



IMPLEMENTACIÓN DE LA CAPA FÍSICA DEL ESTÁNDAR WiMAX-2004

Trabajo Final de Grado

Realizado en

**Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona**

Universitat Politècnica de Catalunya

por

Joel Padilla Pinedo

En cumplimiento parcial

de los requisitos para el grado en

Ingeniería en Sistemas de Telecomunicación

Tutor: Ferran Casadevall Palacio

Barcelona, Junio 2016

Resumen

Las comunicaciones en entornos rurales, polígonos industriales fuera del casco urbano, etc., en donde el número de usuarios del sistema, y por lo tanto, el tráfico cursado es relativamente bajo, siempre han supuesto un gran reto para los operadores de telecomunicaciones, debido a que la construcción de infraestructuras en estas zonas no es viable (o si lo es presenta muchas dificultades) desde el punto de vista económico. Por ello, se han buscado otras alternativas para poder llevar servicios de telecomunicación en dichas zonas y en general a zonas de difícil acceso. En este entorno, surge una tecnología llamada WiMAX que permite aportar soluciones prometedoras al problema de las comunicaciones en este tipo de entornos.

En este proyecto se ha realizado una implementación de la capa física del estándar WiMAX-2004 mediante el uso de una herramienta de Software Defined Radio (SDR). Para ello se ha desarrollado un modelo usando la aplicación Simulink de Matlab.

Resum

Les comunicacions en entorns rurals, polígons industrials fora del nucli urbà, etc., on el nombre d'usuaris del sistema, i per tant, el trànsit cursat és relativament baix, sempre han suposat un gran repte per als operadors de telecomunicacions, a causa del fet que la construcció d'infraestructures en aquestes zones no és viable (o si ho és presenta moltes dificultats) des del punt de vista econòmic. Per això, s'han buscat altres alternatives per poder dur serveis de telecomunicació en aquestes zones i en general a zones de difícil accés. En aquest entorn, sorgeix una tecnologia anomenada WiMAX que permet aportar solucions prometedores al problema de les comunicacions en aquest tipus d'entorns.

En aquest projecte s'ha realitzat una implementació de la capa física de l'estàndard WiMAX-2004 mitjançant l'ús d'una eina de Software Defined Radio (SDR). Per a això s'ha desenvolupat un model fent ús de l'aplicació Simulink de Matlab.

Abstract

Communications in rural areas, industrial estates outside the village, etc., where the number of users of the system, and therefore the traffic carried is relatively low, have always been a great challenge for telecommunications operators due to build infrastructure in these areas is not feasible (or if it is present many difficulties) from the economic point of view. Therefore, technologic alternatives for providing telecommunications services in these areas, and generally at inaccessible areas, have been envisaged. In this context, a technology, called WiMAX, seems to provide promising solutions to the problem of communications in such environments.

In this project an implementation of the physical layer of WiMAX-2004 standard using a tool of Software Defined Radio (SDR) has been carried out. In particular, we have developed a physical layer model using Matlab Simulink application.

Agradecimientos

Este proyecto no se habría podido llevar a cabo sin el apoyo y el ánimo de muchas personas que me han dado la fuerza suficiente para poder desarrollarlo.

En primer lugar, dar gracias al tutor de este proyecto, Ferran Casadevall, por el apoyo y la ayuda proporcionados, siempre con una mirada positiva y alentadora, tanto a la hora de desarrollar el trabajo como por su esfuerzo en corregir y perfeccionar el trabajo realizado.

En segundo lugar, quisiera dar gracias a los compañeros que me han acompañado durante el grado para transformar los momentos de presión y dificultad en buenos ratos. Además, quisiera agradecer a los amigos que me han animado y comprendido durante el desarrollo de este proyecto, ayudándome a afrontar y superar cualquier obstáculo.

En tercer lugar, dar gracias a mis padres por hacer posible mis estudios y por su apoyo en todo momento. También agradecer a toda mi familia por su apoyo incondicional.

A todos vosotros os doy las gracias.

Historial de revisiones y registro de aprobación

Revisión	Fecha	Objetivo
0	23/05/2016	Creación del documento
1	05/06/2016	Revisión del documento
2	07/06/2016	Corrección del documento
3	17/06/2016	Revisión final

LISTADO DE DISTRIBUCIÓN DEL DOCUMENTO

Nombre	e-mail
Joel Padilla Pinedo	padillapinedojoel@gmail.com
Ferran Casadevall Palacio	ferranc@tsc.upc.edu

Escrito por:		Revisado y aprobado por:	
Fecha	17/06/2016	Fecha	22/06/2016
Nombre	Joel Padilla Pinedo	Nombre	Ferran Casadevall Palacio
Posición	Autor del proyecto	Posición	Tutor del proyecto

Tabla de contenidos

Resumen	1
Resum	2
Abstract	3
Agradecimientos	4
Historial de revisiones y registro de aprobación	5
Tabla de contenidos	6
Listado de Figuras	8
Listado de Tablas:	10
1. Introducción	11
1.1. Objetivos	12
1.2. Cronograma	14
2. Marco referencial:	15
2.1. Sistema WiMAX	15
2.2. Software Defined Radio	16
3. Metodología:	17
3.1. Introducción a Simulink	17
3.1.1. Creación de un modelo	17
3.2. Bloques y dimensionado	19
3.2.1. Transmisor	20
3.2.2. Codificador y Decodificador de canal	21
3.2.2.1. Codificador Reed-Solomon	21
3.2.2.2. Codificador Convolutacional (CC)	23
3.2.3. Moduladores	27
3.2.4. Moduladores OFDM	29
3.2.5. Canal	32
3.2.6. Sincronización y estimación del canal	33
3.2.7. Otros bloques	35
3.2.8. Interfaz de Usuario	36
4. Resultados	39
4.1. Bertool	39
4.2. BER vs E_b/N_0 modulación BPSK	41
4.3. BER vs E_b/N_0 modulación QPSK	42
4.3.1. Codificación Reed-Solomon	43



4.3.2.	Codificació Convolutiva.....	43
4.3.3.	Codificador Reed-Solomon + Convolutiva	44
4.4.	BER vs Eb/No modulació 16QAM	45
4.4.1.	Codificació Reed-Solomon	46
4.4.2.	Codificació Convolutiva.....	46
4.4.3.	Codificador Reed-Solomon + Convolutiva	47
4.5.	BER vs Eb/No modulació 64QAM	48
4.5.1.	Codificació Reed-Solomon	48
4.5.2.	Codificació Convolutiva.....	48
4.5.3.	Codificador Reed-Solomon + Convolutiva	49
4.6.	Sincronització	50
5.	Presupost	53
5.1.	Coste mano de obra	53
5.2.	Coste software	53
5.3.	Amortització.....	53
6.	Conclusions y líneas futuras:	54
	Bibliografía:	56
	Glosario	57

Listado de Figuras

Figura 1. Diagrama de Gantt	14
Figura 2. Icono Simulink Library	17
Figura 3. Simulink Library Browser	18
Figura 4. Espacio de trabajo	18
Figura 5. Diagrama de bloques del transceptor	19
Figura 6. Bloque del sistema completo	19
Figura 7. Generador de bits	20
Figura 8. Ilustración de la codificación RS	21
Figura 9. Codificador y decodificador RS	23
Figura 10. Diagrama codificador convolucional de tasa $r=1/2$	23
Figura 11. Codificador y decodificador convolucional	26
Figura 12. Modulador y demodulador BPSK	27
Figura 13. Modulador y demodulador QPSK	27
Figura 14. Modulador y demodulador MQAM	28
Figura 15. Bloques BusCreator y BusSelector	28
Figura 16. Modulador y demodulador reconfigurable	29
Figura 17. Modulador y demodulador OFDM	30
Figura 18. Índices de las sub-portadoras piloto	31
Figura 19. Canal Gaussiano	32
Figura 20. Esquema del canal con eco	33
Figura 21. Estructura de la trama	33
Figura 22. Esquema del bloque sincronizador y estimador	34
Figura 23. Bloques auxiliares	36
Figura 24. Interfaz de usuario	37
Figura 25. Esquema completo del bloque WiMAX-2004 PHY LAYER	38
Figura 26. Aplicación BERTool	39
Figura 27. BERTool opción Monte Carlo	40
Figura 28. Warning generado por la aplicación BERTool	41
Figura 29. BER vs E_b/N_0 modulación BPSK	41
Figura 30. BER vs E_b/N_0 modulación QPSK y codificación RS	43
Figura 31. BER vs E_b/N_0 modulación QPSK y codificación convolucional	43
Figura 32. BER vs E_b/N_0 modulación QPSK y codificación RS + convolucional	44
Figura 33. BER vs E_b/N_0 modulación 16QAM y codificación RS	46

Figura 34. BER vs E_b/N_0 modulaci3n 16QAM y codificaci3n convolucional	46
Figura 35. BER vs E_b/N_0 modulaci3n 16QAM y codificaci3n RS + convolucional	47
Figura 36. BER vs E_b/N_0 modulaci3n 64QAM y codificaci3n RS.....	48
Figura 37. BER vs E_b/N_0 modulaci3n 64QAM y codificaci3n convolucional	49
Figura 38. BER vs E_b/N_0 modulaci3n 64QAM y codificaci3n RS + convolucional	49
Figura 39. Interfaz de usuario y salida del Modulador OFDM para la secuencia pre3mbulo	50
Figura 40. Salida del correlador	51
Figura 41. Interfaz de usuario y salida del Modulador OFDM para la secuencia pre3mbulo	51
Figura 42. Salida del correlador	52

Listado de Tablas:

Tabla 1. Estándares WiMAX y breve descripción	16
Tabla 2. Resumen de la codificación por modulación	21
Tabla 3. Resumen codificación RS	22
Tabla 4. Resumen codificador convolucional	25
Tabla 5. Resumen de la ganancia conseguida	42
Tabla 6. Resumen de la ganancia conseguida	45
Tabla 7. Resumen de la ganancia conseguida	47
Tabla 8. Resumen de la ganancia conseguida	50
Tabla 9. Coste mano de obra	53
Tabla 10. Coste Software	53
Tabla 11. Depreciación.....	53

1. Introducción

Durante los últimos años, tras el éxito que tuvieron las operadoras de internet con la implantación de la tecnología WiFi, rápidamente los fabricantes empezaron a pensar en una tecnología bastante más avanzada y ahí surgió la certificación WiMAX en el año 2001. La tecnología WiMAX está pensada para construir una infraestructura de red cuando el entorno i/o distancia no son favorables para desplegar una red cableada (por ejemplo, zonas rurales de difícil acceso). Es una alternativa de despliegue más rápido y barato que tener que instalar cable o fibra óptica. Aunque, actualmente, su nicho de mercado más importante reside en proporcionar conectividad (voz sobre IP e Internet) a empresas, por ejemplo, con sedes muy separadas entre sí, en los últimos años muchas operadoras se han lanzado a ofrecer este servicio a los consumidores finales, a precios y velocidades muy atractivos.

El estándar WiMAX está diseñado específicamente para:

- Operar en entornos de exteriores, en condiciones tanto de visión directa (LOS - line of sight) como en difracción (NLOS- non line of sight) y con radio enlaces cuya longitud puede llegar hasta las decenas de kilómetros.
- Soportar tasas de transferencia elevadas (aprox. 40Mbps por canal), con una escalabilidad de hasta 1000 usuarios por controlador de acceso y proporcionar mecanismos de calidad de servicio (QoS-Quality of service).
- Puede operar en bandas con y sin licencia.
- Permite proporcionar servicios Triple Play: datos (acceso Internet), voz (llamadas telefónicas) y vídeo (TV, vídeo bajo demanda).

La capa física simulada en este proyecto es la versión WirlesMAN-OFDM PHY, que se puede encontrar en la sección 8.3 del estándar IEEE 802.16-2004. WirelessMAN-OFDM. Dicha versión de la capa física utiliza una metodología de acceso OFDM y está pensada para trabajar a frecuencias portadoras menores a los 11GHz, lo que permite el uso del sistema en entornos NLOS. OFDM (Orthogonal Frequency Division Multiplexing) es un proceso de multiplexado de la información que permite la transmisión de múltiples canales de comunicación a muy alta velocidad, con un nivel de complejidad de implementación reducido. Al simplificar la implementación del receptor, el uso de OFDM reduce el efecto de retardo de la transmisión, permitiendo latencias del servicio del orden de los ms. Para compensar los errores en la información transmitida introducidos por el canal de comunicaciones, el sistema incorpora el uso de codificación de canal (FEC - forward error correction), mediante la concatenación de un código bloque Reed-Solomon y un código convolucional. Esta concatenación de códigos hace que el sistema FEC sea especialmente robusto a los errores introducidos por el canal de comunicaciones, y resulta muy conveniente sobretodo en condiciones de propagación NLOS, como las que debe hacer frente el sistema.

Cabe destacar que para hacer la emulación de la capa física de WiMAX se ha utilizado la herramienta Simulink de Matlab. Esta herramienta posibilita estudiar, evaluar e implementar, de una forma razonablemente simple, las distintas funcionalidades y procedimientos que constituyen la capa física del sistema WiMAX, y constituye la antesala necesaria para realizar un prototipo del sistema mediante el uso de técnicas SDR (Software Defined Radio). En efecto, a partir del modelo Simulink, existen “drivers”

que permiten pasar directamente de este modelo a un prototipo SDR, por ejemplo, utilizando dispositivos USRP (Universal Software Radio Peripheral)¹.

La memoria del proyecto se estructura en 6 capítulos, incluyendo éste introductorio en donde se ha situado el marco referencial del proyecto, se han explicado sus objetivos y se ha detallado el cronograma de las tareas realizadas durante el proyecto y las modificaciones que se han llevado a cabo.

El segundo capítulo se dedica a explicar el contexto del proyecto. En primer lugar, se ha expuesto las razones que motivan el estudio de la tecnología WiMAX frente a otras alternativas como WiFi, para posteriormente realizar una corta descripción de la tecnología SDR, como elemento de prototipado que ha de permitir validar en la realidad el desarrollo Simulink implementado.

El tercer capítulo describe cual ha sido la metodología llevada a cabo para el diseño de los distintos bloques y módulos a dimensionar e implementar, incluyendo el diseño de una interfaz gráfica que permite al usuario introducir los parámetros del sistema que desea considerar para realizar la simulación/emulación.

El cuarto capítulo contiene los resultados de validación de los distintos subsistemas implementados. Primeramente, se presentan los resultados de la tasa de error (BER -bit error rate) resultante bajo diferentes condiciones de relación señal a ruido (E_b/N_0) y considerando distintas modulaciones y tasas de codificación. Para la obtención de la BER se ha utilizado la herramienta BerTool de Matlab. En segundo lugar, se presentan los resultados obtenidos correspondientes a la validación del bloque de sincronización.

El quinto capítulo incluye el detalle del presupuesto del proyecto.

Finalmente, el sexto capítulo contiene las conclusiones y una reflexión sobre las líneas futuras para continuar con el desarrollo del proyecto.

1.1. Objetivos

El propósito de este proyecto es la implementación de la capa física del estándar WiMAX-2004, del estándar IEEE 802.16-2004, usando la herramienta Simulink.

Las especificaciones de la capa física en el estándar WiMAX-2004, están dirigidas para el funcionamiento en las bandas de frecuencia comprendidas entre los 11GHz y los 66GHz para el caso LOS y frecuencias inferiores a 11GHz para el caso NLOS. Está diseñada con un alto grado de flexibilidad con el fin de permitir que los proveedores de servicios tengan la capacidad de optimizar las implementaciones del sistema, con respecto a la planificación de células, el coste, capacidades radio, servicios y capacidad.

Los objetivos principales del proyecto son:

1. Diseño del transmisor radio basado en las especificaciones del estándar WiMAX-2004 mediante el uso de la herramienta de diseño Simulink de Matlab. Se considera únicamente la variante OFDM de las tres consideradas en el estándar.
2. Diseño del receptor radio basado en las especificaciones del estándar WiMAX-2004 (variante OFDM) mediante el uso de la herramienta de diseño Simulink de Matlab.

¹ USRP (Universal Software Radio Peripheral) es un transceptor flexible y asequible que convierte un PC estándar en un potente sistema de prototipado inalámbrico.

Requerimientos del proyecto:

- Programa de software MATLAB/ Simulink.
- Ordenador personal para el uso del software listado.

Especificaciones del proyecto:

- La capa física del WirelessMAN-OFDM está diseñada para trabajar en condiciones de no visibilidad directa (NLOS) en frecuencias inferiores a los 11GHz. En particular operaremos en la banda libre ISM de 2.4GHz.
- Modulación OFDM de 256 puntos.
- Modulaciones BPSK (señalización), QPSK, 16-QAM y 64-QAM por sub-portadora.
- Codificación FEC concatenada con códigos Reed-Solomon y convolucionales.
- Canalización:
 - ❖ 1.25MHz, 5MHz en la banda de 2.4GHz

Modificaciones del plan de trabajo:

En el documento del plan de proyecto, se especificaba que se iba a realizar la integración del software en dispositivos USRPs de ETTUS para crear un demostrador de laboratorio del estándar. Sin embargo, durante la primera fase del trabajo se encontraron distintos problemas, como el dimensionamiento de los codificadores, la caracterización del canal, o la creación de un interfaz de usuario reconfigurable, que ocasionaron un aumento del tiempo planeado para abordar satisfactoriamente estas actividades, y como consecuencia, se decidió en el critical review que no se haría la integración del software desarrollado sobre los dispositivos USRPs.

1.2. Cronograma

En la siguiente figura se muestran las diferentes tareas realizadas durante el transcurso del proyecto. En el Anexo 1 se proporciona información detallada de cada Work Package (WP), tareas e hitos del proyecto.

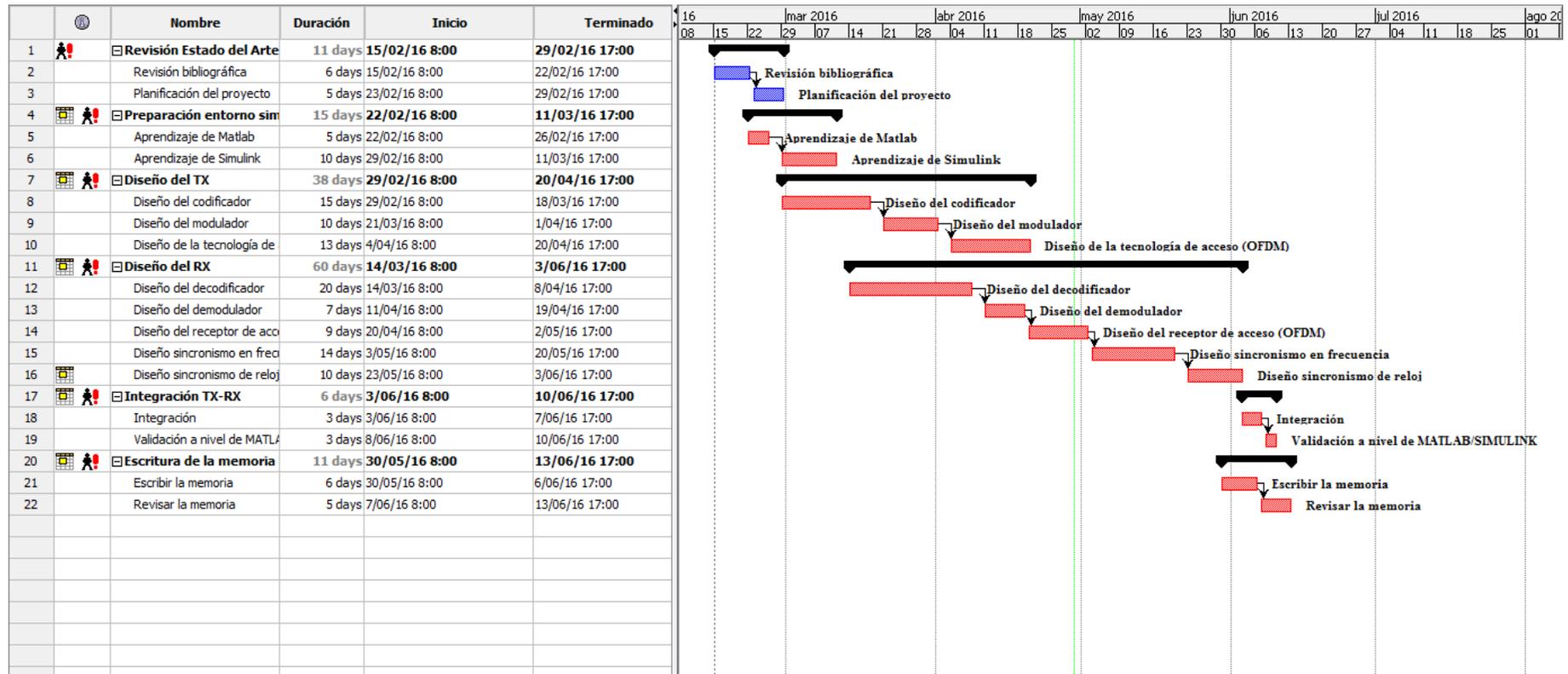


Figura 1. Diagrama de Gantt

2. Marco referencial:

Este capítulo explica el contexto de este proyecto. En primer lugar, se exponen las razones por las que se ha optado por estudiar la tecnología WiMAX frente a otras alternativas como WiFi. Posteriormente se justifica brevemente la razón por la que se ha decidido realizar la implementación de la capa física mediante el uso de tecnologías SDR.

2.1. Sistema WiMAX

A mediados de los años 90, las compañías de telefonía móvil y los proveedores de internet (ISPs-Internet Service Providers), empezaron a trabajar en tecnologías de conexión inalámbrica de banda ancha. Este tipo de tecnología se concibió como una manera de mantener la seguridad y velocidad de una conexión por cable, pero manteniendo los bajos costes de una red inalámbrica. En este contexto, WiMAX es la tecnología radio de banda ancha basada en el estándar IEEE 802.16 y promovida por WiMAX Forum. Este Forum, es una organización no lucrativa que se compone de miembros de alrededor de 512 empresas que comparten el objetivo de promover esta tecnología en el ámbito de los negocios y de los consumidores de todo el mundo.

El estándar IEEE 802.16-2004 se genera a partir de la revisión y consolidación de los estándares anteriores IEEE 802.16-2001, IEEE 802.16a-2003 y 802.16c-2002. En la siguiente tabla podemos ver un resumen de la evolución del estándar 802.16 y sus principales características.

ESTÁNDAR	COMENTARIOS
802.16	Este es el estándar básico publicado en 2001. Creado para proporcionar velocidades de transmisión altas en enlaces de comunicación básicos a frecuencias entre 11 y 60GHz.
802.16a	Esta actualización abordó algunos problemas de eficiencia espectral y permitió el funcionamiento para frecuencias inferiores a los 11GHz.
802.16b	Esta actualización aumentó el espectro que se había especificado para incluir frecuencias de entre 5 y 6 GHz. Se incluyeron mecanismos de calidad de servicio (QoS-Quality of service).
802.16c	El objetivo de la actualización fue crear mayores niveles de interoperabilidad permitiendo operar entre 10 y 66GHz.
802.16d (802.16-2004)	Esta versión, fue una revisión completa de estándar 802.16 que incluyó una serie de mejoras como incluir el uso de 256 sub-portadoras OFDM. También se proporcionan los resultados para las pruebas de cumplimiento y se alineó con el estándar ETSI HiperMAN para su despliegue global. La norma se limitó únicamente al modo de operación estático.
802.16e (802.16-2005)	Pensada para un uso estático y/o móvil. Opera con velocidades de transmisión de datos más bajas, 15Mbps contra los 70 Mbps, con respecto a la versión 802.16d. Permitted el uso del estándar en entornos móviles al incluir mecanismos de handover.
802.16f	Mejora la gestión de la información.
802.16g	Incluye procedimientos y servicios de la capa de gestión.

802.16h	Mejora del mecanismo de convivencia para el funcionamiento del estándar WIMAX en bandas exentas de licencia.
802.16j	Especificación del funcionamiento Multi-hop.
802.16m	Interfaz inalámbrica avanzada. Esta modificación está pensada mirando hacia el futuro ya que permite ofrecer velocidades de datos de 100Mps para aplicaciones móviles y 1Gbps para aplicaciones estáticas. Permite cobertura, macro y micro celular.

Tabla 1. Estándares WiMAX y breve descripción

WiMAX tiene el propósito de democratizar el acceso al Internet de Banda Ancha, es decir permitir el acceso a banda ancha inalámbrica y de alta velocidad a un precio razonablemente bajo, y en cualquier tipo de entorno.

2.2. Software Defined Radio

Los sistemas y servicios de telecomunicación han estado en constante evolución en las últimas décadas. Entre las innovaciones tecnológicas, el uso de las tecnologías digitales resulta en un factor muy relevante. Los sistemas de telecomunicación digitales han demostrado su eficiencia con los años, y han acabado trayendo un nuevo elemento a la cadena de transmisión y recepción, el procesador digital. Este dispositivo ofrece a los nuevos equipos de radio, la flexibilidad de un sistema programable. Hoy en día, el comportamiento de un sistema de telecomunicaciones puede ser modificado por un simple cambio de su software. Esto dio lugar a un nuevo paradigma llamado Software Defined Radio (o Radio definida por software - SDR). En este nuevo paradigma, se desarrollan en software todas las actividades relacionadas con de las funcionalidades básicas del sistema de comunicaciones, dejando para la parte de implementación hardware el diseño y desarrollo de los cabezales de radiofrecuencia (Radio Frequency front-end). Por lo tanto, en este nuevo paradigma la radio ya no es estática, definida por sus circuitos, sino que se convierte en un elemento dinámico, que puede cambiar sus características de funcionamiento, tales como ancho de banda, modulación, tasa de codificación, etc. con solo cambiar la configuración de software; características que incluso pueden ser modificadas durante el tiempo de ejecución del software.

Algunas de las ventajas que podemos destacar del SDR son:

- ❖ Múltiples modos de operación
- ❖ Reconfiguración
- ❖ Actualizaciones del modo de operación en funcionamiento (“Over the Air”)
- ❖ Menor coste de desarrollo

Es justamente esta flexibilidad de reconfiguración y el menor coste de desarrollo, lo que hace atractiva la tecnología SDR para el prototipado, sobretodo de las capas físicas de los distintos estándares de los sistemas inalámbricos y celulares, como es el caso del estándar WIMAX considerado en este proyecto.

3. Metodología:

En este apartado se explica la forma en la que se ha procedido para llevar a cabo la implementación de la capa física del estándar WiMAX-2004 en Simulink de Matlab. Primeramente, ha sido necesaria una lectura del estándar, versión OFDM, para saber qué bloques era necesario implementar y cuál debía ser su estructura y dimensionado. Una vez se ha recogido toda la información necesaria para comenzar a trabajar con Simulink, se han ido creando distintos subsistemas, espacios de trabajo en terminología Simulink, donde se han implementado los diferentes bloques que componen el sistema completo. A continuación, se explica brevemente en que consiste cada uno de los bloques generados.

3.1. Introducción a Simulink

Simulink® es un entorno de diagramas de bloque que permite la simulación multidominio y el diseño basado en modelos. Admite el diseño y la simulación a nivel de sistema, la generación automática de código y la prueba y verificación continuas de los sistemas embebidos.

Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y procedimientos (“solvers”) para modelar y simular sistemas dinámicos. Se integra con MATLAB®, lo que permite incorporar algoritmos de MATLAB en los modelos y exportar los resultados de la simulación a MATLAB para llevar a cabo más análisis.

3.1.1. Creación de un modelo

En primer lugar, será necesario abrir la aplicación MATLAB. Una vez MATLAB se encuentre operativo, podremos hacer click sobre el icono **Simulink Library**, como se muestra en la figura 2.

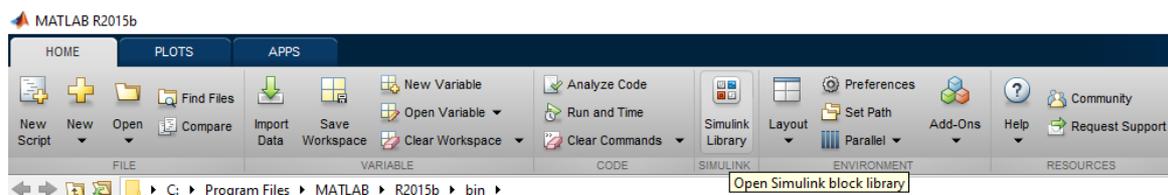


Figura 2. Icono Simulink Library

A continuación, se nos abrirá una ventana como la mostrada en la figura 3, donde tendremos la opción de crear un nuevo modelo.

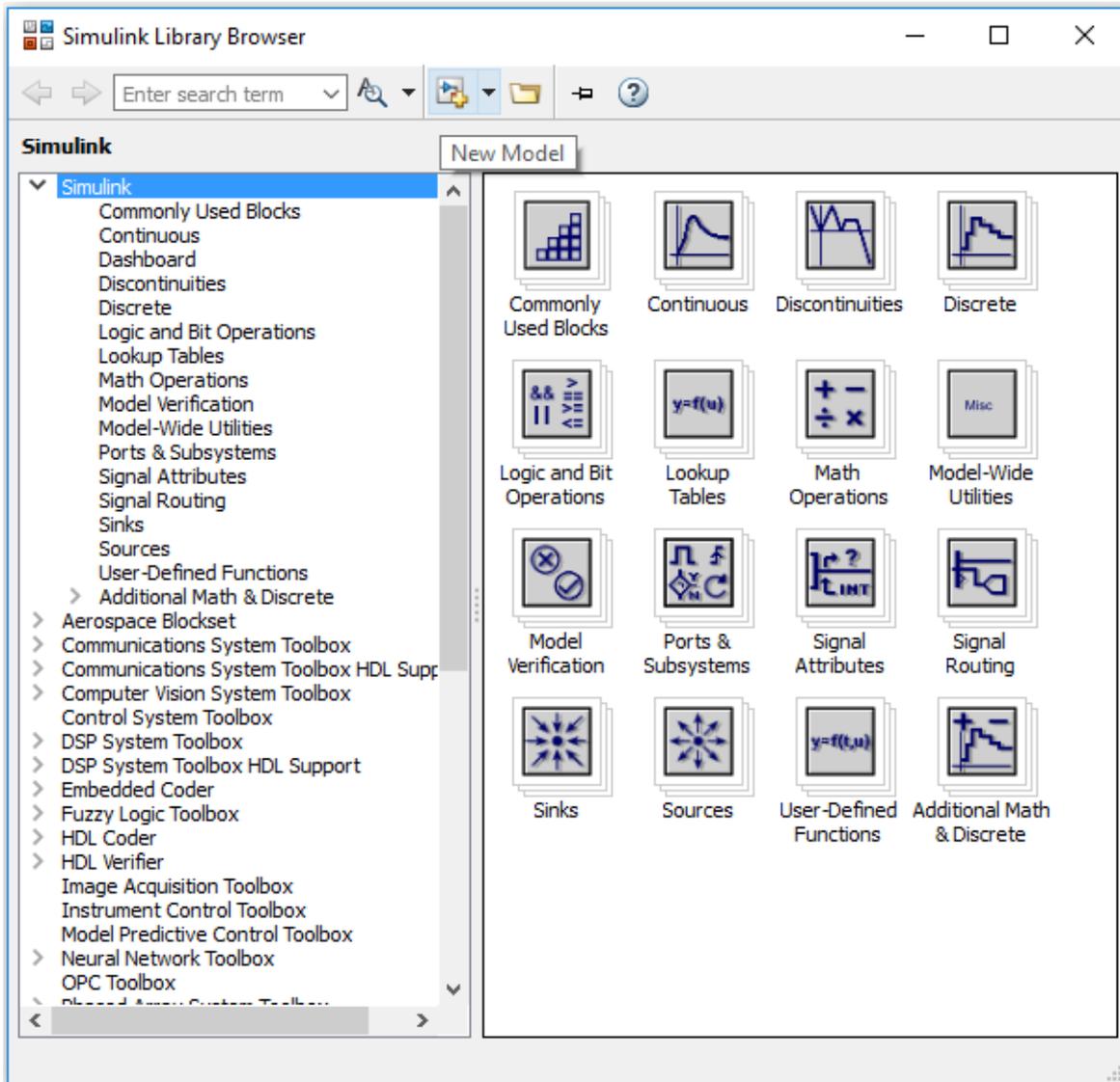


Figura 3. Simulink Library Browser

Al hacer click sobre el icono *New Model*, se nos abrirá el espacio de trabajo (figura 4) donde podremos añadir nuestros bloques, arrastrándolos desde la ventana Simulink Library Browser mostrada anteriormente.

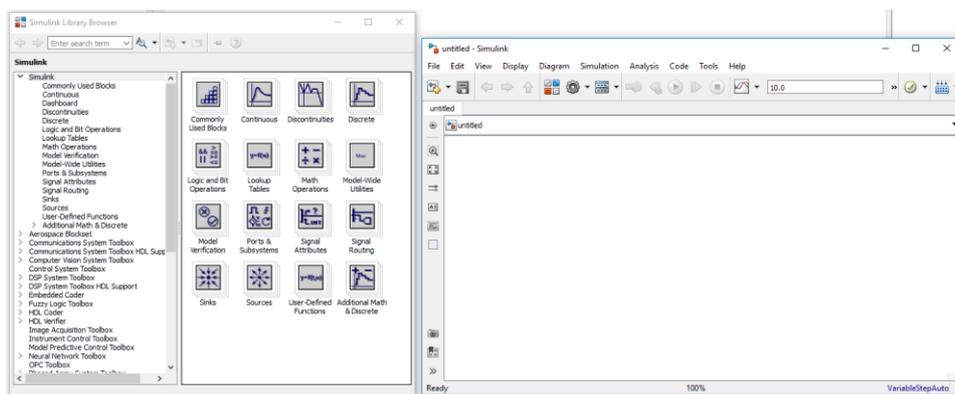


Figura 4. Espacio de trabajo

3.2. Bloques y dimensionado

La figura 5, muestra el diagrama de bloques del transceptor creado en el proyecto. Podemos dividir el diagrama del transceptor en dos bloques, el transmisor y el receptor. El transmisor se compone de un generador de datos, un bloque de codificación de canal, un modulador y un modulador OFDM. El receptor está compuesto por un demodulador OFDM, un demodulador, el decodificador de canal y cualquier dispositivo capaz de leer los datos transmitidos.

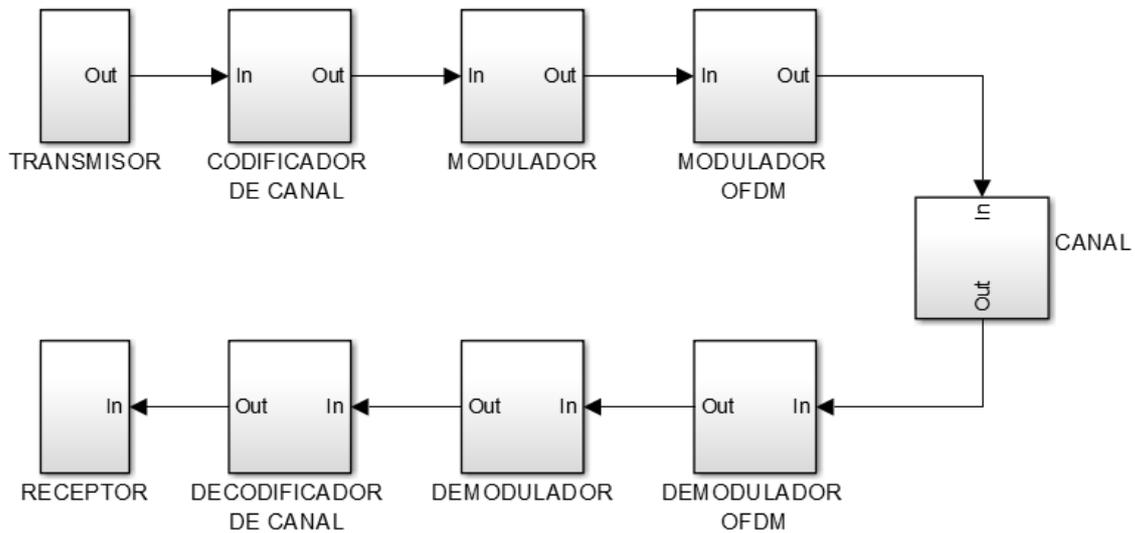


Figura 5. Diagrama de bloques del transceptor

La figura 6 muestra la integración del transceptor de la figura 5 en el software Simulink. En este caso leemos los datos en una pantalla (*display*) que muestra el número de bits transmitidos, el número de bits erróneos y la relación entre bits transmitidos y bits erróneos (BER).

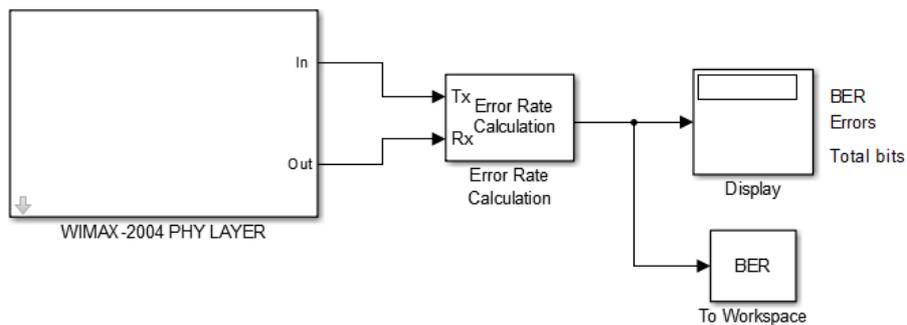


Figura 6. Bloque del sistema completo

3.2.1. Transmisor

Para crear una fuente de información ficticia, se ha usado el bloque “*Bernoulli Binary Generator*”, que se encuentra en la siguiente dirección dentro de la Simulink Library Browser: **Communications System Toolbox/Comm Sources/Random data sources**.

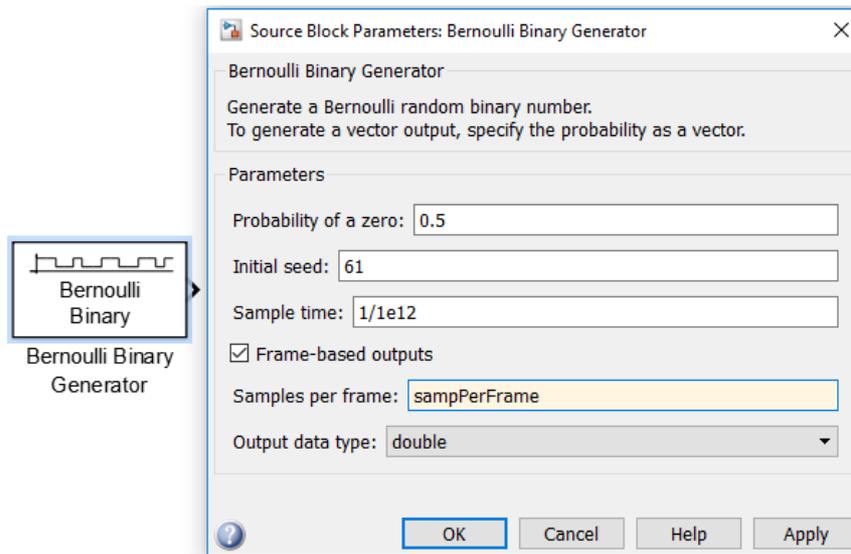


Figura 7. Generador de bits

Este bloque, genera bits aleatorios 1 o 0 con la probabilidad que indiquemos. En nuestro caso, hemos dejado el valor por defecto de 0.5 ya que queremos una fuente con idéntica probabilidad de generación de los bits de información. También nos permite elegir la velocidad de bits por segundo que queremos que genere. En nuestro caso hemos elegido el modo de funcionamiento basado en tramas y hemos definido el número de bits por trama usando una variable que la hemos llamado “*sampPerFrame*”, ya que la dimensión de la trama podrá variar dependiendo de la tasa de codificación que elijamos desde la interfaz de usuario.

3.2.2. Codificador y Decodificador de canal

La codificación de canal, está compuesta por la concatenación de un codificador Reed-Solomon y un codificador convolucional. El estándar WiMAX fija que obligatoriamente se deben implementar los siguientes esquemas de modulación codificación. (Ver tabla 215 del estándar).

Table 215—Mandatory channel coding per modulation

Modulation	Uncoded block size (bytes)	Coded block size (bytes)	Overall coding rate	RS code	CC code rate
BPSK	12	24	1/2	(12,12,0)	1/2
QPSK	24	48	1/2	(32,24,4)	2/3
QPSK	36	48	3/4	(40,36,2)	5/6
16-QAM	48	96	1/2	(64,48,8)	2/3
16-QAM	72	96	3/4	(80,72,4)	5/6
64-QAM	96	144	2/3	(108,96,6)	3/4
64-QAM	108	144	3/4	(120,108,6)	5/6

Tabla 2. Resumen de la codificación por modulación

3.2.2.1. Codificador Reed-Solomon

El código Reed-Solomon de referencia siempre es RS(255,239,16), que significa que a la entrada del codificador necesitamos tener 239 Bytes, y que a la salida se tienen estos mismos 239 Bytes más 16 Bytes de redundancia, de modo que este código puede llegar a corregir hasta 8 Bytes erróneos.

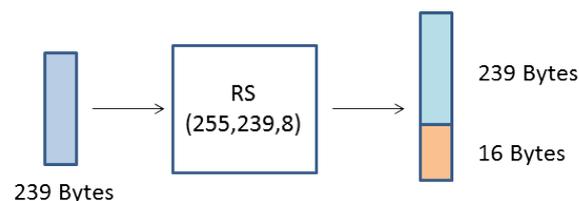


Figura 8. Ilustración de la codificación RS

Cualquier código de la Tabla 215 del estándar, se obtiene a partir del esquema anterior. Por ejemplo, si queremos implementar un código (N,K,T) debemos seguir el siguiente procedimiento:

- Si K es menor que 239 Bytes, rellenamos los Bytes que faltan con ceros hasta llegar a 239, es decir, debemos añadir 239-K Bytes cero antes de los K Bytes de la entrada.
- A la salida del codificador RS(255,239,16) tendremos 255 Bytes, que se corresponde al valor N. Los primeros 239-K Bytes son cero, después tenemos K Bytes iguales a los K Bytes de la entrada y finalmente tenemos 16 Bytes de redundancia.

- c. De los 16 Bytes de redundancia, eliminaremos los 16-N+K primeros Bytes, (equivalentemente 16-2T), y por lo tanto tendremos un código con capacidad de corregir $T = \frac{N-K}{2}$ Bytes.

Por lo tanto, para los códigos indicados en la Tabla 215 del estándar tenemos:

N (Bytes)	K (Bytes)	T (Bytes)	Bytes “ceros” a añadir	Bytes a eliminar de la redundancia
12	12	0	223	16
32	24	4	207	8
40	36	2	199	12
64	48	8	175	0
80	72	4	159	8
108	96	6	131	4
120	108	6	119	4

Tabla 3. Resumen codificación RS

Nota.- El código RS(12,12,0) es equivalente a no tener código, ya que finalmente tenemos los mismos Bytes a la salida del codificador que los que teníamos a la entrada.

Para implementar la codificación y decodificación Reed-Solomon en Simulink, se han usado los bloques “Binary-Input RS Encoder” y “Binary-Output RS Decoder” que se encuentran en la siguiente dirección dentro de la Simulink Library Browser: **Communications System Toolbox/Error Detection and Correction/Block.**

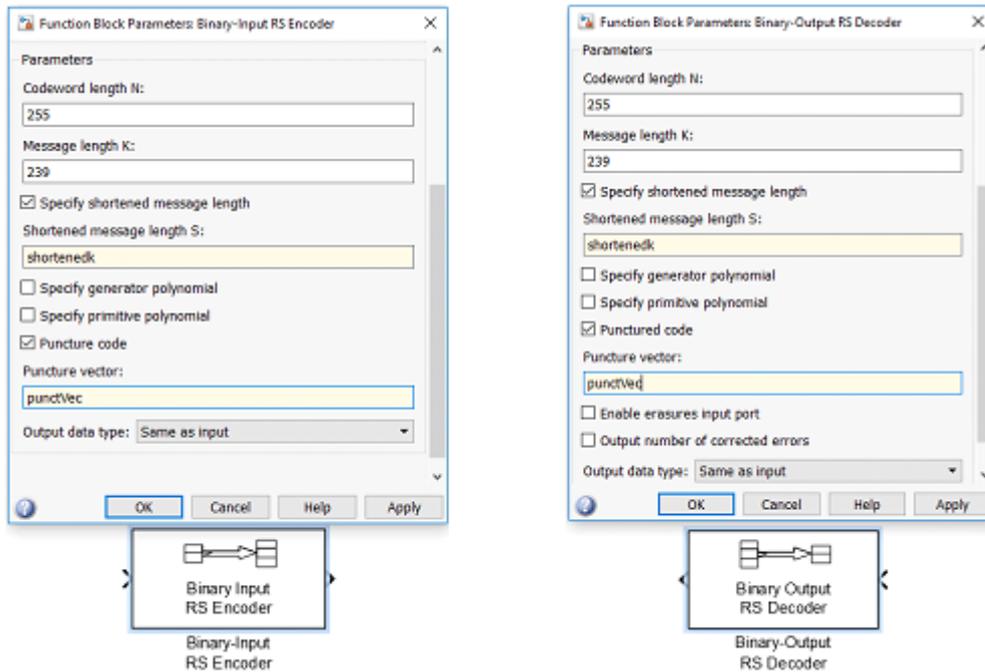


Figura 9. Codificador y decodificador RS

Estos bloques nos permiten usar códigos con longitudes del mensaje acortados de la forma (N,K,T) como los que nos especifica el estándar. Para ello hacemos uso de la opción "Specify shortened message length" y la opción "Puncture code". Los dos parámetros de estas dos opciones, "shortenedk" y "punctVec" están definidos como dos variables debido a que pueden cambiar dependiendo del tipo de codificación y modulación que elijamos desde la interfaz de usuario. Por ejemplo, para una codificación Reed-Solomon (40,36,2), tendremos un valor de $shortenedk = 36$ y $punctVec = [ones(4,1); zeros(12,1)]$.

3.2.2.2. Codificador Convolutivo (CC)

El codificador convolutivo que especifica el estándar, es un codificador de tasa $\frac{1}{2}$ que se genera mediante el siguiente diagrama.

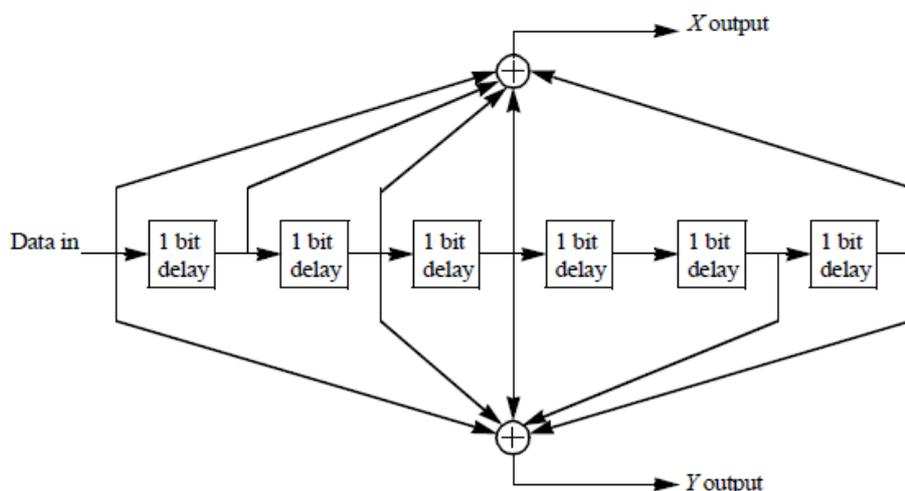


Figura 10. Diagrama codificador convolutivo de tasa $r=1/2$

El polinomio generador de X se caracteriza como: $1111001 \rightarrow (1)_8 (111)_8 (001)_8$, es decir: $G_x=171$ en octal.

Por su parte, el polinomio generador se Y se caracteriza como: $1011011 \rightarrow (1)_8 (011)_8 (011)_8$, es decir: $G_y=133$ en octal.

Además, cualquiera de los dos valores de salida (X) o (Y) depende del bit actual y de los seis bits previos, es decir tiene una memoria ("constraint length" en literatura anglosajona) $K = 7$.

Para adaptar la tasa de codificación a otros valores, la secuencia de bits a la salida del codificador se acorta usando una técnica de diezmado o "puncturing". Así:

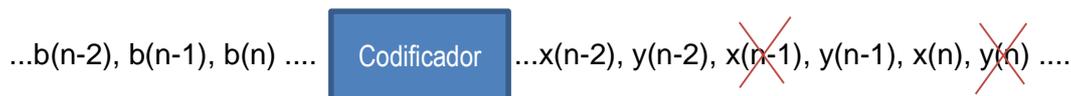
1. Para obtener una tasa de codificación de 2/3 se elimina un bit de cada cuatro de salida.



Por consiguiente, el proceso de diezmado (puncturing) se puede representar como:

- Matriz según $P = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$
- Vector de diezmado (puncturing vector) $P = 1101$

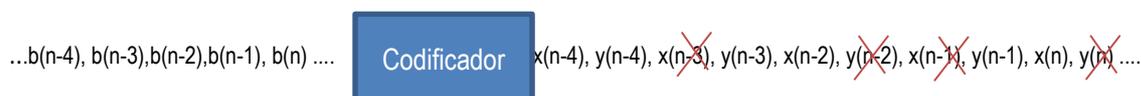
2. Para obtener una tasa de codificación de 3/4 se elimina un bit de cada tres de salida.



En este caso, el proceso de diezmado (puncturing) se puede representar como:

- Matriz según $P = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$
- Vector de diezmado (puncturing vector) $P = 110110$

3. Para obtener una tasa de codificación de 5/6 se eliminan cuatro bits de cada bloque de diez de salida.



Ahora el proceso de diezmado (puncturing) se puede representar como:

- Matriz según $P = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$
- Vector de diezmado (puncturing vector) $P = 1101100110$

En recepción se utilizará un decodificador de Viterbi, y antes de hacer el proceso de decodificación en el lugar de los símbolos borrados se pondrá ruido (el bloque de detección lo hará automáticamente).

- Para el decodificador de tasa 2/3 tendremos:

...x(n-1), y(n-1), **ruido**, y(n)....

- Para el decodificador de tasa 3/4 tendremos:

...x(n-2), y(n-2), **ruido**, y(n-1), x(n), **ruido**....

- Para el decodificador de tasa 5/6 tendremos:

...x(n-4), y(n-4), **ruido**, y(n-3), x(n-2), **ruido**, **ruido**, y(n-1), y(n), **ruido**....

Tabla de codificación por tasa y modulación:

K (Bytes)	N (Bytes)	Tasa (Bytes)	Modulación
12	24	1/2	BPSK
32	48	2/3	QPSK
40	48	5/6	QPSK
64	96	2/3	16QAM
80	96	5/6	16 QAM
108	144	3/4	64 QAM
120	144	5/6	64 QAM

Tabla 4. Resumen codificador convolucional

Para implementar la codificación y decodificación Convolucional en Simulink, se han usado los bloques “*Convolutional Encoder*” y “*Viterbi Decoder*” que se encuentran en la siguiente dirección dentro de la Simulink Library Browser: **Communications System Toolbox/Error Detection and Correction/Convolutional**.

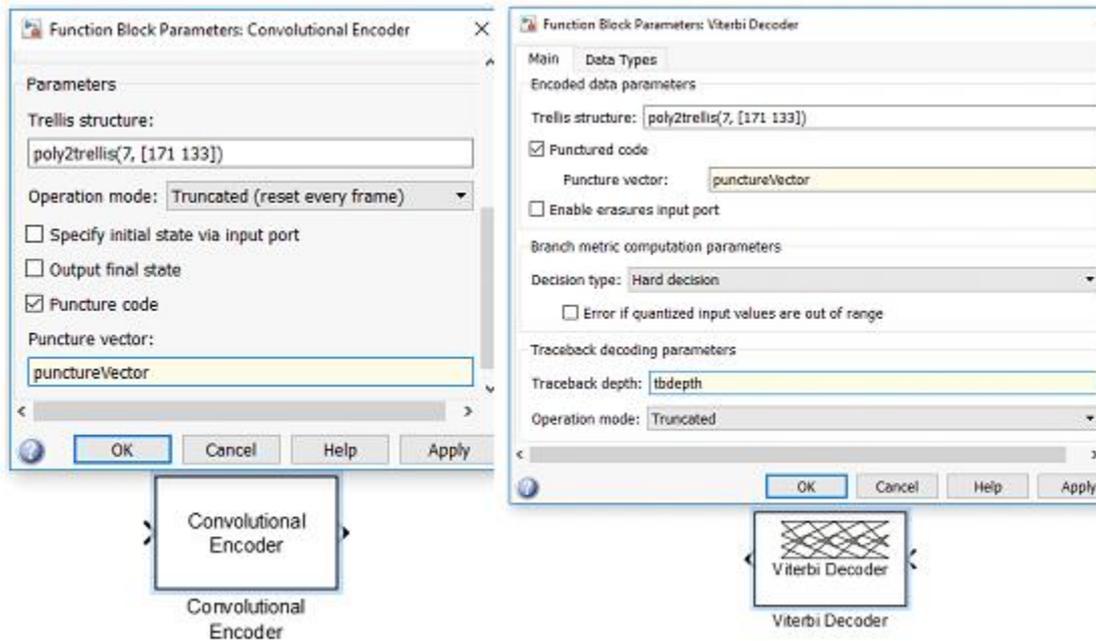


Figura 11. Codificador y decodificador convolucional

Para configurar adecuadamente los bloques, debemos activar el modo “Puncture vector” y el modo de operación tiene que ser “Truncated (reset every frame)”, ya que esto permite borrar la memoria del decodificador trama a trama (recordar que el codificador convolucional tiene una memoria o constraint length de 7 bits). El decodificador de Viterbi (Viterbi Decoder) puede entregar la señal detectada en modo binario (Hard) o cuantificada a un cierto número de niveles. Puesto que la señal a la salida del decodificador de Viterbi debe entregarse al decodificador Reed Solomon, y este trabaja con Bytes (secuencias de 1 y 0), en el dialogo de configuración del bloque “Viterbi Decoder” deberemos activar la opción “Decision Type” en modo “Hard decisión”. Por otra parte, la profundidad de la traza antes de realizar la decisión en el codificador de Viterbi (el parámetro “Traceback depth” del modelo Simulink) viene definido por el estándar según:

Traceback depth:

- Codificación 1/2 → Traceback depth = 5(k – 1).
- Codificación 2/3 → Traceback depth = 7,5(k – 1).
- Codificación 3/4 → Traceback depth = 10(k – 1).
- Codificación 5/6 → Traceback depth = 15(k – 1).

Siendo k igual al “constraint length” del codificador, k=7 en nuestro caso.

Los parámetros “Puncture vector” y “Traceback depth”, se definen como variables debido a que pueden cambiar dependiendo de la modulación y la codificación que se elija desde la interfaz de usuario. Para una codificación, por ejemplo, 16QAM CC(80,96), tendremos un valor de $Puncture\ vector = [1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0]$, y $Traceback\ depth = 15 * (7 - 1)$.

3.2.3. Moduladores

El IEEE Std 802.16-2004 indica que la capa física debe ser capaz de operar con distintos tipos de modulación. Las modulaciones que indica el estándar son las siguientes: BPSK, QPSK, 16QAM y 64QAM. Para implementarlas en Simulink, podemos hacer uso de los bloques ya existentes en la librería de comunicaciones, que los podremos encontrar en la siguiente dirección: **Communications System Toolbox/Modulation/Digital baseband Modulation**.

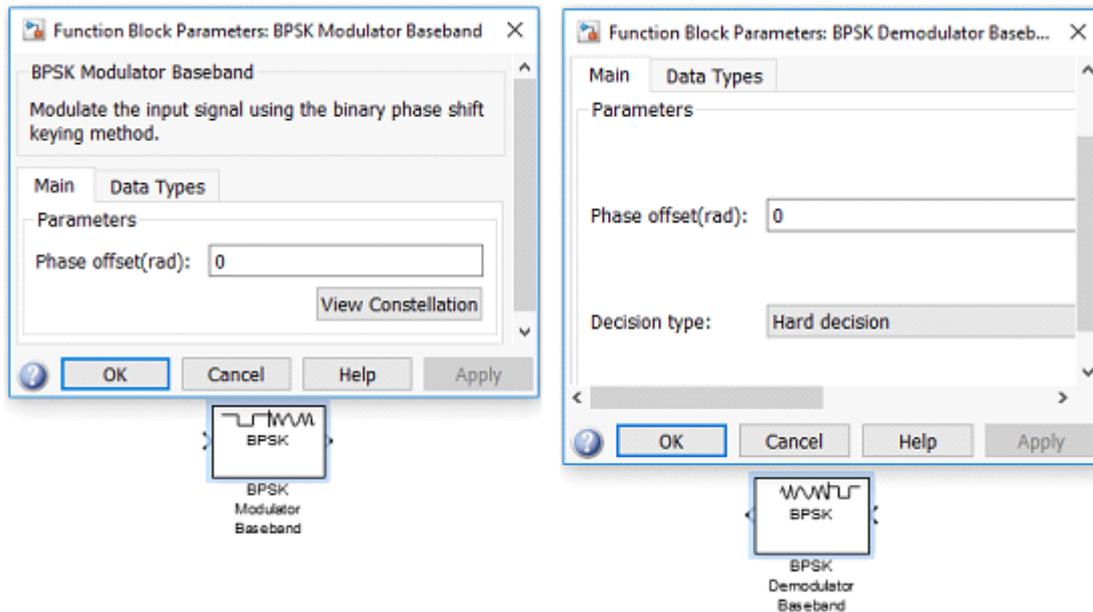


Figura 12. Modulador y demodulador BPSK

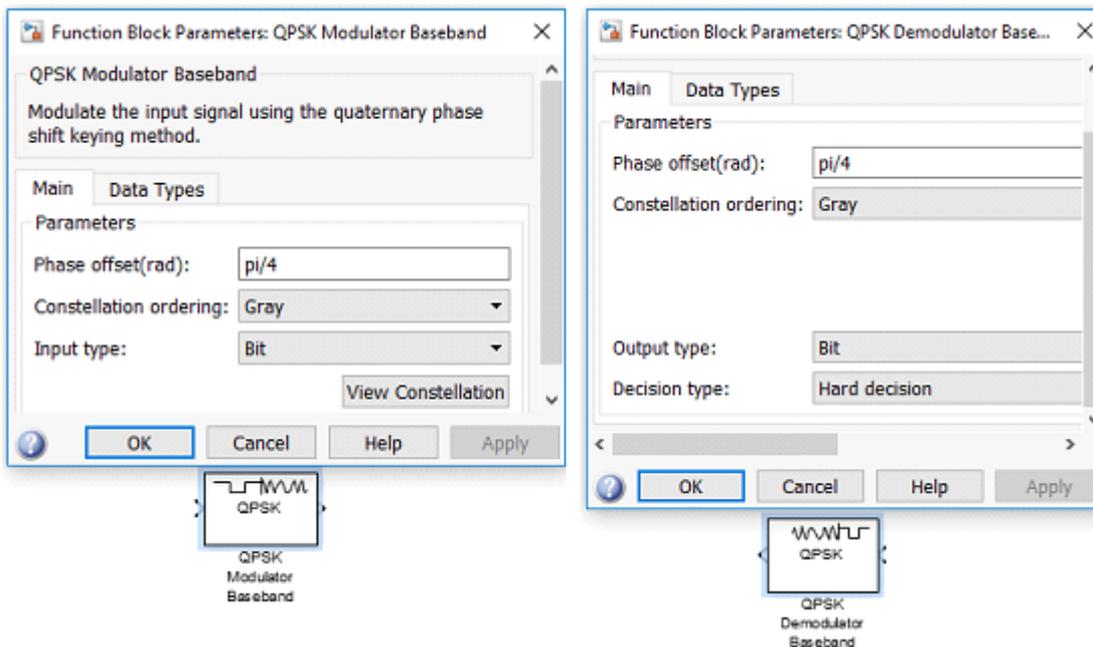


Figura 13. Modulador y demodulador QPSK

Los moduladores BPSK y QPSK son muy sencillos de configurar, por lo que sólo tendremos que elegir entre algunas opciones de las que nos ofrecen los bloques. Debemos asegurarnos de que trabajan con Bits y que en el demodulador la opción

“Decision type” se encuentra como “Hard decision”. Los demás parámetros podemos dejarlos por defecto.

Para las modulaciones 16QAM y 64QAM usamos el bloque “Rectangular QAM Modulator Baseband”. Este bloque nos permite elegir el nivel de modulación deseado.

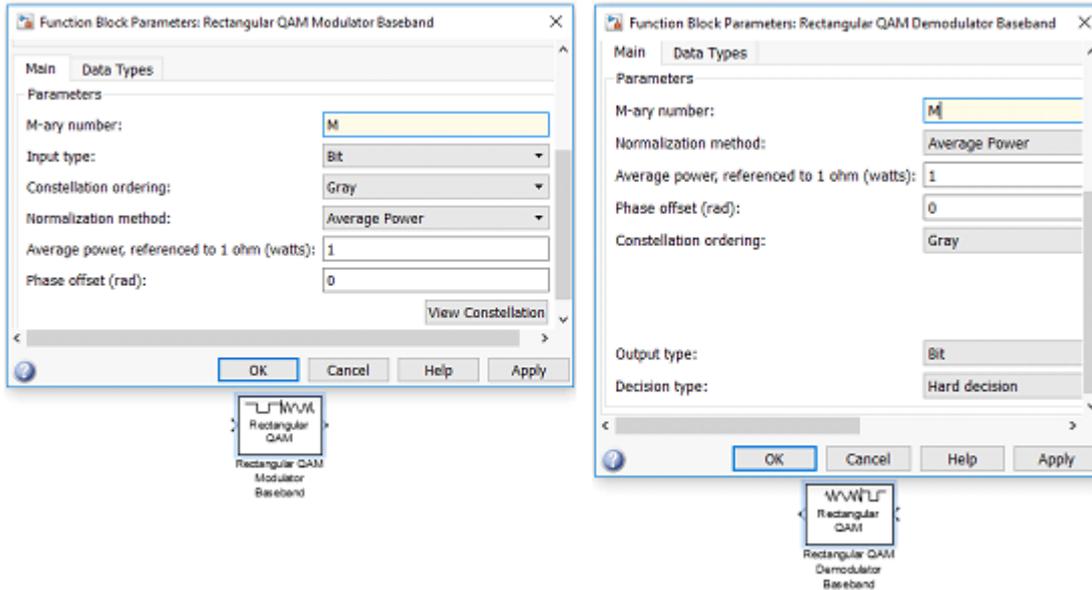


Figura 14. Modulador y demodulador MQAM

Como podemos ver en las figuras anteriores, hemos introducido una variable en el campo “M-ary number”, que nos permite elegir entre una modulación 16QAM o 64QAM desde la interfaz de usuario, dependiendo del valor de “M”.

Simulink no tiene en sus librerías ningún bloque que incluya todas las modulaciones y que nos permita usar la que deseemos para cada simulación, por lo que es necesario incluir todos los bloques de cada modulación y más tarde crear un sistema que nos permita elegir entre todas las modulaciones. Para hacerlo, hemos hecho uso de los bloques “BusCreator” y “BusSelector”. Estos bloques se encuentran en la siguiente dirección del Simulink Library Browser: **Simulink/Signal Routing**.

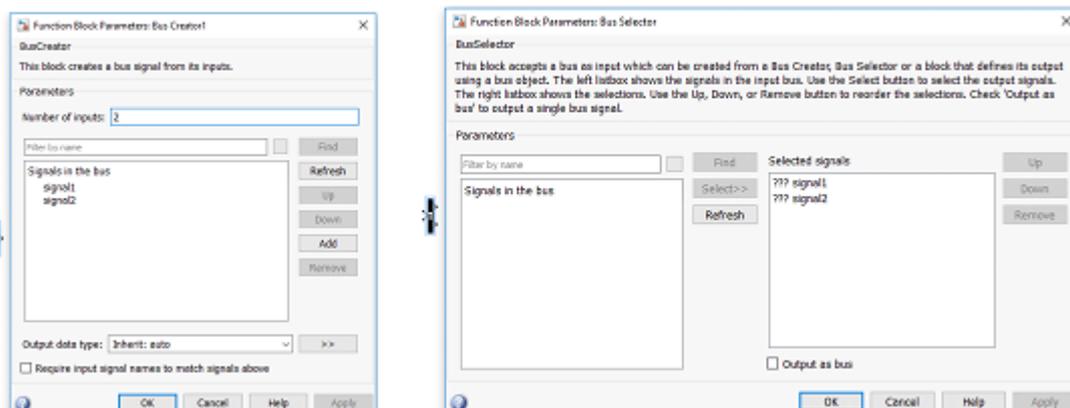


Figura 15. Bloques BusCreator y BusSelector

El bloque “BusCreator” crea un bus de datos de tantas señales de entrada como queramos y el bloque “BusSelector” nos permite elegir cual de todas las señales que hay dentro del bus queremos tener a la salida. De esta forma, hemos usado los bloques para crear nuestro modulador reconfigurable que nos permite cambiar la modulación desde la interfaz de usuario.

La siguiente figura muestra cómo se procedido para crear el modulador/demodulador reconfigurable.

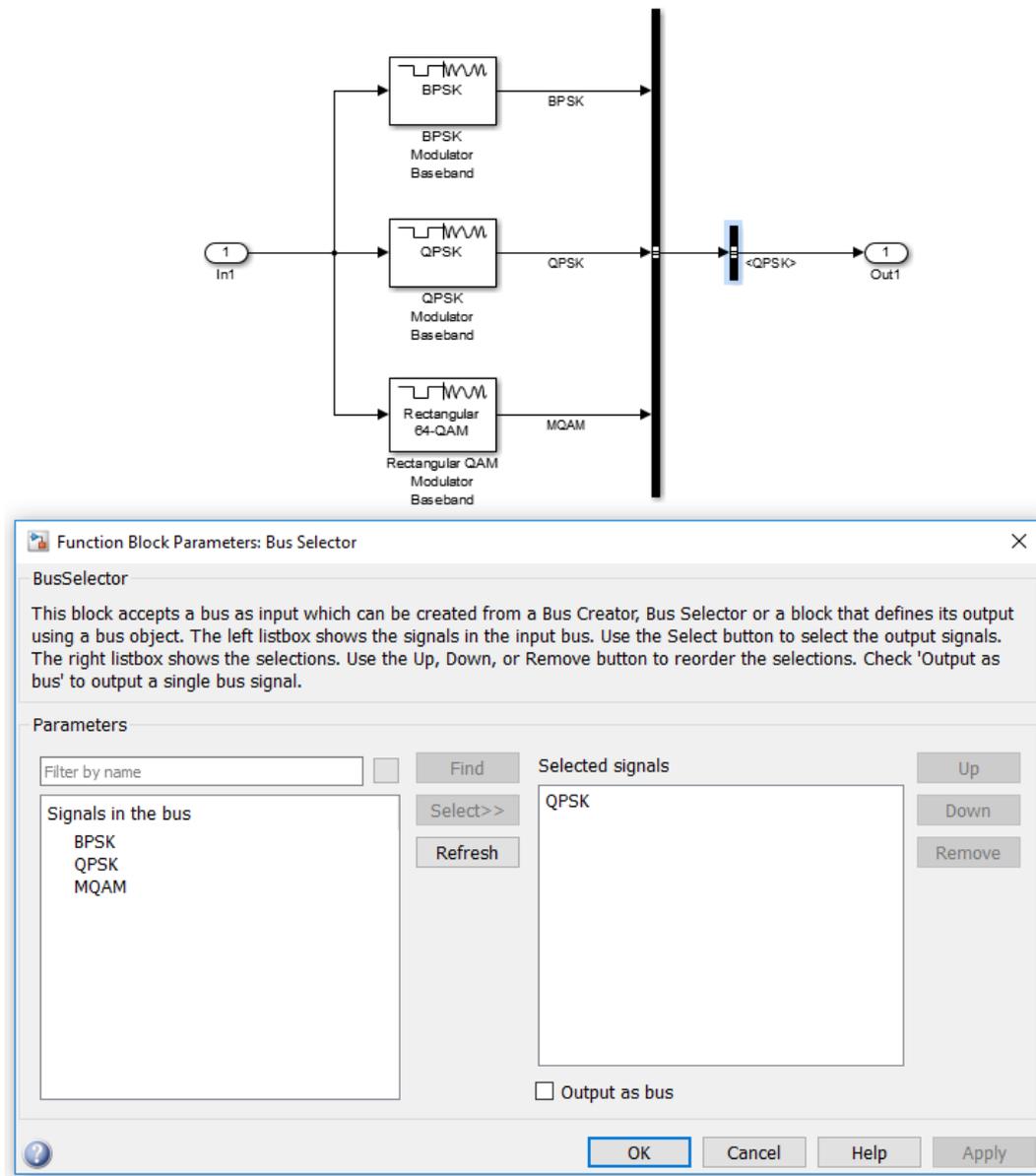


Figura 16. Modulador y demodulador reconfigurable

3.2.4. Moduladores OFDM

Un símbolo OFDM se compone de un conjunto de sub-portadoras, el número de las cuales determina el tamaño de la FFT que se utiliza.

Existen varios tipos de sub-portadora en función del tipo de información que transporten:

- **Sub-portadoras de datos:** Se utiliza para la transmisión de datos de usuario.

- **Sub-portadoras piloto:** Se utiliza para estimar la función de transferencia del canal, y en su caso para ajustar los coeficientes del igualador de canal implementado en el dominio de la frecuencia. Este igualador en el dominio de la frecuencia es muy fácil de implementar ya que básicamente consiste en multiplicar la señal de cada sub-portadora por la inversa de la respuesta en frecuencia del canal en dicha sub-portadora. Esta es una de las ventajas de la transmisión OFDM con respecto a los sistemas, más clásicos, de portadora única.
- **Portadora nula:** No hay transmisión en estas sub-portadoras. Se utiliza para dejar bandas de seguridad (bandas de guarda) entre radiocanales OFDM adyacentes y para marcar el centro del canal (frecuencia portadora o “DC subcarrier” en terminología OFDM).

Seguindo las indicaciones del estándar, hemos generado el modulador y demodulador OFDM con Simulink, a partir de los bloques “OFDM Modulador” y “OFDM Demodulador”, que hemos configurado de forma que su funcionamiento se ajuste al descrito en el estándar. Los bloques los podemos encontrar dentro del Simulink Library Browser en la siguiente dirección: **Communications System Toolbox/Modulation/Digital baseband Modulation/OFDM**.

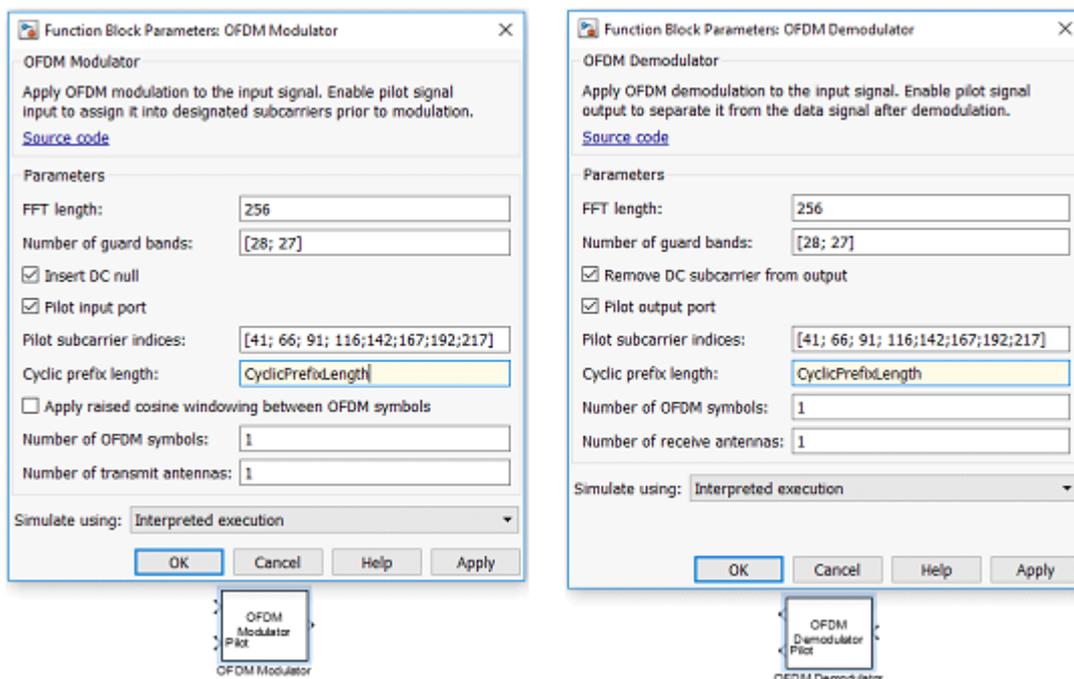


Figura 17. Modulador y demodulador OFDM

Seguindo la tabla 213 del estándar, deberíamos dimensionar los moduladores con los siguientes valores:

- Número de puntos de FFT/IFFT $N_{FFT} = 256$
- Longitud normalizada del prefijo cíclico $G = \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$
- Número de sub-portadoras nulas en la banda de guarda inferior 28
- Número de sub-portadoras nulas en la banda de guarda superior 27
- Posición de las sub-portadoras piloto :

Frequency offset indices of pilot carriers = - 88, - 63, - 38, - 13, 13, 38, 63, 88

Como podemos comprobar en la imagen de la figura 17, algunos de los parámetros no coinciden con los listados anteriormente. El parámetro “Pilot subcarrier indices”, no admite valores negativos en Simulink, ya que las sub-portadoras se numeran entre 1 y 256 mientras que en el estándar están definidas entre -128 y 127. Por esta razón, hemos tenido que trasladar el indexado de las sub-portadoras para que quedaran definidas entre 1 y 256 como muestra la figura 18. Para calcular el valor correspondiente de los índices de cada una de las sub-portadoras piloto, lo único que hay que tener en cuenta es que el índice 0 pasa a ser el 129 y a partir de aquí podemos obtener todos los demás valores. Por ejemplo, la sub-portadora piloto con el índice -88 pasa a tener el índice 41 siguiendo la siguiente expresión:

$$\text{valor índice } 0 + \text{valor índice} = \text{nuevo índice} \quad (129 - 88 = 41).$$

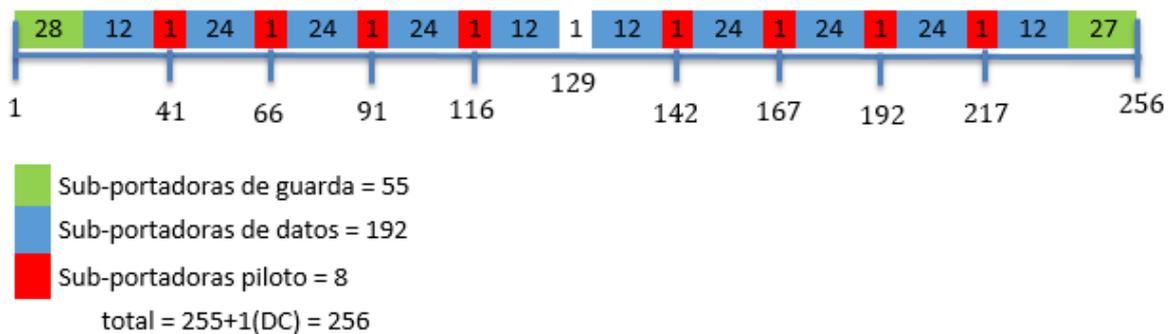


Figura 18. Índices de las sub-portadoras piloto

El otro valor que puede llevar a confusión es el “Cyclic prefix length” ya que el estándar no nos lo especifica directamente, sino que nos da los valores del parámetro G para que lo podamos calcular. Para nuestro caso queda de la siguiente forma:

- Si $G = \frac{1}{4} \rightarrow \text{Cyclic prefix length} = 64$ (equivalente a $256 \cdot G$)
- Si $G = \frac{1}{8} \rightarrow \text{Cyclic prefix length} = 32$ (equivalente a $256 \cdot G$)
- Si $G = \frac{1}{16} \rightarrow \text{Cyclic prefix length} = 16$ (equivalente a $256 \cdot G$)
- Si $G = \frac{1}{32} \rightarrow \text{Cyclic prefix length} = 8$ (equivalente a $256 \cdot G$)

En los bloques de Simulink hemos definido este parámetro como una variable que hemos llamado “CyclicPrefixLength” para poder cambiarla desde la interfaz de usuario y así poder escoger el tamaño del prefijo cíclico que deseemos.

Por otro lado, los parámetros “Number of OFDM symbols” y “Number of receive antennas” podemos, cambiarlos según el sistema que estemos implementando. En nuestro caso los dos valores están fijados a 1 porque no estamos usando diversidad y queremos configurar el modulador OFDM símbolo a símbolo.

Por último, destacar que a pesar de haber definido las sub-portadoras piloto, en nuestro caso, están definidas con un valor de 0 a la entrada ya que no las usamos para estimar el canal, al trabajar fundamentalmente con canales gaussianos y no haber implementado el igualador de canal en el dominio de la frecuencia.

3.2.5. Canal

Para simular el canal de comunicaciones hemos hecho uso del bloque “*AWGN Channel*”. El bloque lo podemos encontrar dentro del Simulink Library Browser en la siguiente dirección: ***Simulink Libary Browswe/Communications System Toolbox/Channels***. Este bloque simula un canal AWGN que nos permite elegir entre algunos modos de funcionamiento y algunos parámetros. Para este proyecto se han usado los modos “*Signal to noise ratio (SNR)*” para el sistema total y el modo “*Signal to noise ratio (Eb/No)*” para realizar las simulaciones con las que se han obtenido los resultados del comportamiento de la codificación de errores del canal, ya que esta es la manera habitual de caracterizar las prestaciones de dichas técnicas de codificación.

Para el sistema genérico hemos definido el parámetro “SNR (dB)” como una variable que podemos cambiar desde la interfaz de usuario.

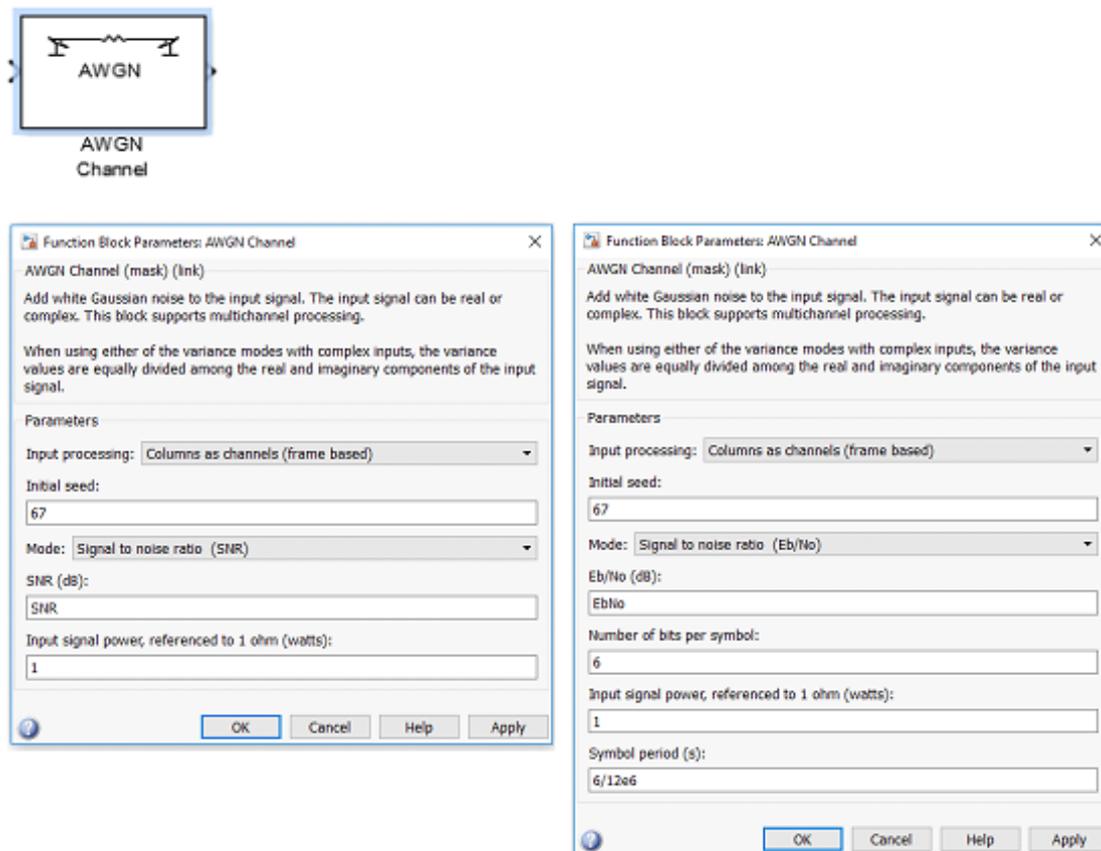


Figura 19. Canal Gaussiano

Por otra parte, para comprobar el correcto funcionamiento del sistema de sincronización temporal implementado, también debemos disponer de un canal que nos permita simular la propagación multicamino. En particular, hemos considerado el caso de tener un camino principal y un eco. Para simular un canal con un eco simplemente hemos retrasado la señal de entrada del canal y le hemos dado una ganancia distinta antes de sumarlo la señal original, justo antes de la entrada del canal gaussiano. En la figura 20 podemos ver la implementación.

- Se puede observar también que cada símbolo contiene un prefijo cíclico generado de la forma convencional.

La correlación de la secuencia transmitida tiene las siguientes propiedades:

$$R_{S_{128}}(\Delta t) = E[S_{128}S_{128}^*] = \int_{S_{128}} S_{128}(t)S_{128}^*(t + \Delta t)dt = \begin{cases} 1 & \text{si } \Delta t = 0 \\ 0 & \text{si } \Delta t \neq 0 \end{cases}$$

A partir de estas propiedades se demuestra que podemos estimar el canal a partir de la secuencia preámbulo. En efecto si denominamos $r(t)$ a la señal recibida (en ausencia de ruido), siendo $r(t) = h(t) * S_{128}(t)$, y calculamos la correlación cruzada entre la señal recibida $r(t)$ y una secuencia preámbulo $S_{128}(t)$, almacenada localmente en el receptor, obtenemos:

$$\begin{aligned} R_{r,S_{128}}(\Delta t) &= \int_{T_{128}} r(t)S_{128}^*(t + \Delta t)dt = \int_{T_{128}} [r(t)S_{128}^*(t + \Delta t)]S_{128}^*(t + \Delta t)dt = \\ &= \iint_{T_{128}} [S_{128}(t)h(t + \Delta l)dl]S_{128}^*(t + \Delta t)dt = \\ &= \int h(t + \Delta l) \left[\int S_{128}(t)S_{128}^*(t + \Delta t)dt \right] dl = \int h(t + \Delta l)\delta(t)dt = \mathbf{h(t)} \end{aligned}$$

De este modo somos capaces de detectar ² la respuesta impulsional (los ecos) del canal.

La implementación en Simulink del algoritmo anterior se ha hecho de la siguiente forma:

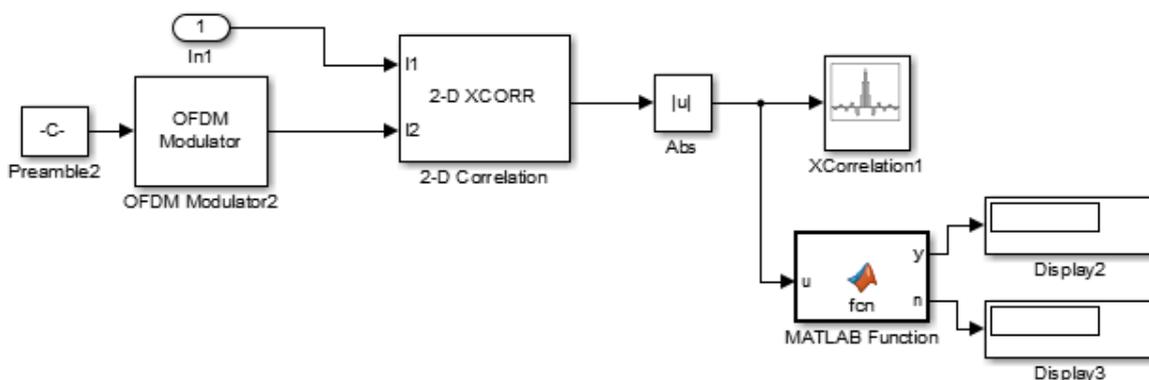


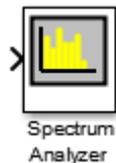
Figura 22. Esquema del bloque sincronizador y estimador

² Realmente obtenemos una estimación de la respuesta impulsional del canal, ya que la señal recibida está afectada por ruido.

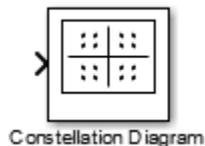
El bloque “Matlab Function” nos permite crear un bloque donde podemos definir su funcionamiento mediante una función escrita en Matlab. En nuestro caso, el bloque devuelve el valor máximo de la correlación y el índice dónde se encuentra. Dada la naturaleza cuasi-ortogonal de las secuencias utilizadas en el preámbulo, sabemos que el máximo de la correlación se producirá cuando las secuencias preámbulo local y recibida se encuentren perfectamente alineadas temporalmente. Por lo tanto, a partir del conocimiento del valor del índice de la secuencia en donde se produce el máximo, podemos dividir la señal recibida en dos partes, una correspondiente a la secuencia preámbulo, que desechamos y otra correspondiente al resto de símbolos transmitidos en la trama³, que sólo contiene información de usuarios, y que nos quedamos para su ulterior procesado.

3.2.7. Otros bloques

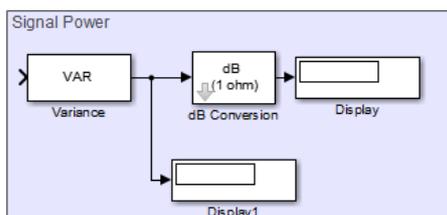
Durante el diseño del sistema, se han usado algunos bloques auxiliares para ayudar a visualizar las características de las señales implicadas, tanto en el dominio de la frecuencia (analizador de espectros), como en el dominio del tiempo, como en el dominio del plano de señal (diagrama de la constelación), además de otros bloques que permiten complementar todas las funciones necesarias a realizar. Algunos de estos bloques son los siguientes:



Permite visualizar el espectro de la señal.

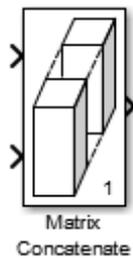


Dibuja el diagrama de la constelación.

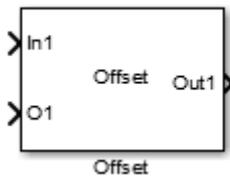


Muestra la potencia de la señal.

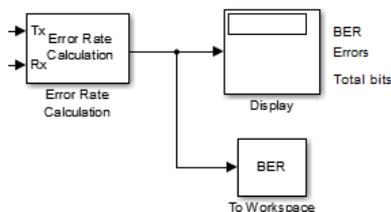
³ En nuestra implementación particular, por simplicidad, sólo hemos considerado tramas con un único símbolo de datos de usuario, pero el procedimiento descrito es absolutamente generalizable a cualquier número de símbolos de datos por trama.



Concatena vectores.



Trunca un vector a partir del valor de Offset que le entra.



Calcula la BER, y nos permite visualizarla, además del número de bits transmitidos y el número de errores totales. El bloque "To Workspace" guarda el valor de la BER en una variable del Workspace de Matlab con el nombre que elijamos.

Figura 23. Bloques auxiliares

3.2.8. Interfaz de Usuario

Con objeto de facilitar el uso del emulador de la capa física del sistema WiMAX, hemos creado "ad-hoc" un interfaz de usuario. Para ello hemos embebido todos los bloques listados anteriormente en un bloque genérico que hemos llamado "WiMAX-2004 PHY LAYER". Para crear este bloque genérico solo hay que seleccionar los bloques deseados, hacer "click" derecho sobre la selección y elegir la opción "Create Subsystem from selection". De esta forma, podemos crear una máscara sobre este nuevo bloque que contiene embebidos a todos los demás. Para crear dicha máscara haremos "click" derecho sobre el subsistema y elegiremos la opción "Mask/Create Mask". La máscara nos permite crear una interfaz de usuario donde podemos definir y programar todas las variables contenidas en los bloques de su interior. En el anexo 2 podemos encontrar el código completo de la máscara.

Una vez programada la máscara, podemos acceder a la interfaz de usuario haciendo doble "click" sobre el bloque en cuestión. En nuestro caso, debemos hacer doble "click" sobre el bloque "WiMAX-2004 PHY LAYER". La figura 24 muestra la interfaz de usuario que hemos creado. Esta interfaz nos permite elegir el tipo de modulación, la tasa de codificación, la longitud del prefijo cíclico, el valor de la relación señal a ruido (SNR) del canal, así como, en el caso de considerar un canal dispersivo de dos ecos, el retardo del segundo eco con respecto al primero y el nivel de este segundo eco, normalizado con respecto al nivel del primer eco.

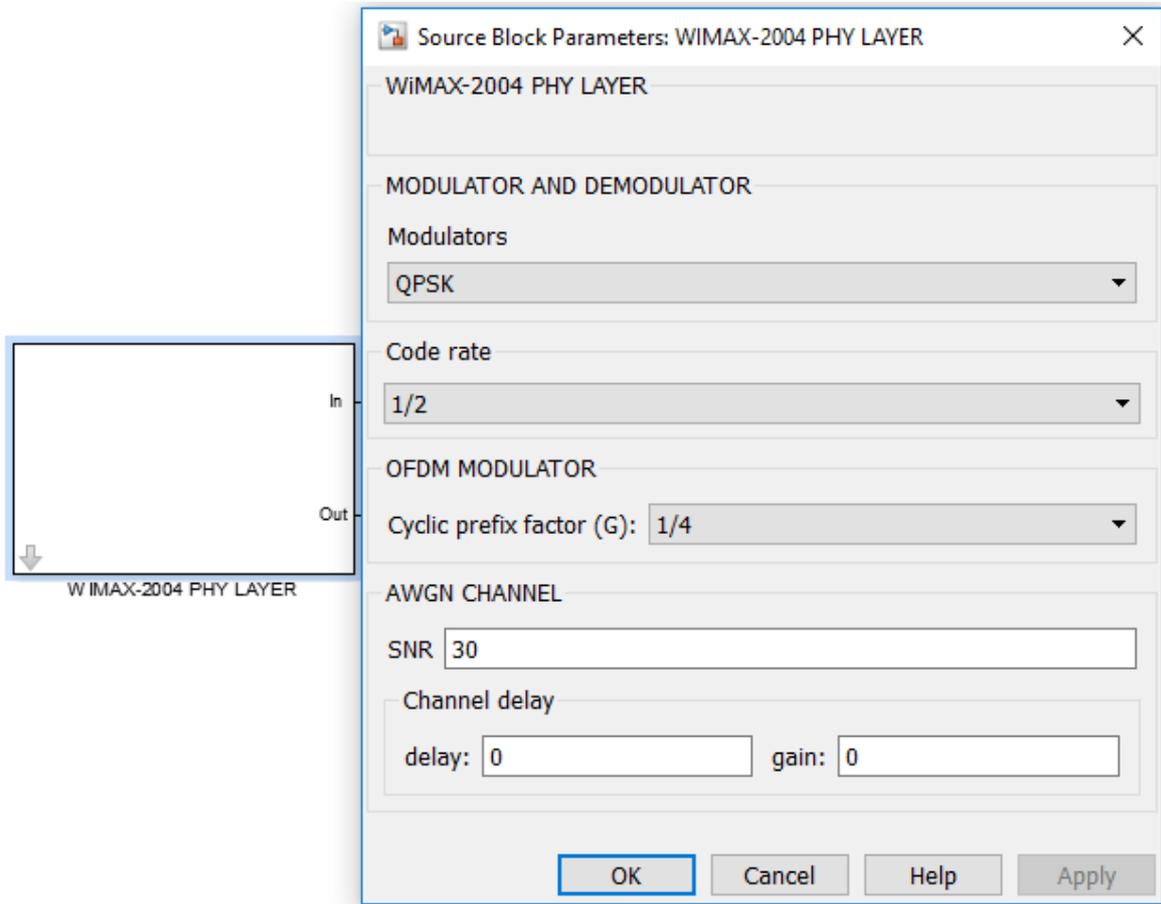


Figura 24. Interfaz de usuario

Si queremos acceder al interior del bloque podemos hacerlo haciendo “click” sobre la flecha que se encuentra abajo a la izquierda del bloque “WiMAX-2004 PHY LAYER”. Esto nos permite ver los bloques que están embebidos, que son los mostrados en el esquema de la figura 25.

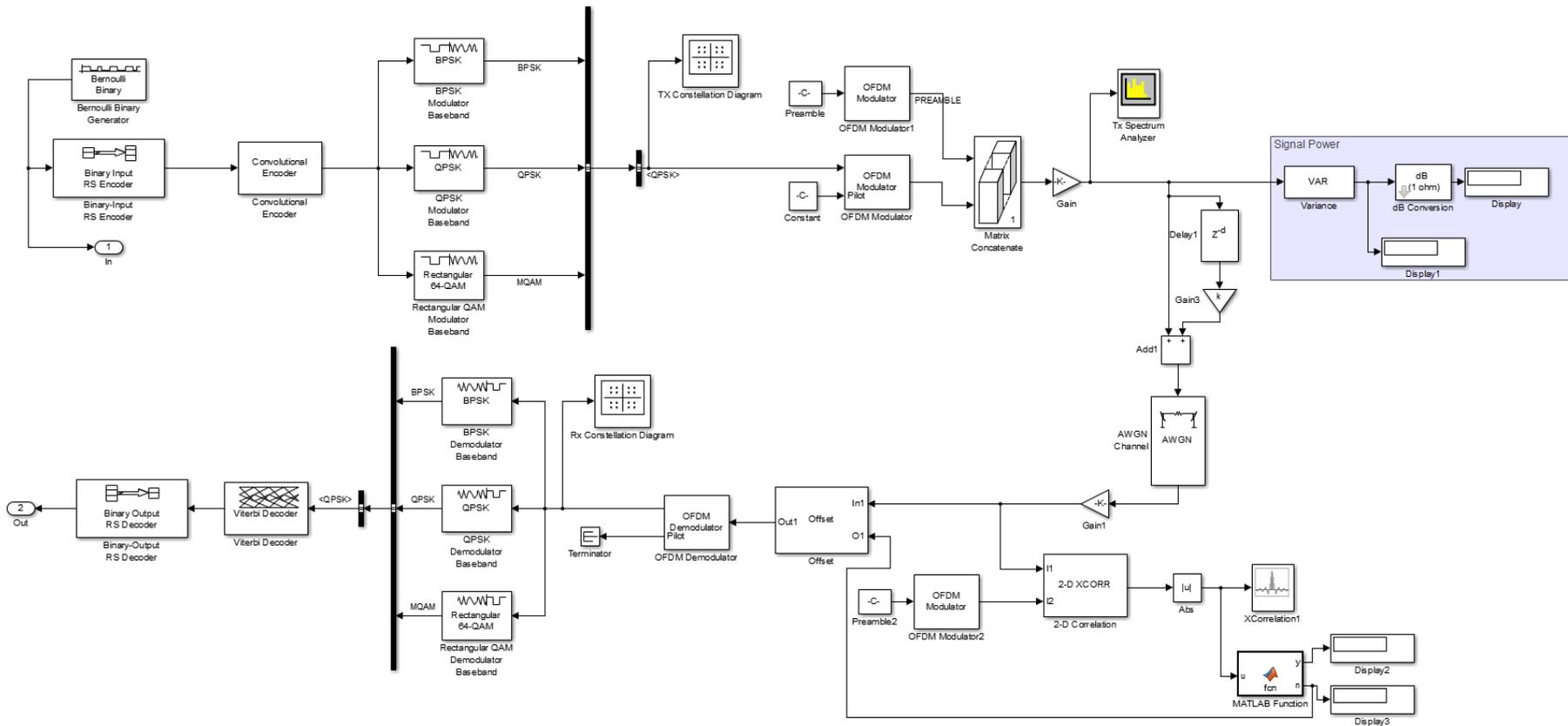


Figura 25. Esquema completo del bloque WiMAX-2004 PHY LAYER

4. Resultados

Con objeto de verificar el correcto funcionamiento de los diferentes bloques que componen el emulador de la capa física del sistema WiMAX, se han realizado una serie de pruebas, cuyos resultados se presentan en este apartado. Para ello, se ha utilizado la herramienta de Matlab “*Bertool*”, que permite realizar una simulación de los diferentes componentes del sistema, y también del sistema global, y obtener resultados gráficos en términos de la tasa de error (BER), en función de un rango de valores de relación señal a ruido (SNR) o equivalentemente del cociente entre la energía por bit y la densidad de potencia de ruido (E_b/N_0) considerada en el modelo.

4.1. Bertool

El comando *bertool* inicia la aplicación Bit Error Rate Analysis Tool (BERTool). La aplicación permite evaluar el comportamiento de la tasa de error de bit (BER) de sistemas de comunicaciones. BERTool no sólo calcula la BER en función de la relación señal-ruido utilizando técnicas de simulación estocástica (simulaciones de Monte-Carlo) mediante SIMULINK sino que también permite comparar los resultados obtenidos con una evaluación semi-analítica mediante funciones de MATLAB y/o con los resultados proporcionados por expresiones en forma cerrada (expresiones analíticas) aplicables a determinados tipos de sistemas de comunicación.

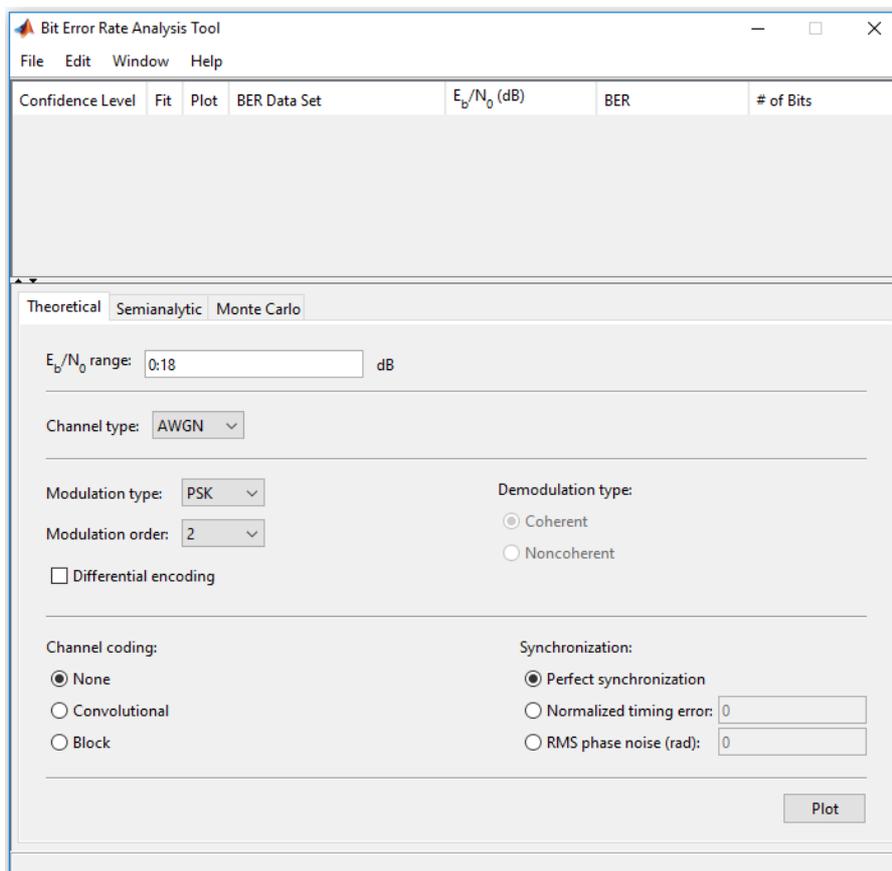


Figura 26. Aplicación BERTool

Al ejecutar el comando *bertool* en el espacio de trabajo de Matlab, se nos abrirá una ventana como la mostrada en la figura 26. Desde esta ventana podremos escoger el tipo de análisis que deseamos hacer: *Theoretical*, *Semianalytic* y *Monte Carlo*. Para realizar el análisis de nuestro sistema hemos usado la opción *Theoretical* para obtener gráficas teóricas del valor de la BER en función del parámetro E_b/N_0 para cada tipo de modulación considerada, y la opción *Monte Carlo* para calcular la BER del modelo desarrollado.

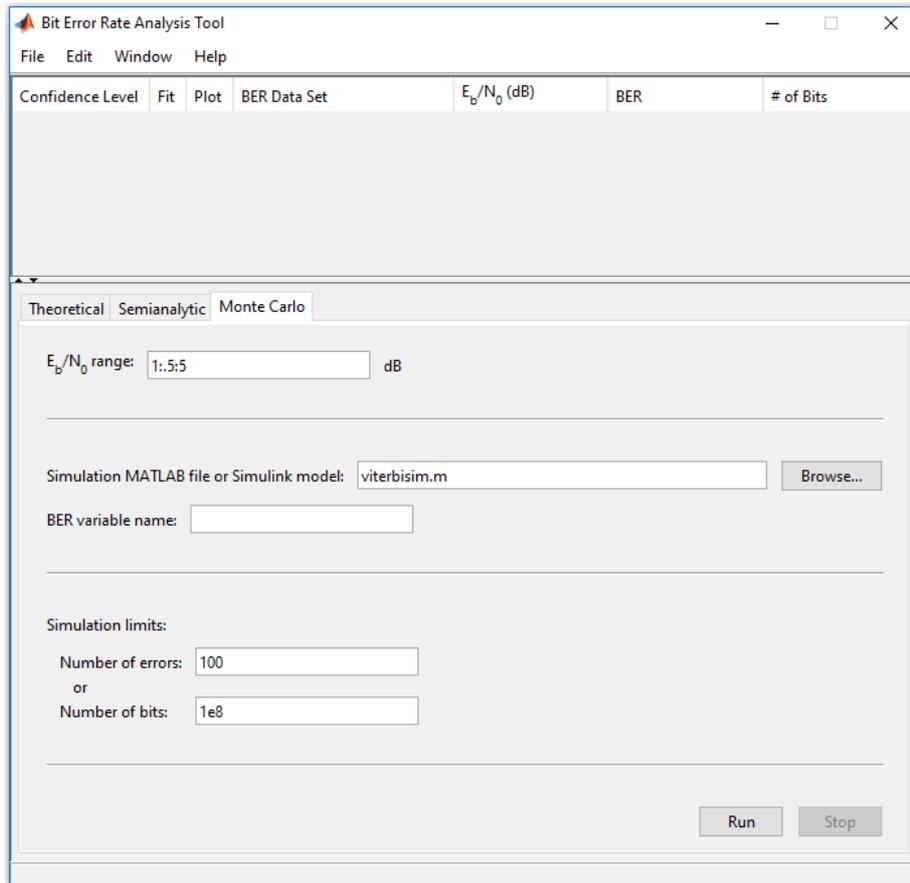


Figura 27. BERTool opción Monte Carlo

Como muestra la figura 27, la opción *Monte Carlo* nos permite seleccionar el modelo Simulink sobre el que queremos realizar el estudio de la BER en función del parámetro E_b/N_0 . En la opción "*BER variable name*" deberemos poner el nombre de la variable que hemos definido para el bloque "*To Workspace*" como el mostrado en la figura 23.

La herramienta BERTool, permite presentar varias simulaciones sobre el mismo gráfico para poder analizar y comparar los resultados. No obstante, en nuestro caso, hemos exportado los resultados obtenidos a una hoja Excel, debido a que la aplicación no es capaz de dibujar líneas de tendencias con variaciones de pendiente muy grandes y aparece un mensaje de aviso ("warning") en el área de trabajo ("Workspace") de Matlab con la siguiente información:

```
Warning: The following warnings are caused by BERTool trying to perform a curve fit.
> In bertool (line 657)
  In bertool (line 577)
Warning: No desirable fit was found. A valid fit must be real valued and monotonically decreasing
between 0 and 0.5 inclusively
> In bertool (line 658)
  In bertool (line 577)
```

Figura 28. Warning generado por la aplicación BERTool

En efecto, el mensaje de aviso nos indica que la aplicación BERTool no es capaz de dibujar la línea de tendencia de los puntos obtenidos en la simulación debido a que la pendiente cambia demasiado rápido.

4.2. BER vs Eb/No modulación BPSK

Con objeto de validar el correcto funcionamiento de los esquemas de modulación codificación implementados, hemos representado la evolución de la tasa de Error (BER) en función del parámetro (Eb/No) para una modulación BPSK con codificación convolucional de tasa 1/2. A efectos de comparación, también se representa la evolución teórica de BER para el caso de un sistema sin codificar.

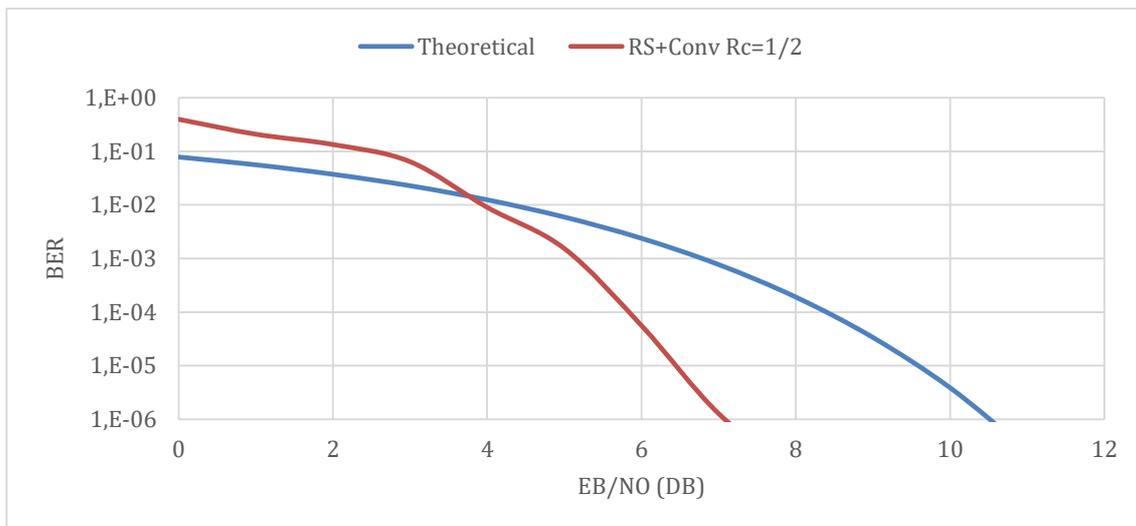


Figura 29. BER vs Eb/No modulación BPSK

Como se puede observar en la figura 29, para valores de Eb/No inferiores a 4dBs aplicando codificación se obtienen valores de BER peores, mientras que para valores superiores la tasa de error decrece muy rápidamente, con una pendiente muy superior comparada con la tendencia que muestra la evolución de la BER en el caso sin codificación (caso teórico). Si observamos los valores obtenidos con la herramienta BERTool, podemos hacer un cálculo de la ganancia aproximada obtenida con la codificación.

Tomando como referencia una BER de $10E-4$, para la modulación BPSK obtenemos una ganancia de 2dB en términos de E_b/N_0 . En la figura 29, también se puede observar como la ganancia de codificación aumenta al incrementar la relación señal a ruido.

Finalmente, constatar que para la modulación BPSK no hemos calculado la evolución de BER con codificación Reed-Solomon (RS) ya que para esta modulación el código RS no corrige ningún error (véase tabla 3 en página 22) y por lo tanto es totalmente transparente

E_b/N_0	Theoretical	RS+Conv Rc=1/2
0	0,0786496	0,392655367
1	0,05628195	0,207165109
2	0,03750613	0,133093525
3	0,02287841	0,06360424
4	0,01250082	0,00895208
5	0,00595387	0,001523305
6	0,00238829	5,57188E-05
7	0,00077267	1,27033E-06
8	0,00019091	1,00E-07
9	3,3627E-05	1,00E-08
10	3,8721E-06	0
11	2,6131E-07	0
12	9,006E-09	0

	E_b/N_0 Performance for $10E-4$ bit error rate	GAIN
Theoretical	8 dB	-
RS+Conv Rc=1/2	6 dB	2 dB

Tabla 5. Resumen de la ganancia conseguida

4.3. BER vs E_b/N_0 modulación QPSK

De forma análoga al caso de la modulación BSK, para la modulación QPSK hemos calculado la BER en función del parámetro E_b/N_0 para tres casos distintos:

- modulación QPSK con codificación Reed-Solomon,
- modulación QPSK con codificación convolucional y
- modulación QPSK con los dos codificadores concatenados.

Para todos y cada uno de los esquemas de modulación y codificación indicados anteriormente, hemos comparado la evolución de la BER obtenida con la evolución prevista para una modulación QPSK teórica sin codificación de canal.

4.3.1. Codificación Reed-Solomon

En la siguiente grafica podemos observar los resultados en el caso de una modulación QPSK con codificación Reed-Solomon, considerando las dos tasas de codificación previstas en el estándar.

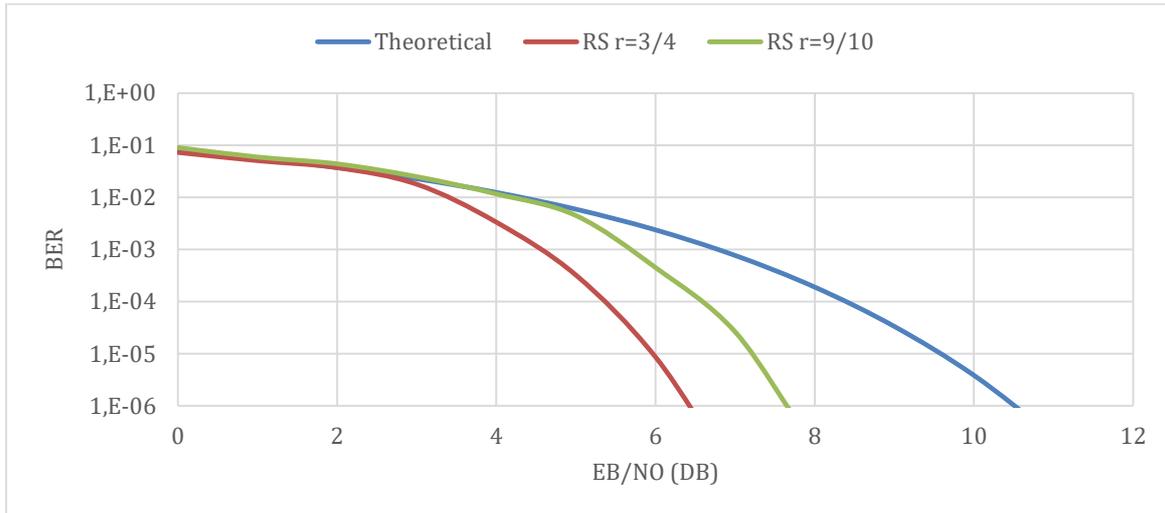


Figura 30. BER vs Eb/No modulación QPSK y codificación RS

En la figura 30, observamos que la codificación Reed-Solomon mejora substancialmente la tasa de error del sistema en el caso de valores de Eb/No razonablemente grandes. Para valores bajos de Eb/No los valores de BER se mantienen igual a los que proporciona una modulación QPSK teórica. Sin embargo, para valores de Eb/No superiores entre 3 y 5 dBs, la tasa de error del sistema codificado disminuye bastante más rápidamente que en el caso del sistema sin codificación (simulación teórica), independientemente de la tasa de codificación. Por otro lado, vemos que la tasa de error en el caso de codificación Reed-Solomon con tasa $r = 3/4$ empieza a decrecer antes que en la codificación con tasa $r = 9/10$. Esto es debido a que con una tasa $r = 3/4$ corregimos 8 Bytes frente a los 4 que corrige la tasa $r = 9/10$.

4.3.2. Codificación Convolutional

En la siguiente gráfica de la figura 31 podemos observar los resultados para el caso de una modulación QPSK con codificación Convolutional considerando, también, las dos tasas de codificación previstas en el estándar.

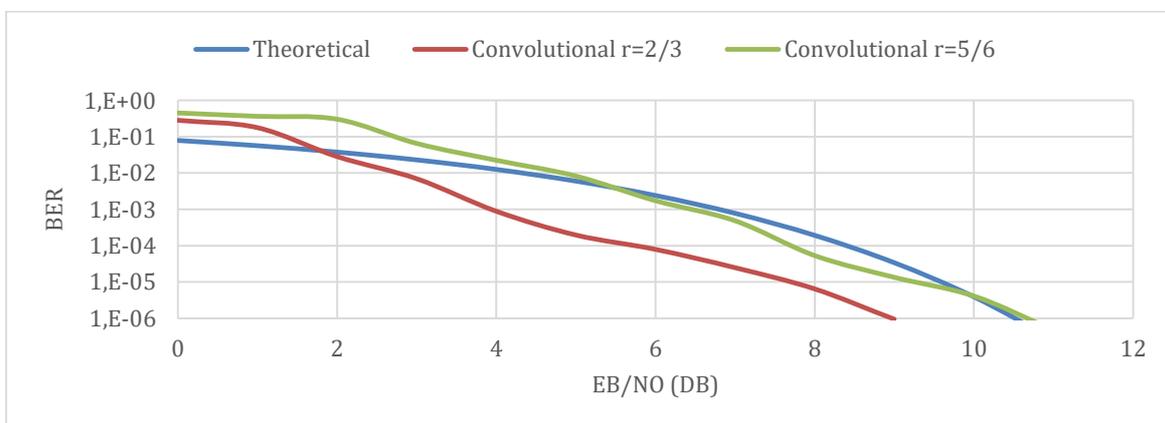


Figura 31. BER vs Eb/No modulación QPSK y codificación convolutional

En la figura 31, podemos observar que, a diferencia de la codificación Reed-Solomon, para valores bajos de relación señal a ruido la tasa de error con codificación convolucional sí que empeora con respecto a un sistema sin codificar. No obstante, a partir de un valor de E_b/N_0 de 2dB, para la codificación de tasa $r = 2/3$, la tasa de error obtenida ya es inferior a la correspondiente para el caso de la simulación teórica de un sistema sin codificar. Por el contrario, para un sistema con codificación convolucional de tasa $r = 5/6$, observamos que, si bien existe una cierta mejora con respecto al sistema sin codificar a partir de un valor de E_b/N_0 de unos 6 dBs, la mejora obtenida es mínima y la línea de tendencia obtenida no suficientemente definida, sobre todo para valores de E_b/N_0 superiores a 9 dBs. Probablemente esto es debido a la limitación del número de bits transmitidos durante la simulación de Montecarlo, con objeto de no alargar en exceso el tiempo de simulación, que limita el número de errores contabilizados y por lo tanto la calidad de la estimación del valor de BER para estos casos.

4.3.3. Codificador Reed-Solomon + Convolucional

En la siguiente gráfica podemos observar los resultados obtenidos para el caso de una modulación QPSK con codificación Reed-Solomon concatenada con codificación convolucional.

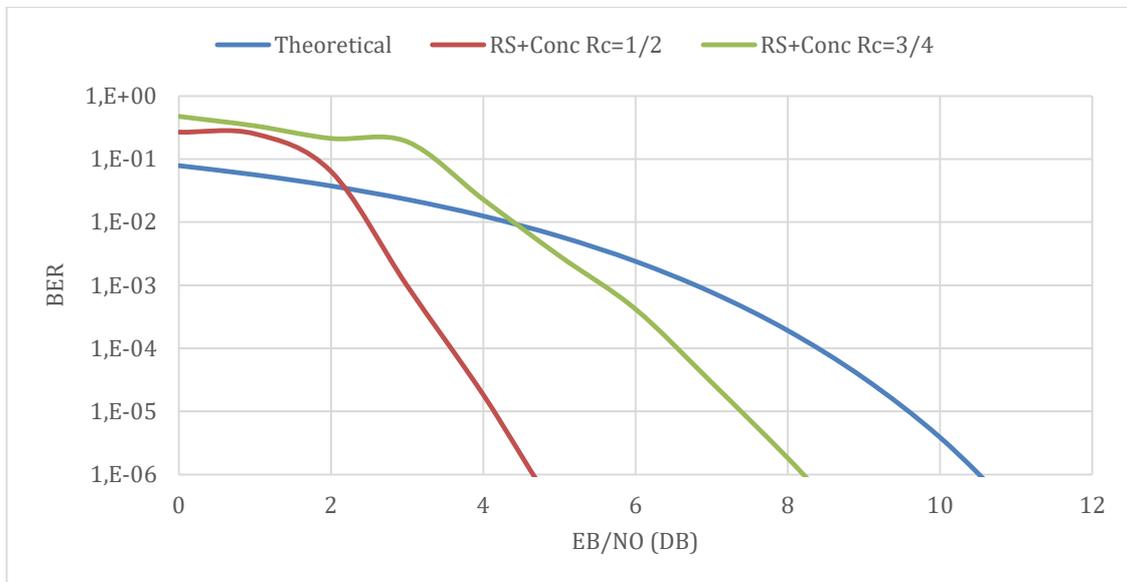


Figura 32. BER vs E_b/N_0 modulación QPSK y codificación RS + convolucional

La figura 32, muestra cómo, si bien para relaciones señal a ruido bajas al concatenar ambos codificadores obtenemos valores altos de BER, sin embargo, podemos comprobar que para valores de E_b/N_0 razonablemente bajos, del orden de entre 2 y 4,5 dBs, la mejora aportada por la codificación es muy relevante.

Como en el caso de la modulación BPSK, si observamos la evolución de los valores de tasa de error, obtenidos con la herramienta BERTool, en función de la relación señal a ruido (equivalentemente E_b/N_0) podemos hacer un cálculo de la ganancia aproximada obtenida con la codificación.

	Eb/No Performance for 10E-4 bit error rate	GAIN
Theoretical	8 dB	-
RS r=3/4	5 dB	3 dB
RS r=9/10	7 dB	2 dB
Convolutional r=2/3	6 dB	2 dB
Convolutional r=5/6	7 dB	1 dB
RS+Conc Rc=1/2	4 dB	4 dB
RS+Conc Rc=3/4	6 dB	2 dB

Tabla 6. Resumen de la ganancia conseguida

Tomando como referencia una BER de $10E-4$, para la modulación QPSK con una concatenación de códigos RS y convolucional y una tasa de codificación global $R_c = 1/2$, se obtiene una ganancia de 4dB en términos de E_b/N_0 . Obviamente como en el caso de modulación BPSK la ganancia aumenta al incrementar la relación señal a ruido tal como muestran las líneas de tendencia de la figura 32.

4.4. BER vs Eb/No modulación 16QAM

Como en el caso de la modulación QPSK, para la modulación 16QAM también hemos evaluado la evolución de la BER en función del parámetro E_b/N_0 para tres casos distintos:

- modulación 16QAM con codificación Reed-Solomon,
- modulación 16QAM con codificación convolucional y
- modulación 16QAM con los dos codificadores concatenados.

Para todos y cada uno de los esquemas de modulación y codificación indicados anteriormente hemos comparado la evolución de la BER obtenida con la evolución prevista para una modulación 16QAM teórica sin codificación de canal.

4.4.1. Codificación Reed-Solomon

En la siguiente gráfica podemos observar los resultados en el caso de una modulación 16QAM con codificación Reed-Solomon, considerando las dos tasas de codificación previstas en el estándar para esta modulación.

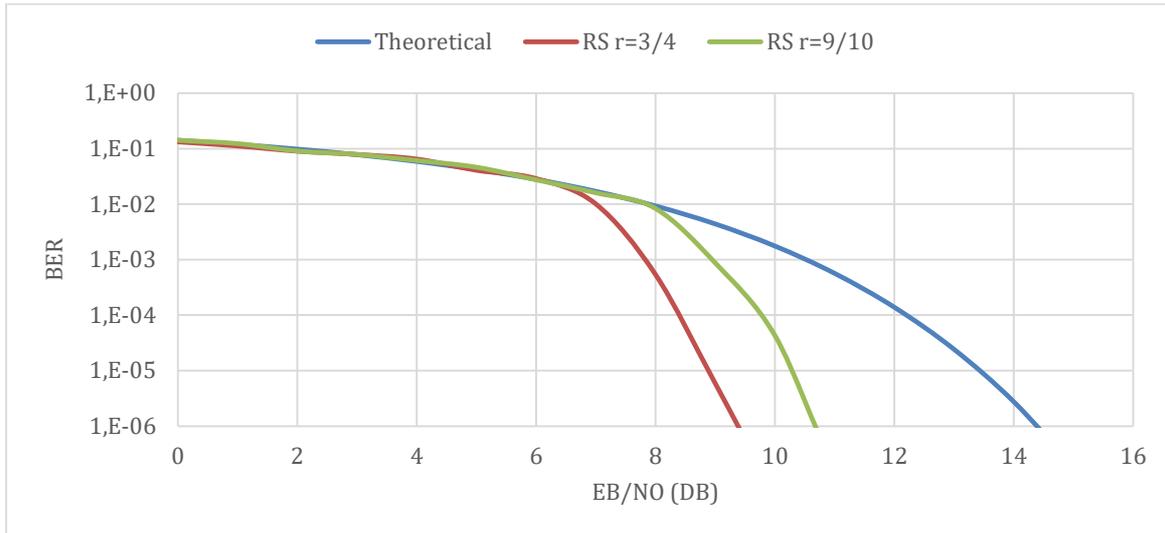


Figura 33. BER vs Eb/No modulación 16QAM y codificación RS

En la figura 33, observamos que los resultados son similares a los que obtenemos en una modulación QPSK con codificación Reed-Solomon, salvo que al estar utilizando una modulación con mayor eficiencia espectral (4 bits por símbolo) necesitamos una mayor relación señal a ruido para obtener resultados equivalentes en términos de BER.

4.4.2. Codificación Convolutiva

En la siguiente gráfica podemos observar los resultados de la evolución de la BER en función de la relación Eb/No considerando una modulación 16QAM con codificación Convolutiva.

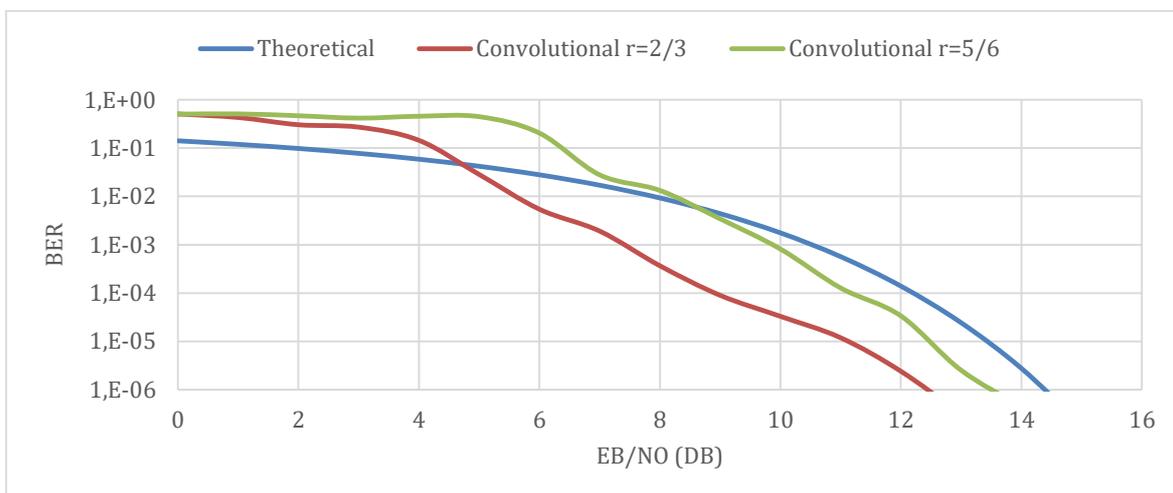


Figura 34. BER vs Eb/No modulación 16QAM y codificación convolutiva

Como en el caso de los codificadores Reed-Solomon, en la figura 34 podemos observar que los resultados obtenidos son muy similares a los equivalentes para una modulación QPSK. En el caso de considerar una tasa de codificación $r = 2/3$ la mejora de la BER es

bastante apreciable, mientras que cuando consideramos una tasa de codificación $r = 5/6$ la mejora aportada por la codificación es marginal.

4.4.3. Codificador Reed-Solomon + Convulacional

En la siguiente gráfica podemos observar los resultados obtenidos para una modulación 16QAM y codificación Reed-Solomon más convulacional concatenada, considerando las dos tasas de codificación indicadas en el estándar. La evolución de la tasa de error se compara con la obtenida para un sistema sin codificar.

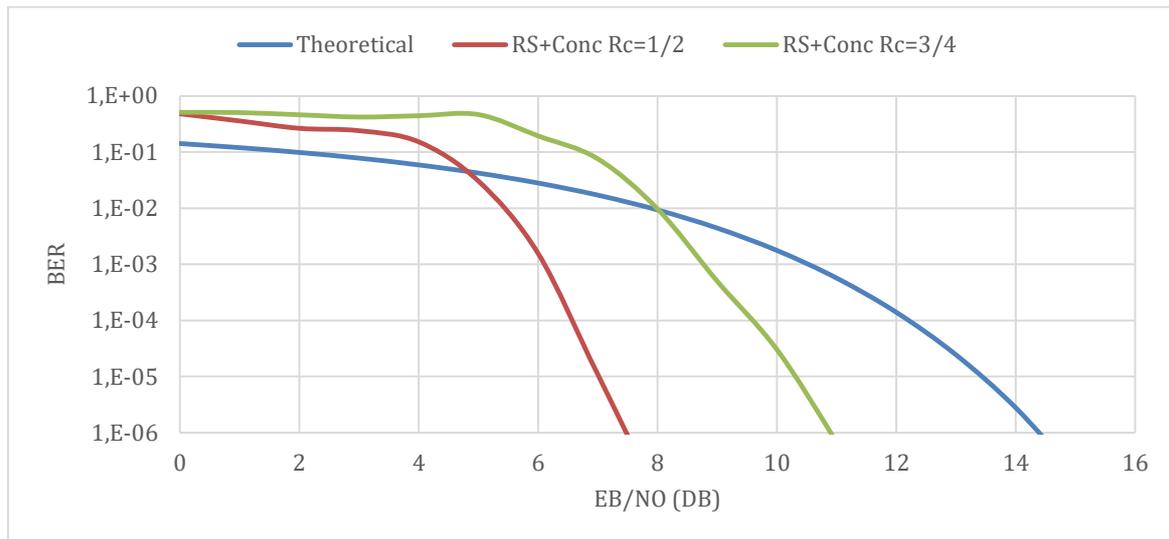


Figura 35. BER vs Eb/No modulación 16QAM y codificación RS + convulacional

En la figura 35 podemos ver como al concatenar los dos codificadores conseguimos corregir gran cantidad de errores para valores de Eb/No superiores 5 dBs en el caso de una tasa de codificación global $r = 1/2$. Para el caso de una tasa de codificación global $r = 3/4$, esta mejora se produce para valores de Eb/No superiores a 8 dBs.

Como en los casos anteriores, podemos hacer un cálculo aproximado de la ganancia de codificación obtenida con sólo comparar los valores de tasa de error obtenidos con la herramienta BERTool.

	Eb/No Performance for 10E-4 bit error rate	GAIN
Theoretical	11 dB	-
RS $r=3/4$	8 dB	3 dB
RS $r=9/10$	9 dB	2 dB
Convolutional $r=2/3$	8 dB	3 dB
Convolutional $r=5/6$	10 dB	1 dB
RS+Conc Rc=1/2	7 dB	4 dB
RS+Conc Rc=3/4	8 dB	3 dB

Tabla 7. Resumen de la ganancia conseguida

Tomando como referencia una BER de 10E-4, para la modulación 16QAM y tasa de codificación $Rc = 1/2$ hemos obtenido una ganancia de 4dBs en términos de Eb/No. Obviamente, como en los casos de los esquemas de modulación y codificación

analizados anteriormente, la ganancia de codificación aumenta al incrementar la relación señal a ruido tal como muestran las líneas de tendencia de la figura 35.

4.5. BER vs Eb/No modulación 64QAM

También para la modulación 64QAM hemos calculado la evolución de la tasa de error (BER) para tres casos distintos:

- modulación 64QAM con codificación Reed-Solomon,
- modulación 64QAM con codificación convolucional y
- modulación 64QAM con los dos codificadores concatenados y tasas de codificación como las previstas en el estándar para esta modulación.

También hemos comparado los resultados obtenidos con los que se obtienen cuando se considera una modulación 64QAM sin codificar.

4.5.1. Codificación Reed-Solomon

En la siguiente gráfica podemos observar los resultados de la evolución de la tasa de error para una modulación 64QAM con codificación Reed-Solomon.

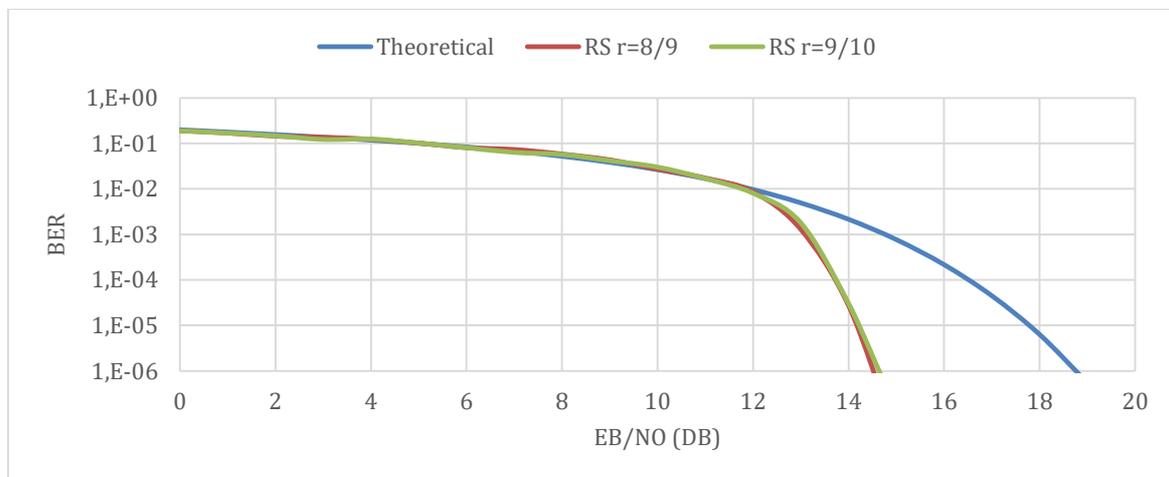


Figura 36. BER vs Eb/No modulación 64QAM y codificación RS

En la figura 36 podemos ver que la línea de tendencia de los valores obtenidos de BER para ambas tasas de codificación son prácticamente idénticas. Este resultado es el que podríamos esperar ya que, para ambas tasas de codificación, estamos corrigiendo 12 Bytes. Así mismo se aprecia que la codificación es efectiva para valores de Eb/No superiores a 12 dBs, tal como era de esperar ya que estamos considerando una modulación con una eficiencia espectral teórica (sin codificar) de 6 bits/Hz.

4.5.2. Codificación Convolucional

En la siguiente gráfica podemos observar los resultados de la BER en el caso de una modulación 64QAM y codificación Convolucional.

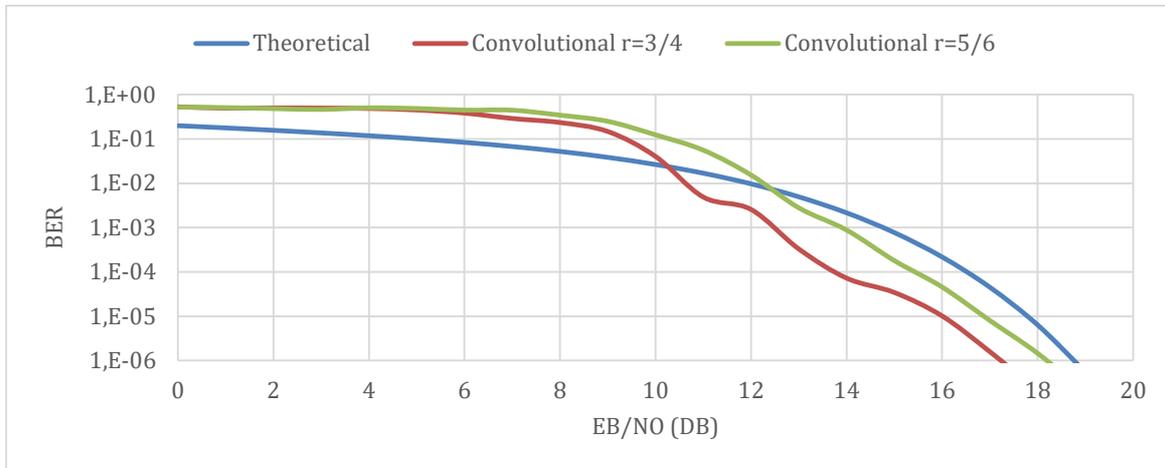


Figura 37. BER vs Eb/No modulaci3n 64QAM y codificaci3n convolucional

En la figura 37 podemos comprobar como los resultados obtenidos son muy similares a los calculados anteriormente, tambi3n para codificaci3n convolucional, considerando otros esquemas de modulaci3n. Por lo general, podemos observar que los valores de BER obtenidos con codificaci3n convolucional fluctu3n m3s que los obtenidos con Reed-Solomon ya que como se observa en las gr3ficas anteriores, la l3nea de tendencia para Reed-Solomon sigue una forma m3s uniforme respecto a la variaci3n de la relaci3n se3al a ruido.

4.5.3. Codificador Reed-Solomon + Convolucional

En la siguiente gr3fica podemos observar los resultados de la BER en el caso de una modulaci3n 64QAM y codificaci3n Reed-Solomon concatenada con codificaci3n convolucional.

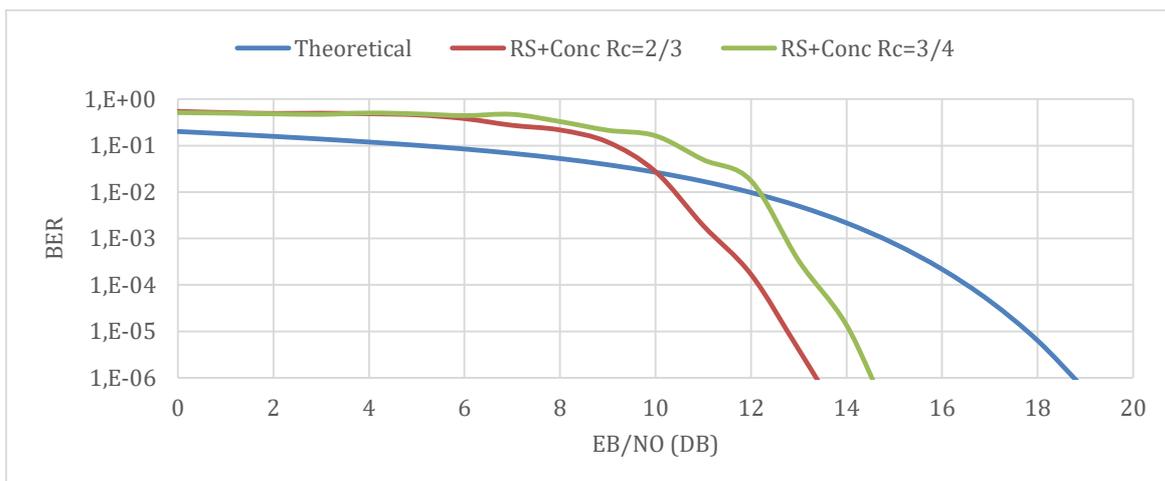


Figura 38. BER vs Eb/No modulaci3n 64QAM y codificaci3n RS + convolucional

La figura 38 muestra los resultados obtenidos al concatenar los dos codificadores mencionados anteriormente. Observando los valores obtenidos con la herramienta BERTool, podemos hacer un c3lculo de la ganancia aproximada obtenida con la codificaci3n, tal como se indica en la siguiente tabla considerando como referencia una BER de 10E-4.

	Eb/No Performance for 10E-4 bit error rate	GAIN
Theoretical	15 dB	-
RS r=8/9	13 dB	2 dB
RS r=9/10	13 dB	2 dB
Convolutional r=3/4	13 dB	2 dB
Convolutional r=5/6	14 dB	1 dB
RS+Conc Rc=2/3	12 dB	3 dB
RS+Conc Rc=3/4	13 dB	2 dB

Tabla 8. Resumen de la ganancia conseguida

Para la modulación 64QAM y tasa de codificación $R_c = 1/2$ hemos obtenido una ganancia de 3dB en términos de Eb/No. También como en los casos anteriores podemos observar como la ganancia de codificación aumenta al aumentar la relación señal a ruido (véase figura 38).

4.6. Sincronización

Según lo visto en el apartado 3.2.6, haciendo uso de la secuencia preámbulo deberíamos ser capaces de:

- a) Detectar su presencia dentro de la trama enviada,
- b) Estimar la respuesta impulsional del canal, de modo que nos permita determinar, en el caso del canal de dos rayos considerado en el estudio, los niveles de ambos rayos y el retardo entre ambos.

Para realizar las pruebas hemos introducido en la interfaz de usuario un retardo de canal de 16 muestras con una amplitud de 0.5.

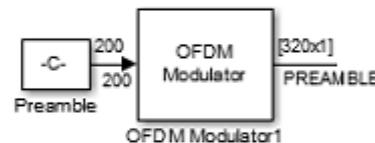
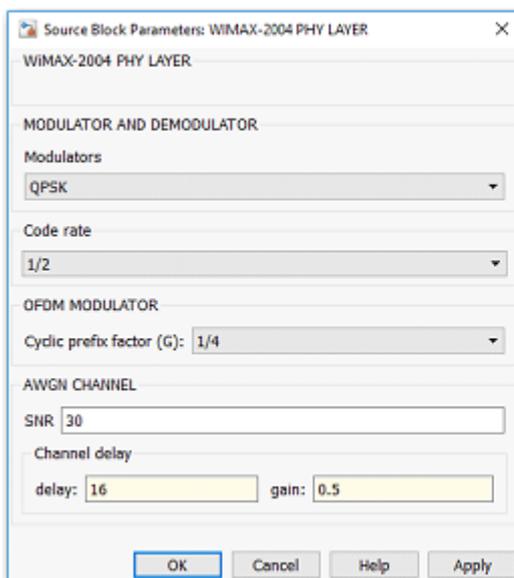


Figura 39. Interfaz de usuario y salida del Modulador OFDM para la secuencia preámbulo

Como se puede observar en la figura 39, con la configuración seleccionada desde la interfaz de usuario, tenemos un paquete preámbulo de 320 muestras a la salida del Modulador OFDM.

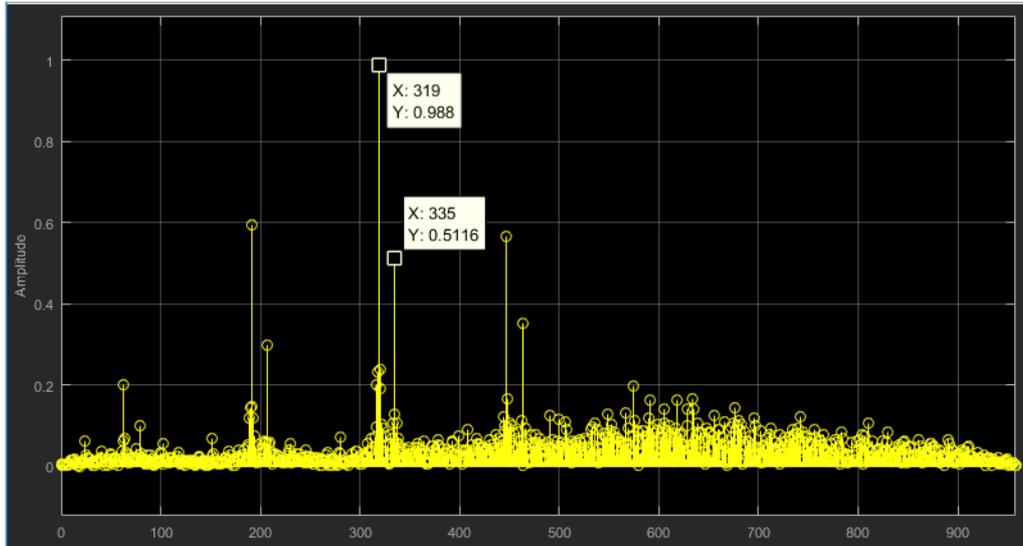


Figura 40. Salida del correlador

En la figura 40 podemos ver la salida de la correlación cruzada entre la trama enviada y una secuencia preámbulo generada en el receptor. Si nos fijamos en la figura 40, observamos como tenemos un máximo de correlación en la muestra 319. Este máximo se encuentra en la última muestra de la secuencia preámbulo por lo que vemos que somos capaces de detectarla. Por otro lado, podemos observar que 16 muestras más adelante, en la muestra 335 de la correlación, tenemos un máximo relativo de correlación con la misma amplitud que habíamos introducido en la amplitud del retardo del canal desde la interfaz de usuario, por lo que vemos que somos capaces de detectar la respuesta impulsional del canal.

Haciendo distintas pruebas para diferentes casos, podemos comprobar que siempre somos capaces de detectar la secuencia preámbulo y los retardos.

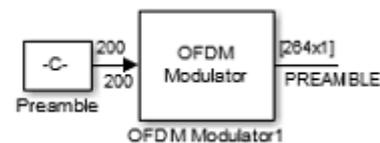
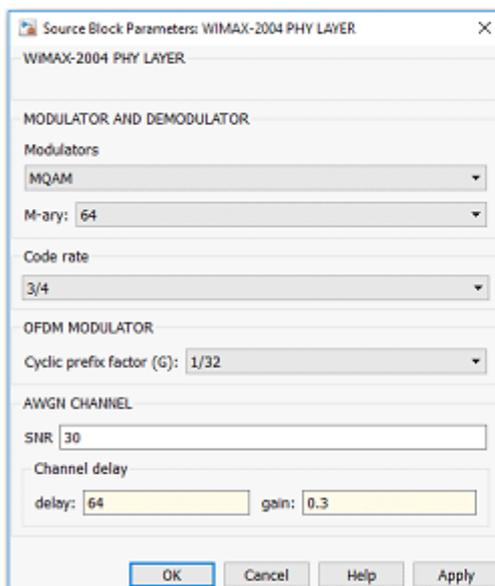


Figura 41. Interfaz de usuario y salida del Modulador OFDM para la secuencia preámbulo

En este caso hemos cambiado la modulación y hemos escogido un retardo de 64 muestras con una amplitud de 0.3. El preámbulo a la salida del demodulador ahora es de 264 muestras.

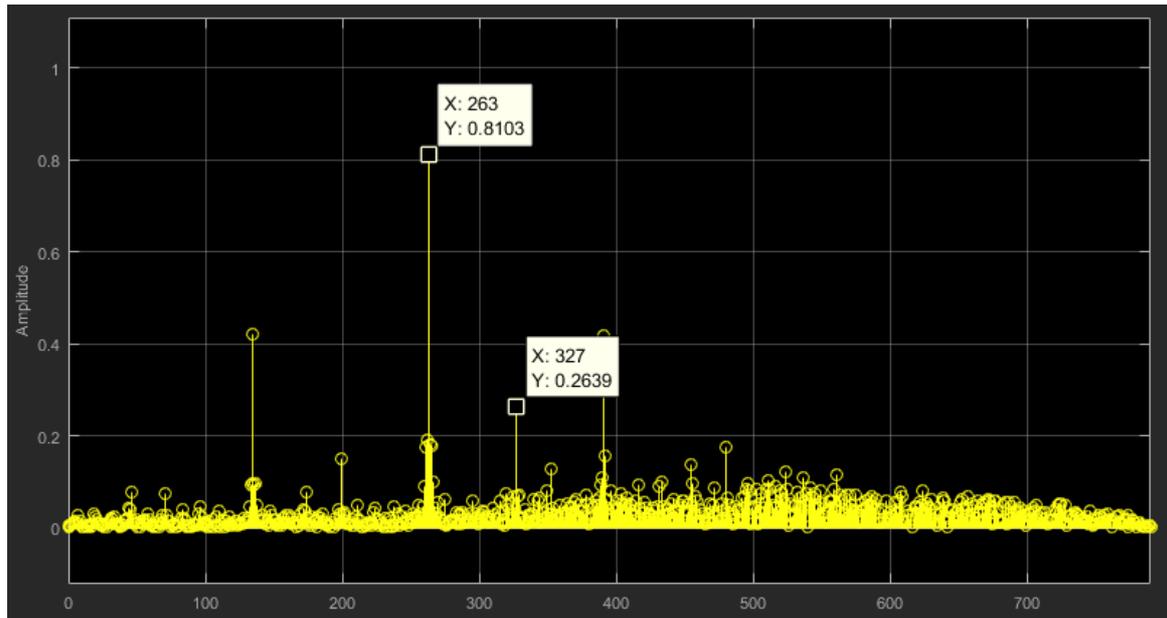


Figura 42. Salida del correlador

En la figura 42 podemos ver como ahora el máximo de correlación se encuentra en la muestra 263 correspondiente al preámbulo y tenemos el máximo relativo 64 muestras más adelante con una amplitud de 0.2639 correspondiente al retardo.

5. Presupuesto

Para establecer el presupuesto del proyecto hemos considerado tres costes fundamentales: coste de la mano de obra, coste del software y amortización de los ordenadores utilizados en la simulación.

5.1. Coste mano de obra

El cuadro de costos de personal a continuación se basa en las siguientes figuras:

- Un ingeniero junior recibiendo 10,30 euros por hora.
- El empleado cuesta un 30% más debido a la seguridad social y otros impuestos, elevando el coste por hora a 13.39 euros.

Tarea	Horas dedicadas	Coste (euros)
WP 1: Revisión estado del arte y planificación.	33	441,87
WP 2: Preparación entorno de simulación	60	803,4
WP 3: Diseño del TX	160	2142,4
WP 4: Diseño del RX	260	3481,4
WP 5: Integración TX-RX	48	642,72
WP 6: Escritura de la memoria	44	589,16
Total	605	8.100,95

Tabla 9. Coste mano de obra

5.2. Coste software

El único software no gratuito utilizado en el desarrollo del proyecto es Matlab. El coste se describe a continuación.

Licencia de Matlab para la educación 500€/año
Usada durante 143 días
Coste: 196€

Tabla 10. Coste Software

5.3. Amortización

El proyecto se ha llevado a cabo únicamente con un ordenador de sobremesa

Ordenador de sobremesa: 900€
Periodo de amortización, 4 años, valor residual, 25%
Usado durante 143 días
Depreciación: 88,15€

Tabla 11. Depreciación

6. Conclusiones y líneas futuras:

Para la realización de este proyecto ha sido esencial la búsqueda y procesado de la información relacionada con el estándar WiMAX-2004 así como del funcionamiento de la herramienta Simulink de Matlab.

La realización de este proyecto me ha permitido validar y poner en práctica conceptos y conocimientos de comunicaciones adquiridos durante los estudios universitarios, así como aprender y validar otros nuevos como las técnicas de codificación. Una parte muy importante del desarrollo del proyecto ha sido también el aprendizaje de uso y funcionamiento de la herramienta Simulink.

La herramienta Simulink de Matlab nos ha permitido hacer el diseño y la implementación de la capa física del estándar WiMAX-2004 así como la simulación de la misma para la obtención de resultados cualitativos y cuantitativos.

- Se han identificado los bloques correspondientes a cada módulo del sistema de telecomunicación especificado por el estándar.
- Se ha dimensionado cada bloque de forma que su funcionamiento fuese el correcto y deseado.
- Se ha creado una interfaz de usuario de modo que el sistema es fácilmente reconfigurable, lo que permite evaluar de modo razonablemente simple las prestaciones de las distintas configuraciones del mismo.

La interfaz de usuario se ha programado de forma que el usuario pueda escoger la modulación, la tasa de codificación, la longitud del prefijo cíclico, la relación señal a ruido de la transmisión (SNR), así como el retardo y la amplitud del mismo introducidos por el canal. Al ejecutar la simulación del sistema el usuario puede ver a través de gráficos y distintas pantallas ("display"), la tasa de error (BER) de la transmisión en tiempo real, la constelación transmitida y recibida y la correlación cruzada de la señal transmitida con una señal preámbulo que permite la sincronización.

Para realizar un estudio del funcionamiento del sistema, hemos utilizado la aplicación BERTool de Matlab. Esta aplicación nos ha permitido hacer los cálculos de la relación BER vs E_b/N_0 para todos los modos de funcionamiento del sistema, y de esta forma, poder hacer un estudio y una comparativa de los distintos modos de funcionamiento del bloque de codificación de canal. Usando los valores obtenidos por la aplicación BERTool, hemos podido crear unas tablas que nos indican la ganancia en términos de E_b/N_0 obtenida por la codificación de canal.

A continuación, se exponen diferentes propuestas para proseguir con el desarrollo y evolución del software creado en este proyecto:

- Creación de un bloque que elija la modulación y la codificación de forma dinámica a partir de la estimación de la calidad del canal. A este método se le denomina "Link Adaptation".
- Hacer el sistema compatible para canales con distribución Rayleigh con desvanecimientos rápidos, como el característico de los sistemas de comunicaciones móviles, es decir un canal afectado por fading. Esto supone el uso de igualadores de canal en el dominio de la frecuencia para compensar, entre otros efectos degradantes, los cambios rápidos de fase debido a la frecuencia Doppler.

- Preparar el sistema para transmisión y recepción MIMO (Multiple-input Multiple-output)
- Integración del software en USRP de ETTUS para crear un demostrador de laboratorio, con transmisión a través del espectro radioeléctrico de la señal OFDM.

Bibliografia:

- [1] IEEE Standard 802.16-2004, October 2004. <http://ieee802.org/16/published.html>.
- [2] IEEE 802.16 Broadband Wireless Access Working Group, "Channel models for fixed wireless applications," IEEE 802.16a-03/01, 2003-06-27.
- [3] Carl Eklund, et.al., "WirelessMAN: Inside the IEEE 802.16 Standard for Wireless Metropolitan Area Networks," IEEE Press, 2006.
- [4] J. G. Andrews, A. Ghosh and R. Muhamed, "Fundamentals of WiMAX: Understanding Broadband Wireless Networking," Prentice Hall, 2007
- [5] Giordano, A. and Levesque, A. (n.d.). Modeling of digital communication systems using Simulink.
- [6] Es.mathworks.com. (2016). Create Reed-Solomon code from binary vector data - Simulink. [online] Available at: <http://es.mathworks.com/help/comm/ref/binaryinputrsencoder.html> [Accessed 29 Feb. 2016].
- [7] Es.mathworks.com. (2016). Decode Reed-Solomon code to recover binary vector data - Simulink. [online] Available at: <http://es.mathworks.com/help/comm/ref/binaryoutputrsdecoder.html> [Accessed 29 Feb. 2016].
- [8] Es.mathworks.com. (2016). Create convolutional code from binary data - Simulink. [online] Available at: <http://es.mathworks.com/help/comm/ref/convolutionalencoder.html> [Accessed 25 Mar. 2016].
- [9] Es.mathworks.com. (2016). Decode convolutionally encoded data using Viterbi algorithm - Simulink. [online] Available at: <http://es.mathworks.com/help/comm/ref/viterbidecoder.html> [Accessed 25 Mar. 2016].
- [10] Es.mathworks.com. (2016). Bit Error Rate (BER) - MATLAB & Simulink. [online] Available at: <http://es.mathworks.com/help/comm/ug/bit-error-rate-ber.html#a1058820615b1> [Accessed 02 May. 2016].
- [11] Es.mathworks.com. (2016). IEEE® 802.16-2004 OFDM PHY Link, Including Space-Time Block Coding - MATLAB & Simulink Example. [online] Available at: <http://es.mathworks.com/help/comm/examples/ieee-802-16-2004-ofdm-phy-link-including-space-time-block-coding.html?refresh=true> [Accessed 07 May. 2016].
- [12] Es.mathworks.com. (2016). Mask Editor Overview - MATLAB & Simulink. [online] Available at: <http://es.mathworks.com/help/simulink/gui/mask-editor-overview.html> [Accessed 06 Apr. 2016].
- [13] Masks, B., Overview, M. and Fundamentals, M. (2016). Create a Simple Mask - MATLAB & Simulink. [online] Es.mathworks.com. Available at: <http://es.mathworks.com/help/simulink/ug/how-to-mask-a-block.html> [Accessed 07 Apr. 2016].
- [14] Es.mathworks.com. (2016). Block Masks - MATLAB & Simulink. [online] Available at: <http://es.mathworks.com/help/simulink/block-masks.html> [Accessed 07 Apr. 2016].
- [15] Software Defined Radio: Architectures, Systems and Functions (Markus Dillinger, Kambiz Madani, Nancy Alonistioti) Page xxxiii (Wiley & Sons, 2003, ISBN 0-470-85164-3)

Glosario

- **CC (Convolutional Coder):** codificador que utiliza códigos convolucionales.
- **FEC (Forward Error Correction):** es un tipo de mecanismo de corrección de errores que permite su corrección en el receptor sin retransmisión de la información original.
- **ISPs (Internet Service Providers):** proveedores de servicios de internet.
- **LOS (Line of Sight):** propagación de las ondas electromagnéticas con línea de visión.
- **MIMO (Multiple Input Multiple Output):** multiple entrada multiple salida, aumenta la eficiencia espectral de un sistema de comunicación inalámbrica por medio de la utilización del dominio espacial.
- **NLOS (Non Line of Sight):** propagación de las ondas electromagnéticas sin línea de visión.
- **OFDM (Orthogonal Frequency Division Multiplexing):** técnica de transmisión que consiste en la multiplexación de un conjunto de ondas portadoras de diferentes frecuencias, donde cada una transporta información modulada en QAM o PSK.
- **PC (Personal Computer):** Ordenador personal.
- **PSK (Phase Shift Keying):** es una forma de modulación angular que consiste en hacer variar la fase de la portadora entre un número de valores discretos.
- **QAM (Quadrature Amplitude Modulation):** es una técnica que transporta dos señales independientes, mediante la modulación de una señal portadora, tanto en amplitud como en fase.
- **QoS (Quality of Service):** es el rendimiento promedio de una red de telefonía o de ordenadores, particularmente el rendimiento visto por los usuarios de la red.
- **RS (Reed-Solomon):** hace referencia al codificador Reed-Solomon.
- **SDR (Software Defined Radio):** es un sistema de radiocomunicaciones donde varios de los componentes típicamente implementados en hardware son implementados en software.
- **SNR (Signal Noise Ratio):** se define como la proporción existente entre la potencia de la señal que se transmite y la potencia del ruido que la corrompe.
- **USRP (Universal Software Radio Peripheral):** es un transceptor flexible y rentable que convierte un PC estándar en un potente sistema de generación de prototipos inalámbricos.
- **WP (Work Package):** paquete de trabajo.