsInfoPorPalabraCodigo = numPaquetesPorTramaBB*numBitsPaquete;

codificadorBCH_5G = comm.BCHEncoder('CodewordLength', nBCH, ...
    'MessageLength', kBCH, ...
    'PrimitivePolynomialSource', 'Property', ...
    'PrimitivePolynomial', BCHPolPrimitivo, ...
    'GeneratorPolynomialSource', 'Property', ...
    'GeneratorPolynomial', BCHPolGen, ...
    'CheckGeneratorPolynomial', false);

kLDPC = nBCH;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               ENTRELAZADOR
%                               %%%%%%%%%%%%%%%

%5G
%Parámetros
numFilas_5G = nLDPC+24;
filas_5G = (1:numFilas_5G).';

%No hay entrelaado en BPSK ni QPSK

if isequal (tipoModulacion, '8PSK')
    numCol = 3;
    filasTemp = reshape(filas_5G, numFilas_5G/numCol, numCol).';
    if codeRate == 3/5 %Caso especial
        filasTemp = flipud(filasTemp); %Inverrtir orden array
    end
    filas_5G = filasTemp(:);

elseif isequal(tipoModulacion, '16APSK')
    numCol = 4;
    filasTemp = reshape(filas_5G, numFilas_5G/numCol, numCol).';
    filas_5G = filasTemp(:);

elseif isequal(tipoModulacion, '32APSK')
    numCol = 5;
    filasTemp = reshape(filas_5G, numFilas_5G/numCol, numCol).';
    filas_5G = filasTemp(:);

```

```

end

%Entrelazador
entrelazador_5G = comm.BlockInterleaver(filas_5G);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% MODULADOR %%%%%%%%%
%5G
switch tipoModulacion

    case 'BPSK'
        Qy = [+1; -1];
        constelacion = complex (Qy);
        mapeadoSimbolo = [0 1];
        offsetFase = 0;

    case 'QPSK'
        Qy = [+1; +1; -1; -1];
        Iy = [+1; -1; +1; -1];
        constelacion = (Qy + 1i*Iy)/sqrt(2);
        mapeadoSimbolo = [0 2 3 1];
        offsetFase = pi/4;

    case '8PSK'
        A = sqrt(1/2);
        Qy = [+A +1 -1 -A 0 +A -A 0].';
        Iy = [+A 0 0 -A 1 -A +A -1].';
        constelacion = Qy + 1i*Iy;
        mapeadoSimbolo = [1 0 4 6 2 3 7 5];
        offsetFase = 0;

    case '16APSK'
        constelacion = dvbsapskmod((0:15)', 16, 's2', ...
            codeRate2, 'UnitAveragePower', true);
        mapeadoSimbolo = [12 14 15 13 4 0 8 10 2 6 7 3 11 9 1 5];
        offsetFase = [pi/4 pi/12];

    case '32APSK'
        constelacion = dvbsapskmod((0:31)', 32, 's2', ...
            codeRate2, 'UnitAveragePower', true);
        mapeadoSimbolo = [15 13 29 31 14 6 7 5 4 12 28 20 21 23 22 30 ...
            11 10 2 3 1 0 8 9 25 24 16 17 19 18 26 27];
        offsetFase = [pi/4 pi/12 pi/16];

    otherwise
        error(message('¡Error en el tipo de modulación!'));
end

M = length(constelacion);
numeroBitsPorSimbolo = log2(M);
numeroSimbolosPorPalabraCodigo = nLDPC/numeroBitsPorSimbolo;

modulador_5G = comm.PSKModulator('ModulationOrder', 2^numeroBitsPorSimbolo,
...
    'BitInput', true, ...
    'PhaseOffset', offsetFase, ...
    'SymbolMapping', 'Custom', ...
    'CustomSymbolMapping', mapeadoSimbolo);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RECEPTOR %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DECODIFICADOR EXTERNO BCH %%%%%%%%%
decodificadorBCH_5G = comm.BCHDecoder('CodewordLength', nBCH, ...
    'MessageLength', kBCH, ...

```

```

'PrimitivePolynomialSource', 'Property', ...
'PrimitivePolynomial', BCHPolPrimitivo, ...
'GeneratorPolynomialSource', 'Property', ...
'GeneratorPolynomial', BCHPolGen, ...
'CheckGeneratorPolynomial', false);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DESENTRELAZADOR %%%%%%%%%
%5G
desentrelazador_5G = comm.BlockDeinterleaver(filas_5G);

```

El siguiente script, que hace uso de la función anterior, permite simular la nueva cadena de transmisión.

```

% cadenaNuevaCodificacion5G_DVBS2
%%% CADENA DVB-S2 CON BLOQUE DE CODIFICACIÓN MODIFICADO %%%
% Permite simular la nueva codificación LDPC 5G implementada en el sistema
DVB-S2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Inicialización parámetros %%%%%%%%%

nLDPC = 64800; %Long
%nLDPC = 16200; %Short

tipoModulacion = '16APSK'; %Modulación
codeRate2 = '3/4'; %Formato texto
codeRate = str2num(codeRate2); %Code rate

EsNodB = -1:1:1; %Relación Es/No (en dB)
numTramas = 1; %Número de tramas a simular

maxIterLDPC = 50; %Número máximo de iteraciones LDPC

%Inicialización array BER y EsN0
BER = ones(1, length(EsNodB));
EsNo = ones(1, length(EsNodB));

%Parámetros codificador LDPC 5G
nbg = 1;
chs.NBG = 1;
chs.rv = 0;
chs.Modulation = tipoModulacion;
chs.NLayers = 1;
outlen = nLDPC + 24;

%Cargado bloques cadena DVBS2 adaptado a la nueva codificación
% + Entrelazado adaptado
bloquesDVBS2_5G;

for i=1:length(EsNodB)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % CANAL %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %Ruido
    varianzaRuido = 1/(10^(EsNodB(i)/10));
    varianzaRuidoEstimada = varianzaRuido/(2*sin(pi/M));

    %Canal
    canal_5G = comm.AWGNChannel('NoiseMethod', 'Variance', ...
        'Variance', varianzaRuido);

    %Demodulador

```

```

%5G
demodulador_5G = comm.PSKDemodulator('ModulationOrder',
2^numeroBitsPorSimbolo, ...
'BitOutput', true, ...
'PhaseOffset', offsetFase, ...
'SymbolMapping', 'Custom', ...
'CustomSymbolMapping', mapeadoSimbolo, ...
'DecisionMethod', 'Approximate log-likelihood ratio', ...
'Variance', varianzaRuido);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%          SIMULACIÓN TRANSMISIÓN
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Inicialización número bits erróneos
numBitErroneos = 0;

%Trama BB
tramaBBTx = false(kBCH,1); %Array vacío tamaño kBCHx1

for j=1:numTramas

%EMISOR
bitsTx = randi([0 1], numBitsInfoPorPalabraCodigo, 1);
tramaBBTx(1:numBitsInfoPorPalabraCodigo)=logical(bitsTx);
salidaCodificadorBCH_5G = step(codificadorBCH_5G, tramaBBTx);

%NUEVO CODIFICADOR LDPC 5G
salidaCRC = h5gCRCEncode(salidaCodificadorBCH_5G,'24B');
salidaSegmentada = h5gCodeBlockSegment(salidaCRC,nbg);
salidaCodificadorLDPC_5G = h5gLDPCEncode(salidaSegmentada,nbg);
salidaCodificadorLDPCAdaptada_5G =
h5gRateMatchLDPC_2(salidaCodificadorLDPC_5G,outlen, chs.rv, chs);
salidaEntrelazador = step(entrelazador_5G,
salidaCodificadorLDPCAdaptada_5G);

if M == 4 || M == 8
    salidaModulador = step(modulador_5G, salidaEntrelazador);
else
    salidaModulador = dvbsapskmod(salidaEntrelazador, M, 's2', ...
        codeRate2, 'InputType', 'bit', 'UnitAveragePower', true);
end

%CANAL
salidaCanal = step(canal_5G, salidaModulador);

%RECEPTOR

if M == 4 || M == 8
    salidaDemodulador = step(demodulador_5G, salidaCanal);
else
    salidaDemodulador = dvbsapskdemod(salidaCanal, M, 's2', ...
        codeRate2, 'OutputType', 'approxllr', 'NoiseVar', ...
        varianzaRuido, 'UnitAveragePower', true);
end

salidaDesentrelazador = step(desentrelazador_5G, salidaDemodulador);

%NUEVO DECODIFICADOR LDPC 5G
salidaDecodificadorDesadaptada =
h5gRateRecoverLDPC_2(salidaDesentrelazador,kLDPC,chs.rv,chs);
salidaDecodificadorLDPC = h5gLDPCDecode(salidaDecodificadorDesadaptada,
nbg, maxIterLDPC);
[salidaDesegmentada,err] =
h5gCodeBlockDesegment(salidaDecodificadorLDPC,nbg,kLDPC+24);

```

```

    salidaDesegmentada2 = salidaDesegmentada(1:(length(salidaDesegmentada)-
24));

    salidaDecodificadorBCH = step(decodificadorBCH_5G, salidaDesegmentada2);
    tramaBBRx = salidaDecodificadorBCH;
    bitsRx = salidaDecodificadorBCH(1:numBitsInfoPorPalabraCodigo);

    %ERRORES

    numBitErroneos = numBitErroneos + sum((bitsRx.')~=(bitsTx.'));
end

    numBitsTotales = length(bitsRx)*numTramas;
    BER(i) = numBitErroneos/(length(bitsRx)*numTramas);
end

```

ANEXO 2. RESULTADOS ADICIONALES

En este anexo se incluyen algunos resultados destacados de las simulaciones realizadas para las modulaciones 8PSK, 16PSK y 32PSK en la cadena de transmisión DVB-S2. Se amplía de esta manera, los resultados obtenidos en el apartado 8.2.3).

- **8PSK:**

Tabla 26. Parámetros simulación 8PSK

<i>Code rate</i>	2/3, 3/4, 5/6, 9/10
E_s/N_0	3 a 11 dB (en pasos de 0.5 dB)
Número de tramas	20
Modulación	8PSK

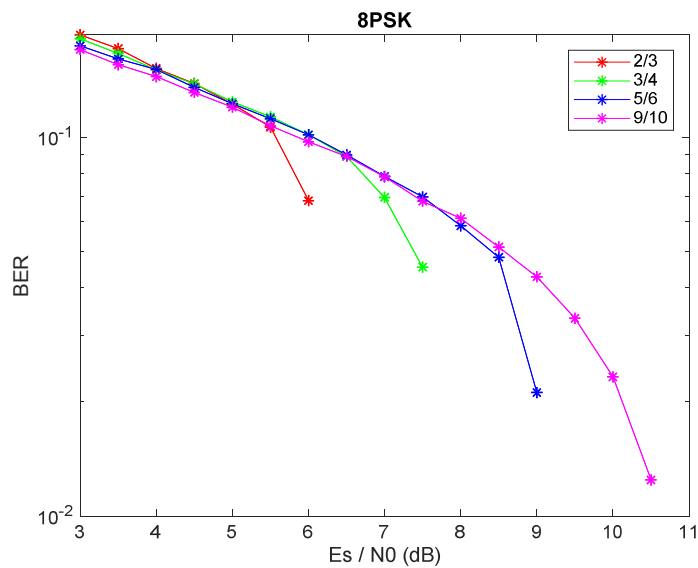


Figura 22. Relación BER vs. E_s/N_0 (8PSK)

- **16APSK:**

Tabla 27. Parámetros simulación 16APSK

<i>Code rate</i>	2/3, 3/4, 5/6, 9/10
E_s/N_0	6 a 14 dB (en pasos de 0.5 dB)
Número de tramas	20
Modulación	16APSK

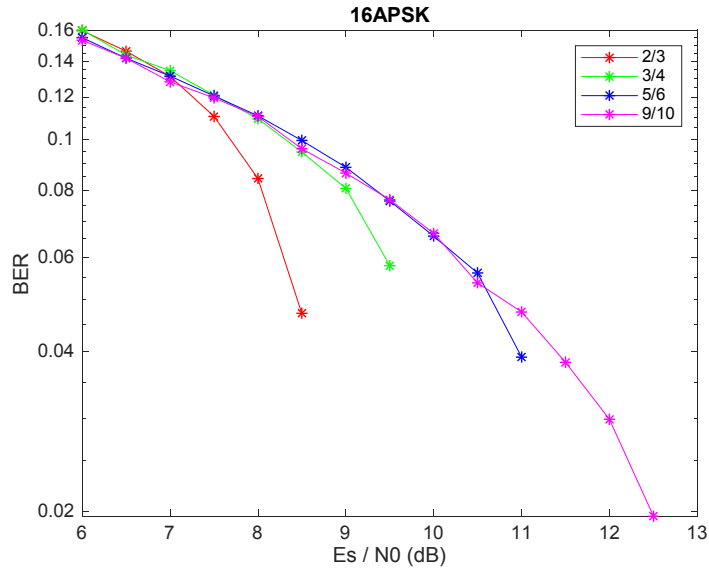


Figura 23. Relación BER vs. E_s/N_0 (16PSK)

- **32APSK:**

Tabla 28. Parámetros simulación 32APSK

Code rate	3/4, 5/6, 8/9, 9/10
E_s/N_0	9 a 17 dB (en pasos de 0.5 dB)
Número de tramas	20
Modulación	32APSK

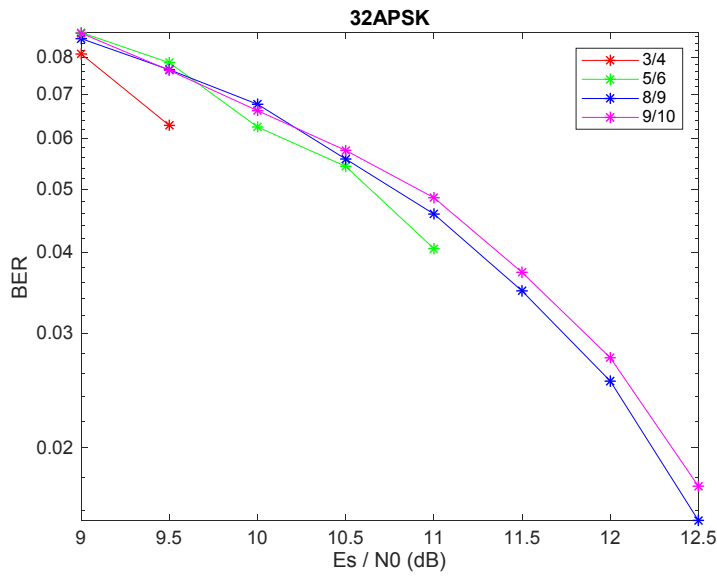


Figura 24. Relación BER vs. E_s/N_0 (32APSK)

ANEXO 3: RENDIMIENTO IDEAL EN DVB-S2

En este anexo, se incluye la tabla proporcionada en el estándar DVB-S2 [17] que relaciona cada modulación y *code rate* utilizado con el valor ideal de E_s/N_0 obtenido en condiciones de propagación cuasi libres de error ($PER = 10^{-7}$). La utilización de esta tabla facilita verificar si las simulaciones realizadas se asemejan a lo esperado teóricamente.

Modo	Eficiencia espectral	E_s/N_0 (dB) ideal
QPSK 1/4	0,490243	-2,35
QPSK 1/3	0,656448	-1,24
QPSK 2/5	0,789412	-0,30
QPSK 1/2	0,988858	1,00
QPSK 3/5	1,188304	2,23
QPSK 2/3	1,322253	3,10
QPSK 3/4	1,487473	4,03
QPSK 4/5	1,587196	4,68
QPSK 5/6	1,654663	5,18
QPSK 8/9	1,766451	6,20
QPSK 9/10	1,788612	6,42
8PSK 3/5	1,779991	5,50
8PSK 2/3	1,980636	6,62
8PSK 3/4	2,228124	7,91
8PSK 5/6	2,478562	9,35
8PSK 8/9	2,646012	10,69
8PSK 9/10	2,679207	10,98
16APSK 2/3	2,637201	8,97
16APSK 3/4	2,966728	10,21
16APSK 4/5	3,165623	11,03
16APSK 5/6	3,300184	11,61
16APSK 8/9	3,523143	12,89
16APSK 9/10	3,567342	13,13
32APSK 3/4	3,703295	12,73
32APSK 4/5	3,951571	13,64
32APSK 5/6	4,119540	14,28
32APSK 8/9	4,397854	15,69
32APSK 9/10	4,453027	16,05



BILBOKO
INGENIARITZA
ESKOLA
ESCUELA
DE INGENIERÍA
DE BILBAO