



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA Y SISTEMAS DE TELECOMUNICACIÓN

PROYECTO FIN DE GRADO

TÍTULO: Diseño de un banco de laboratorio para sistemas de telecomunicación utilizando módulos SDR

AUTOR: Sergio García García

TITULACIÓN: Grado en Ingeniería de Sistemas de Telecomunicación

TUTOR: Antonio Pérez Yuste

DEPARTAMENTO: IAC

VºBº

Miembros del Tribunal Calificador:

PRESIDENTE: Sara Lana Serrano

TUTOR: Antonio Pérez Yuste

SECRETARIO: Pedro García del Pino

Fecha de lectura:

Calificación:

El Secretario,

Resumen

El objetivo de este proyecto fin de grado es rejuvenecer el laboratorio de la asignatura de cuarto semestre Teoría de la Comunicación. Hasta la publicación de este proyecto fin de grado, el laboratorio se basaba en cálculos de señal y ruido, de modulaciones analógicas y digitales, todo ello mediante *scripts* de MATLAB. Esto hacía que los alumnos no fueran del todo conscientes de cada una de las partes involucradas en los sistemas de comunicaciones.

Para ello se ha diseñado un entorno de trabajo en Simulink que consiste en un banco de pruebas para sistemas de telecomunicaciones. Este programa mantiene todas las características ya existentes en el laboratorio de la asignatura e incluye nuevas funcionalidades. Estas nuevas funcionalidades son: la transmisión y recepción de señales, utilizando el SDR Adalm-Pluto de Analog Devices, y la realización de cálculos de tasa de error de bit con diferentes efectos producidos por el canal.

Uno de los requisitos de diseño es utilizar herramientas del ecosistema MATLAB, por ello se ha utilizado Simulink para implementar el programa. Simulink permite diseñar sistemas de forma modular y muy visual, lo que favorece que el alumnado sea consciente en todo momento de cuáles son los bloques que forman parte de los sistemas. Además, esto permite una integración completa con MATLAB que permite a los docentes mantener algunas funcionalidades anteriores.

Los resultados de este proyecto fin de grado son: en primer lugar, el programa para utilizar como banco de pruebas desarrollado en Simulink, y, en segundo lugar, se proponen seis prácticas para que las realicen los alumnos, acompañadas de las seis soluciones correspondientes. En estas prácticas se trabaja con diferentes conceptos como: relación señal a ruido, modulaciones analógicas y digitales, codificación, efectos de canal, filtrado...

Abstract

The aim of this project is to update the laboratory part of the subject of fourth semester Theory of Communications. Before the publication of this final degree project the laboratory was based on MATLAB scripts for calculate signal and noise power and for analog and digital modulations representation. This organization produced that the students were not too conscious about all the parts taking part in a communications system.

For this reason, it has been designed a workspace in Simulink that consists in a telecommunications system bench table. This program keeps the essence of the laboratory but it also adds new functionalities. The new features are: transmission and reception of signals using an Analog Devices Adalm-Pluto SDR and the possibility of calculate the bit error rate of the signal received, when adding channel impairments.

One of the design requirements is to continue using programs of the MATLAB world, because of that Simulink is been choosen. Simulink is a very modular program and very visual. It works with blocks so it helps students to be aware of the blocks they are using. Besides, this allows full integration with MATLAB allowing teachers to keep some of the previous functionality.

The results of this project are: the Simulink benchtable with all the new functionalities, and six new practices for the students. These new practices come with the solutions. These practices go over different communication concepts like: signal to noise ratio, analog and digital modulations, coding, channel impairments, filtering...

Índice de contenidos

Resumen	1
Abstract.....	1
Índice de contenidos	1
Índice de figuras	5
Índice de tablas	8
Lista de acrónimos.....	9
1. Introducción, objetivos y estructura de la memoria	11
1.1 Introducción	11
1.2 Objetivos	12
1.3 Estructura de la memoria	12
2. Marco tecnológico.....	15
2.1 Historia de la tecnología SDR.....	15
2.2 Conceptos generales de la tecnología SDR.....	16
2.2.1 Definición.....	16
2.2.2 Arquitectura.....	16
2.2.3 Modo de funcionamiento	17
2.3 Envolverte compleja.....	18
2.4 Dispositivos SDR.....	19
2.5 Chipsets de RF	20
2.6 Aplicaciones SDR	21
2.7 Analog Devices ADALM-Pluto SDR.....	21
2.8 <i>Software</i>	23
2.8.1 MATLAB	23
2.8.2 WinIQSIM.....	23
2.8.3 GNU Radio.....	24
2.9 Utilización de SDR en enseñanza	26
2.10 Conocimientos y trabajos relacionados.....	27
3. Especificaciones y restricciones.....	29
3.1 Especificaciones.....	29
3.1.1 Rango de frecuencias	29
3.1.2 Señales analógicas.....	29
3.1.3 Cuantificador	30
3.1.4 Señales digitales	30
3.1.5 Codificación de canal	30

3.1.6	Modulación.....	31
3.1.7	Efectos del canal.....	32
3.2	Requisitos.....	33
3.2.1	Capacidad.....	33
3.2.2	Procesado.....	33
3.2.3	Fiabilidad.....	33
3.2.4	Disponibilidad.....	34
3.2.5	Accesibilidad.....	34
3.2.6	Usabilidad.....	34
3.2.7	Coste.....	34
3.3	Herramientas necesarias.....	35
3.3.1	MATLAB y Simulink.....	35
3.3.2	Simulink <i>Toolboxes</i>	36
4.	Descripción de la solución propuesta.....	37
4.1	Bloque de señales analógicas.....	37
4.1.1	Tonos 1 y 2.....	37
4.1.2	Aleatoria Uniforme.....	37
4.1.3	Gaussiana.....	37
4.2	Cuantificador.....	38
4.2.1	Uniforme.....	38
4.2.2	Ley A.....	38
4.2.3	Ley Mu.....	39
4.3	Señales digitales.....	39
4.3.1	Secuencia personalizada.....	39
4.3.2	PRBS.....	39
4.3.3	Archivo binario.....	40
4.4	Codificación de canal.....	41
4.4.1	BCH.....	41
4.4.2	Reed–Solomon.....	41
4.5	Cabeceras.....	42
4.6	Modulación.....	43
4.6.1	AM.....	43
4.6.2	DSB.....	44
4.6.3	SSB.....	45
4.6.4	FM.....	47

4.6.5	ASK.....	48
4.6.6	PSK.....	50
4.6.7	FSK.....	52
4.6.8	MSK	53
4.6.9	QAM.....	55
4.6.10	OFDM	57
4.7	Filtros	59
4.7.1	Filtro Paso Bajo.....	59
4.7.2	Coseno alzado y raíz de coseno alzado	59
4.7.3	Gaussiano	61
4.7.4	Filtro de preénfasis	62
4.8	Efectos del canal	63
4.8.1	AWGN	63
4.8.2	Distorsión lineal	63
4.8.3	Distorsión no lineal	64
4.8.4	Onda continua	65
4.8.5	Multitrayecto	66
4.9	Receptor	68
5.	Resultados	69
5.1	Práctica 1. Visualización de señales en Simulink	69
5.2	Práctica 2. Análisis de Señal y Ruido	73
5.3	Práctica 3. Modulaciones AM, DSB y SSB.....	77
5.4	Práctica 4. Modulación FM.....	83
5.5	Práctica 5. Modulaciones digitales.....	88
5.6	Practica 6. Introducción a DVB-T y Bluetooth	93
5.6.1	OFDM	93
5.6.2	DVB-T.....	94
5.6.3	GFSK.....	97
5.6.4	Bluetooth.....	98
6.	Planos	101
7.	Presupuesto	107
8.	Manual de usuario.....	109
8.1	Instalación y configuración.....	109
8.2	Selección del tipo de señal y de funcionalidad	110
8.3	Añadir nuevos bloques y funcionalidades	111

8.4	Contenido del proyecto	114
8.5	Utilizar imágenes como fuente de datos	114
9.	Conclusiones	117
9.1	Resumen.....	117
9.2	Líneas de trabajo futuro	118
10.	Referencias.....	119

Índice de figuras

Figura 1. Arquitectura hardware de un SDR	16
Figura 2. Envoltente compleja.....	18
Figura 3. Modulador IQ.....	18
Figura 4. RTL-SDR con el chipset RF R820T2 y demodulador RTL2832U.....	20
Figura 5. Estructura interna y PCB del ADALM-Pluto	21
Figura 6. Estructura interna del chip RF AD9363	22
Figura 7. Captura de pantalla de WinIQSIM.....	23
Figura 8. Interfaz de GNU Radio Companion.....	25
Figura 9. Demodulador FM utilizando GNU Radio y ADALM-Pluto	26
Figura 10. Demodulador FM utilizando Simulink y ADALM-Pluto.....	26
Figura 11. Organización del espectro en España entre 400 y 470 MHz	29
Figura 12. Detector de muestras perdidas	34
Figura 13. Editor de Máscaras de Simulink para el bloque de señales digitales.....	36
Figura 14. Cuatro bloques introducidos por las toolbox	36
Figura 15. Efecto de los cuantificadores sobre una senoide.....	38
Figura 16. Interfaz del bloque de secuencia binaria personalizada	39
Figura 17. Implementación del codificador Reed-Solomon.....	41
Figura 18. Estructura del bloque que añade las cabeceras	42
Figura 19. Estructura de las tramas a la salida del modulador.	42
Figura 20. Forma de onda (izquierda) y espectro (derecha) de AM.....	43
Figura 21. Componentes IQ para un tono modulado en AM	43
Figura 22. Forma de onda (izquierda) y espectro (derecha) de DSB	44
Figura 23. Componentes IQ de un tono modulado en DSB	44
Figura 24. Modulador SSB de Hartley	45
Figura 25. Modulador SSB de Weaver.....	45
Figura 26. Modulador SSB aplicando filtro de Hilbert	46
Figura 27. Forma de onda (izquierda) y espectro (derecha) de SSB	46
Figura 28. Componentes IQ de un tono modulado en SSB.....	46
Figura 29. Forma de onda (derecha) y espectro (izquierda) de FM	47
Figura 30. Componentes IQ de un tono modulado en FM	47
Figura 31. Esquema del modulador ASK.....	48
Figura 32. Componentes IQ de 8-ASK	48
Figura 33. Constelación de 8-ASK.....	48
Figura 34. Diagrama de ojo de 8-ASK con $\alpha = 0.7$	49
Figura 35. Forma de onda de ASK (en negro) a la salida del modulador IQ	49
Figura 36. Demodulador ASK.....	49
Figura 37. Componentes IQ de 8PSK	50
Figura 38. Constelación de 8PSK con fase $\pi/8$ y codificación de Gray	50
Figura 39. Diagrama de ojo de 8PSK con $\alpha = 0.7$	51
Figura 40. Forma de onda de PSK (en negro) a la salida del modulador IQ.....	51
Figura 41. Componentes IQ de 4-FSK	52
Figura 42. Forma de onda de 4-FSK (en negro) y señal enviada (en azul).....	52
Figura 43. Componentes IQ de MSK	53
Figura 44. Constelación de MSK desfasada $\pi/4$	53
Figura 45. Diagrama de ojo de MSK.....	54

Figura 46. Señal original y señal modulada en MSK	54
Figura 47. Componentes IQ de 64-QAM	55
Figura 48. Constelación de 64-QAM	55
Figura 49. Diagrama de ojo de 64-QAM con $\alpha = 0.7$	56
Figura 50. Forma de onda de 64-QAM (en negro) a la salida del modulador IQ	56
Figura 51. Esquema de generación de OFDM	57
Figura 52. Utilización del prefijo cíclico en OFDM	57
Figura 53. Bloque de OFDM.....	58
Figura 54. Espectro de OFDM	58
Figura 55. Respuesta al impulso del filtro coseno alzado	59
Figura 56. Respuesta al impulso del filtro raíz de coseno alzado.....	60
Figura 57. Respuesta al impulso de coseno alzado y raíz coseno alzado	60
Figura 58. Respuesta al impulso del filtro gaussiano	61
Figura 59. Respuesta en frecuencia del filtro de preénfasis	62
Figura 60. Efecto del AWGN sobre 16-QAM.....	63
Figura 61. Efecto de 2 dB atenuación y de 8° desplazamiento fase para 16-QAM.....	63
Figura 62. Efecto de la distorsión no lineal sobre 16-QAM.....	64
Figura 63. Efecto de la distorsión no lineal sobre tono en AM.....	64
Figura 64. Efecto de la CW sobre una modulación 8-PSK.	65
Figura 65. Efecto de la CW sobre un tono modulado en AM	65
Figura 66. Modelo de multitrayecto para dos rayos.	66
Figura 67. Implementación del sistema de multitrayecto con 4 rayos	66
Figura 68. Efectos del multitrayecto sobre una modulación 16-QAM.....	67
Figura 69. Parámetros de los rayos utilizados en el ejemplo.....	67
Figura 70. Bloques internos del receptor	68
Figura 71. Captura del bloque de filtrado en el receptor	68
Figura 72. Bloque receptor y display con la BER	68
Figura 73. Interfaz del sistema de la práctica 1	70
Figura 74. Capturas de los medidores de potencia en dBm y W para $y(t)$	70
Figura 75. Osciloscopio con ambas señales y diferentes medidas	71
Figura 76. Gráfica de densidad espectral de potencia en dBm/Hz.....	71
Figura 77. Parámetros obtenidos en Simulink utilizando Time Scope.....	72
Figura 78. Sistema de la práctica 2.....	73
Figura 79. Potencia de cada senoide.....	74
Figura 80. Potencia de las sinusoides	74
Figura 81. Potencia de ruido en un ancho de banda de 800 Hz.....	75
Figura 82. Densidad espectral de potencia de ruido	76
Figura 83. Diagrama de bloques de la práctica 3	77
Figura 84. Señal modulada en AM y su envolvente.....	78
Figura 85. Espectro de un tono modulado en AM.....	78
Figura 86. Espectro de un tono modulado en DSB	79
Figura 87. Espectro de un tono modulado en SSB-LSB (Lower Side Band).....	79
Figura 88. Montaje para ver la señal de salida en el analizador de espectros	80
Figura 89. Tono de 10kHz modulado en AM.....	80
Figura 90. Tono de 10kHz modulado en DSB	81
Figura 91. Tono de 10kHz modulado en SSB-USB	81
Figura 92. Tono de 500 kHz modulado en SSB-LSB	82

Figura 93. Tono de 500 kHz modulado en SSB-USB	82
Figura 94. Espectro de un tono de 3 kHz modulado en FM con $\Delta f = 8\text{kHz}$	83
Figura 95. Potencia en el ancho de banda de Carson y total para FM banda ancha.....	84
Figura 96. Representación temporal de un tono de 3 kHz en FM banda ancha	84
Figura 97. Espectro de FM banda estrecha para un tono de 3 kHz	85
Figura 98. Potencia en el ancho de banda de Carson y total para FM banda estrecha...	86
Figura 99. Representación temporal de un tono de 3kHz en FM banda estrecha	86
Figura 100. Tono de 50kHz modulado en FM de banda ancha.....	87
Figura 101. Tono modulado en FM de banda estrecha	87
Figura 102. Constelaciones 1 y 2 de la práctica 5	88
Figura 103. Constelaciones 3 y 4 de la práctica 5	89
Figura 104. Espectro paso bajo de 8-PSK para la práctica 5.....	89
Figura 105. Diagramas de ojo de 8-PSK con coseno alzado $\alpha=0.2$	89
Figura 106. Espectros paso filtrados en coseno alzado para la práctica 5.....	90
Figura 107. Comparación entre QPSK y MSK	90
Figura 108. Gráficas de BER teóricas y experimentales	91
Figura 109. Efectos de la distorsión sobre una 16-QAM	92
Figura 110. Comparación del ancho de banda de MSK, QPSK y 4-FSK en paso bajo .	92
Figura 111. Diagrama de bloques para obtener el espectro de OFDM	93
Figura 112. Espectro en banda base de OFDM con 4 pilotos y nulo en DC.....	93
Figura 113. Diagrama de bloques de DVB-T modo 2K.....	94
Figura 114. Parámetros de los rayos interferentes.....	95
Figura 115. Curvas de BER para DVB-T con AWGN y multitrayecto	95
Figura 116. Espectro banda base de DVB-T en modo 2k	96
Figura 117. Constelación de 64-QAM utilizada en DVB-T.....	96
Figura 118. Anchos de banda para GFSK y FSK con una señal de 1Mbps	97
Figura 119. Canalización de FSK y GFSK.....	97
Figura 120. Canales de Bluetooth LE y Bluetooth Classic	98
Figura 121. Diagrama de bloques de la interfaz radio de Bluetooth Classic.....	99
Figura 122. Espectro banda base de Bluetooth Classic en diferentes instantes	99
Figura 123. Curvas de BER para Bluetooth	100
Figura 124. Diagrama de bloques del sistema	101
Figura 125. Diagrama de bloques del bloque generador de señales analógicas.....	102
Figura 126. Diagrama de bloques del bloque generador de señales digitales	102
Figura 127. Diagrama de bloques del bloque codificador de canal.....	102
Figura 128. Diagrama de bloques del bloque cuantificador	103
Figura 129. Diagrama de bloques del bloque que añade la cabecera	103
Figura 130. Diagrama de bloques del bloque modulador.....	103
Figura 131. Diagrama de bloques del bloque de efectos de canal.....	104
Figura 132. Diagrama de bloques del modulador IQ	104
Figura 133. Diagrama de bloques del receptor.....	104
Figura 134. Diagrama de bloques del bloque DemodDecod.....	105
Figura 135. Bloques contenidos en el bloque More Blocks	105
Figura 136. Paquete de soporte de ADALM-Pluto	109
Figura 137. Instalación y test de conexión de ADALM-Pluto	109
Figura 138. Selector de fuente analógica o digital	110
Figura 139. Selector de funcionalidad.....	110

Figura 140. Botón para inhabilitar los bloques no utilizados	110
Figura 141. Bloque que incluye la nueva funcionalidad	111
Figura 142. Bloque contenedor de los codificadores digitales	111
Figura 143. Interior del bloque antes de la modificación	111
Figura 144. Añadir una nueva salida al Variant Sink	112
Figura 145. Circuito final después de la modificación	112
Figura 146. Abrir el editor de máscaras de Simulink	112
Figura 147. Editor de máscaras de Simulink	113
Figura 148. Pasos para utilizar una imagen como fuente digital	114
Figura 149. Asistente de importación de imágenes	115
Figura 150. Señales digitales seleccionando imagen del workspace de MATLAB	115
Figura 151. Guardar la señal recibida en el workspace	115
Figura 152. Resultado de la ejecución	116

Índice de tablas

Tabla 1. Comparativa de diferentes dispositivos SDR	19
Tabla 2. Características de los chipsets de RF	20
Tabla 3. Parámetros del generador de señales analógicas	29
Tabla 4. Parámetros de los cuantificadores	30
Tabla 5. Parámetros del generador de señales digitales	30
Tabla 6. Parámetros del codificador	30
Tabla 7. Parámetros de las modulaciones analógicas	31
Tabla 8. Parámetros de las modulaciones digitales	31
Tabla 9. Parámetros del modulador IQ	31
Tabla 10. Parámetros de OFDM	32
Tabla 11. Parámetros de los filtros	32
Tabla 12. Parámetros de los efectos del canal	32
Tabla 13. Polinomios generadores de PRBS	40
Tabla 14. Códigos de Barker	42
Tabla 15. Parámetros de la práctica 1	69
Tabla 16. Parámetros de la práctica 2	73
Tabla 17. Parámetros de la práctica 3	77
Tabla 18. Parámetros de la práctica 4	83
Tabla 19. Parámetros de la práctica 5	88
Tabla 20. Tasa de Error de Bit para 8-PSK y 64-QAM	91
Tabla 21. Parámetros de OFDM	93
Tabla 22. BER calculada para DVB-T con AWGN y multitrayecto	95
Tabla 23. Comparativa entre Bluetooth LE y Bluetooth Classic	98
Tabla 24. Cálculos de BER para Bluetooth	100
Tabla 25. Presupuesto destinado a recursos humanos	107
Tabla 26. Presupuesto destinado a equipamiento hardware	107
Tabla 27. Presupuesto destinado a paquetes software	107
Tabla 28. Presupuesto destinado a material fungible	107
Tabla 29. Presupuesto total del proyecto	108
Tabla 30. Presupuesto para la creación de un laboratorio	108

Lista de acrónimos

- ADC: *Analogue to Digital Converter*
- ADS-B: *Automatic Dependent Surveillance Broadcast*
- AIS: *Automatic Identification System*
- AM: *Amplitude Modulation*
- ASK: *Amplitude-Shift Keying Modulation*
- AWGN: *Additive White Gaussian Noise*
- BCH: *Bose–Chaudhuri–Hocquenghem*
- BER: *Bit Error Rate*
- CDMA: *Code Division Multiple Access*
- CFR: *Crest Factor Reduction*
- DAC: *Digital to Analogue Converter*
- DDC: *Digital Down Converter*
- DPD: *Digital Pre-Distortion*
- DRM: *Digital Radio Mondiale*
- DSB: *Double SideBand Modulation*
- DSP: *Digital Signal Processor*
- DUC: *Digital Up Converter*
- DVB: *Digital Video Broadcasting*
- EVM: *Error Vector Magnitude*
- FM: *Frequency Modulation*
- FPGA: *Field Programmable Gate Arrays*
- FSK: *Frequency Shift Keying Modulation*
- GNSS: *Global Navigation Satellite System*
- GSM: *Global System for Mobile communications*
- GUI: *Graphic User Interface*
- HDL: *Hardware Description Language*
- IF: *Frecuencia intermedia*
- IoT: *Internet of Things*
- LNA: *Low Noise Amplifier*
- LTE: *Long Term Evolution*
- MSK: *Minimum-Shift Keying Modulation*
- OFDM: *Orthogonal Frequency Division Multiplexing*
- PA: *Power Amplifier*
- PAPR: *Peak to Average Power Ratio*
- PRBS: *PseudoRandom Binary Sequence*
- PSK: *Phase-Shift Keying Modulation*
- QAM: *Quadrature Amplitude Modulation*
- RAM: *Random Access Memory*
- RF: *Radiofrecuencia*
- RFIC: *Radio Frequency Integrated Circuit*
- RFID: *Radio Frequency Identification*
- RMS: *Root Mean Square*
- RS: *Reed-Solomon*
- SDR: *Software-Defined Radio*
- SNR: *Signal to Noise Ratio*

- SoC: *System on a Chip*
- SSB: *Single-Side Band Modulation*
- UMTS: *Universal Mobile Telecommunications System*
- USB: *Universal Serial Bus.*
- USRP: *Universal Software Radio Peripheral*
- VHF: *Very High Frequency*
- WCDMA: *Wideband Code Division Multiple Access*
- WLAN: *Wireless Local Area Network*

1. Introducción, objetivos y estructura de la memoria

1.1 Introducción

Uno de los principales valores de la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación (ETSIST) es el compromiso con la formación y el desarrollo de todo el personal que forman parte de la ETSIST. Para seguir manteniendo este compromiso, es necesario realizar actualizaciones periódicas en los diferentes departamentos para reforzar las carencias que van surgiendo con el paso del tiempo.

Con la misión de mantener estos valores, se han detectado algunos puntos de mejora en el laboratorio de la asignatura de cuarto semestre Teoría de la Comunicación. Este proyecto fin de grado se encarga de modernizar el laboratorio de la escuela para conseguir una mejora en el aprendizaje de los alumnos.

El laboratorio actual está basado en la utilización de *scripts* de MATLAB. Estos *scripts* sirven para enseñar a los alumnos el funcionamiento de los sistemas de comunicación. La utilización de estos *scripts* hace que, en ocasiones, los alumnos no sean del todo conscientes de lo que se ejecuta en el código y, por tanto, pueden, no entender lo que ocurriendo o sentirse abrumados al ver muchas líneas de código.

En las asignaturas de la carrera se suelen estudiar los sistemas como un conjunto de bloques que realizan modificaciones en la señal como, por ejemplo, filtrar, modular o añadir efectos de canal. Teniendo esto en cuenta, sería lógico y adecuado que en los laboratorios se trabajase de la misma manera, es decir, con prácticas que utilicen diagramas de bloque a nivel visual.

Para ello se ha diseñado e implementado un banco de pruebas de laboratorio en Simulink. La utilización de Simulink es clave. La interfaz de Simulink muestra todos los subsistemas en forma de diagrama bloques. Esta característica permite ser coherentes con la forma en la que se enseñan los sistemas. Se consigue que los alumnos sean plenamente conscientes de cómo se va modificando la señal y cuáles son los bloques participan en los distintos sistemas.

Además de actualizar el laboratorio, se han introducido nuevas funcionalidades como son la tecnología SDR que permitirá al profesor transmitir señales y que las reciban los alumnos o que se analicen desde un analizador de espectros. Esta funcionalidad permite que todos los alumnos de la Escuela puedan conocer esta tecnología y no solo los alumnos de la especialidad de Sistemas de Telecomunicación. Otra funcionalidad añadida es el cálculo de la tasa de error de bits de forma sencilla.

Esta actualización del laboratorio trae consigo seis nuevas prácticas que mantienen la esencia de la asignatura, pero incluyendo las mejoras ya mencionadas. Estas prácticas se han diseñado partiendo de las ya existentes de la asignatura Teoría de la Comunicación para mantener las competencias del laboratorio. En ellas se incluye la posibilidad de utilizar o no la tecnología SDR.

Los resultados que se esperan de este proyecto son modernizar el laboratorio sin invertir mucho dinero, ya que el presupuesto que se necesita para sacar partido a todas las funcionalidades es relativamente bajo.

1.2 Objetivos

Los objetivos principales de este proyecto fin de grado son:

En primer objetivo es modernizar y actualizar el laboratorio de la asignatura de cuarto semestre Teoría de la Comunicación y de cursos posteriores.

El segundo objetivo es que los alumnos descubran la tecnología SDR en sus primeros años de carrera y que no tengan que esperar hasta el último curso para conocerla. Para ello se quiere implantar esta tecnología en el laboratorio.

El tercer objetivo es diseñar un banco de trabajo utilizando Simulink y utilizarlo para generar seis prácticas para que los alumnos las realicen en el laboratorio.

1.3 Estructura de la memoria

El proyecto se ha estructurado en capítulos los cuales se describen en los siguientes párrafos:

El capítulo 1 se corresponde con esta introducción.

En el capítulo 2 se realiza una introducción al marco tecnológico de la tecnología SDR. Primero se analiza la historia de esta tecnología, desde sus orígenes hasta la situación actual. Después se explican los fundamentos matemáticos, el modo de funcionamiento y los componentes principales de un SDR. También se realiza un pequeño estudio de mercado analizando las prestaciones que ofrecen los dispositivos y el *software* hoy en día. Por último, se han resumido algunos proyectos educativos implementados en otras universidades en los que se utiliza tecnología SDR.

En el capítulo 3 se han explicado cuáles son las especificaciones técnicas que debe cumplir el programa desarrollado. También se realiza una descripción de los requisitos mínimos para poder utilizar este programa de forma correcta. Por último, se describen las herramientas *software* que se van a utilizar para desarrollar el proyecto.

En el capítulo 4 se describe la solución propuesta. En este punto se explican todas y cada una de las funcionalidades han sido diseñadas e implementadas. La explicación de cada una de las funcionalidades se divide en dos partes: en primer lugar, una parte teórica y, en segundo lugar, se explica cómo se ha implementado en el proyecto. Este capítulo es uno de los más extensos de la memoria porque se explican muchos conceptos de los sistemas de comunicación como son la generación de señales, la cuantificación, la codificación de canal, los diferentes tipos de modulación, filtros, efectos de canal...

En el capítulo 5 se utiliza el programa implementado en el capítulo 4, para crear seis prácticas de laboratorio de sistemas de comunicación. Estas prácticas abarcan diferentes áreas de interés de los laboratorios de la escuela. A continuación, se explica, brevemente, en qué consisten estas prácticas. Las prácticas 1 y 2 se centran en la visualización y generación de señales y en el análisis de señal y ruido. Las prácticas 3 y 4 consisten en estudiar las modulaciones analógicas. La práctica 5 trata de modulaciones digitales. Por último, la práctica 6 sirve como introducción a los estándares de DVB-T y Bluetooth.

En el capítulo 6 se muestran los planos del proyecto. Al ser un proyecto *software* se muestran los diagramas de bloques de los bloques implementados en Simulink.

En el capítulo 7 se realiza un presupuesto del coste de realización de este proyecto fin de grado y un presupuesto adicional para montar un laboratorio desde cero.

En el capítulo 8 se incluye el manual de usuario donde se describe cómo utilizar el programa y cómo incluir nuevas funcionalidades, si así lo desea el usuario.

En el capítulo 9 se detallan las conclusiones y las líneas de trabajo futuro.

----- Hoja en blanco -----

2. Marco tecnológico

2.1 Historia de la tecnología SDR

La aparición de la tecnología SDR se remonta a 1985 cuando el equipo de investigación de E-Systems Inc. en Garland, Texas, se vio en la necesidad de crear un equipo de procesamiento de señal, configurado como receptor radio, para poder trabajar con las técnicas de procesamiento digital que estaban desarrollando en el laboratorio. Tuvieron que crear su propio equipo porque el ordenador que tenían no era capaz de trabajar a la velocidad requerida por las señales de banda ancha. A este equipo le llamaron DSP Lab Software Radio [1]. Este equipo era programable y podía llevar a cabo diferentes formas de procesamiento, cosa que entonces no permitían los procesadores dedicados.

En 1992, J. Mitola publicó de forma teórica los principios de la arquitectura de la tecnología SDR [2]. Desde entonces se publicaron varios dispositivos como, por ejemplo, el SpeakEasy phase I cuyos objetivos principales eran: transmitir más de diez tipos de radio militar operando en un rango de frecuencias entre 2 y 2000 MHz y el segundo objetivo era poder introducir de forma sencilla nuevas técnicas de modulación y codificación que pudieran surgir en el futuro [3].

El SpeakEasy utilizaba el DSP TMS320C30 CMOS de Texas Instruments y una gran cantidad de circuitos integrados, lo que provocaba que fuese un dispositivo bastante grande. No es hasta los años 2000 cuando la empresa Broadcom lanza el SoC BCM21551 RF CMOS que permite reducir enormemente la complejidad y el tamaño de los SDR a un solo chip. Es a partir de entonces cuando la tecnología SDR adquiere mayor popularidad entre los radioaficionados.

Junto al desarrollo de los dispositivos *hardware*, aparecen diferentes programas *software* para utilizar esta tecnología. Uno de los programas más populares es GNU Radio creado en 2001 por Eric Blossom.

GNU Radio [4] es un programa de código abierto que proporciona las herramientas para trabajar con SDR y para procesamiento de señal. Al ser un programa de código abierto tiene una gran comunidad de usuarios y desarrolladores a lo largo del mundo. La forma de trabajar en este programa es mediante diagramas de flujo. En ellos pueden añadirse bloques de procesamiento mediante “arrastrar y soltar” (*drag-and-drop*) bloques sobre un panel y uniendo éstos mediante flechas según el orden deseado, haciendo que los diseños sean modulares y puedan implementarse y modificarse de forma rápida y fácil.

El inicio de GNU Radio se vio obstaculizado por la falta de dispositivos SDR de bajo coste. Fue entonces cuando Matt Ettus, uno de los principales colaboradores de GNU Radio, comenzó a trabajar en 2003 en el diseño del proyecto *Universal Software Radio Peripheral* (USRP) [5]. En 2004 se comenzó la fabricación del USRP1 que fue lanzado en 2005. USRP1 se convirtió en el primer SDR de bajo coste que permitió a estudiantes y radioaficionados iniciarse en el mundo de la radio definida por *software*. Con el paso de los años, USRP se ha convertido en una gran familia de dispositivos SDR.

2.2 Conceptos generales de la tecnología SDR

2.2.1 Definición

La IEEE define SDR como “radio en la que algunas o todas las funciones de la capa física están definidas por software” [6]. Es un sistema de radiocomunicación en el que se utilizan técnicas de procesamiento digital de la señal para sustituir a los elementos físicos como mezcladores, moduladores/demoduladores, filtros, etc., eliminando la complejidad *hardware* y trasladando estas funciones al plano digital [7].

La tecnología SDR es muy flexible porque puede adaptarse a diferentes frecuencias, modulaciones y anchos de banda de forma sencilla, sin necesidad de modificar la parte *hardware*. Esta flexibilidad permite que se pueda recibir y/o transmitir con un mismo dispositivo los diferentes estándares UMTS, LTE, GSM, Bluetooth, etc. Incluso actualizar a los nuevos estándares que surjan con el paso del tiempo simplemente modificando la parte digital.

2.2.2 Arquitectura

La arquitectura de los SDR se divide en dos partes [8]. Por un lado, la parte de RF y por otro lado la parte *software*, separadas por un bloque conversor analógico-digital (ADC) o conversor digital-analógico (DAC). Una arquitectura genérica de los SDR puede ser la mostrada en la Figura 1, donde aparecen en la parte superior la zona de RF y en la parte inferior el bloque correspondiente a la parte digital.

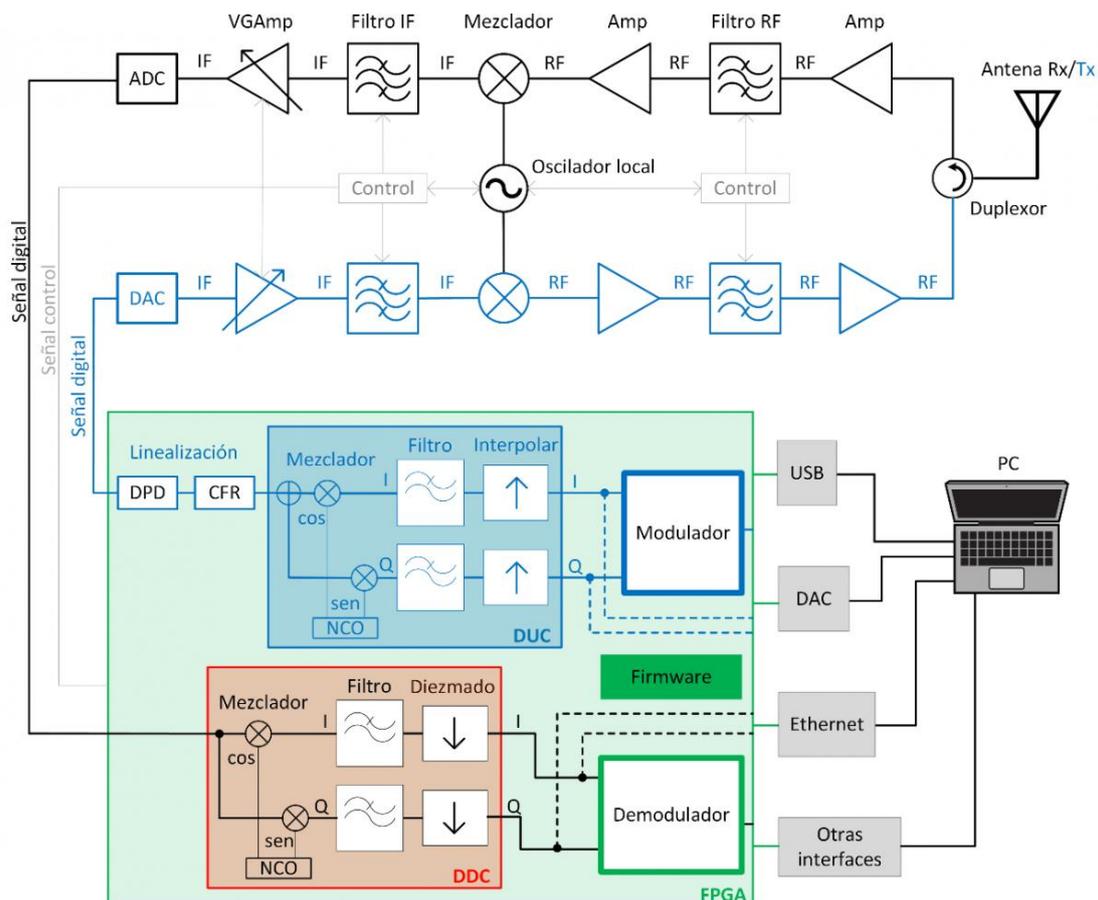


Figura 1. Arquitectura hardware de un SDR

En la Figura 1 puede observarse que existen dos ramas: una para transmisión y otra para recepción. Pueden compartir antena o pueden tener una, o varias, cada rama [9]. A continuación, se analiza la etapa de recepción:

Primero se acondiciona la señal recibida utilizando un amplificador de bajo nivel de ruido (LNA) para poder trabajar con niveles adecuados de amplitud, filtrado para quedarse con la banda de frecuencias RF deseada y de nuevo amplificación.

El segundo paso es convertir RF a frecuencia intermedia (IF) mediante un mezclador u oscilador local, controlado por software, que adapta la señal de entrada a las características que necesita el ADC. En este paso se vuelve a filtrar y amplificar la señal con un amplificador variable que adapta la amplitud de entrada al ADC.

El tercer y último paso es convertir la señal a digital mediante el ADC. Una vez se tiene la señal digital se obtienen las componentes en fase I y en cuadratura Q banda base utilizando un demodulador IQ. Se produce en el DDC un filtrado y diezmado de la señal para reducir la tasa de muestreo. Obtenidas las componentes IQ se demodulan para obtener la señal original en el PC. Este demodulador puede estar en el PC o en una matriz de puertas lógicas programables en campo o *Field-Programmable Gate Array* (FPGA).

Para la etapa de transmisión se siguen los pasos en orden inverso:

En primer lugar, se envían mediante USB o Ethernet las componentes IQ o bien se envía la información y se modula después en una FPGA. Tras esto se produce una interpolación para aumentar la tasa de muestreo de la señal en el DUP. Se modulan las señales IQ con un modulador IQ.

Antes de convertir a analógico, se aplica un reductor del factor de cresta (CFR), para reducir la relación entre el valor de pico de la señal y su potencia media (PAPR), y de pre-distorsión digital (DPD), para linealizar la señal. Tras esto se convierte a analógico con el DAC.

A la salida del DAC tenemos la señal en IF. Se aplican diferentes etapas de filtrado y de amplificación para aumentar la ganancia. Tras esto se convierte la señal IF a RF mediante el oscilador local. Se vuelve a amplificar y filtrar y sale por la antena.

2.2.3 Modo de funcionamiento

Como ya se ha explicado en el apartado anterior, se puede utilizar un SDR para transmitir y recibir información. Podemos clasificar los SDR según el funcionamiento en:

- Simplex: solo puede transmitir o recibir y no puede cambiar de modo.
- Half-duplex: puede transmitir o recibir, pero no a la vez.
- Full-duplex: puede transmitir y recibir a la vez.

Otra clasificación de los SDR puede ser si necesita o no un ordenador para funcionar:

- Los más habituales necesitan un PC para funcionar.
- Existen otros que pueden utilizarse con un *smartphone* mediante el estándar USB OTG (*On-The-Go*), sin necesidad de un ordenador.
- Otros que permiten instalar un sistema operativo en el dispositivo y tienen una pantalla para modificar los parámetros, como el LAN-IQ SDR [10].

2.3 Envolvente compleja

El funcionamiento de los SDR no puede entenderse sin el concepto de envolvente compleja: cualquier señal paso banda real puede representarse como una señal compleja banda base. Por tanto, siendo $Y(f)$ una señal real paso banda, centrada en una frecuencia f_0 , se verifica que su transformada de Fourier es conjugada simétrica. Es decir:

$$Y(f) = Y^*(-f) \quad (1)$$

En este caso, es posible representar $Y(f)$ a partir de una señal banda base compleja $X_c(f)$ como:

$$Y(f) = \frac{1}{2}[X_c(f - f_0) + X_c^*(-f - f_0)] = \text{Re}[X_c(t)e^{j\omega_0 t}] \quad (2)$$

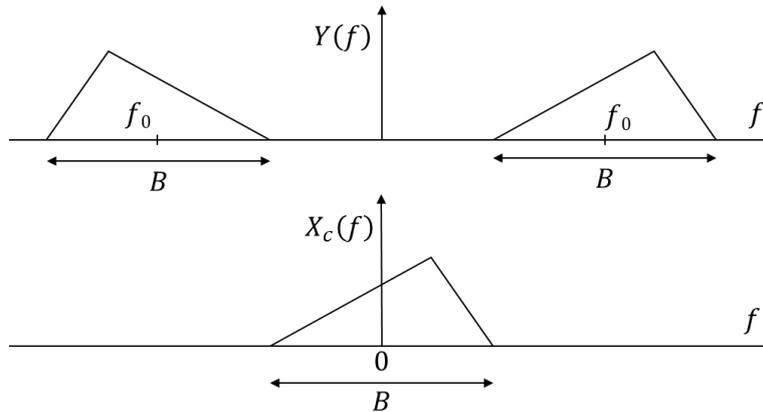


Figura 2. Envolvente compleja

Siendo la señal banda base compleja $X_c(t)$

$$X_c(t) = X_I(t) + jX_Q(t) \quad (3)$$

Y por tanto se puede obtener la señal original real $Y(t)$ de la siguiente manera:

$$Y(t) = X_I(t) \cos(\omega_0 t) - X_Q(t) \sin(\omega_0 t) \quad (4)$$

De la ecuación (4) se puede deducir que cualquier modulación real puede obtenerse mediante un modulador IQ como el indicado en la anterior ecuación. Este mismo concepto sirve también para un demodulador IQ.

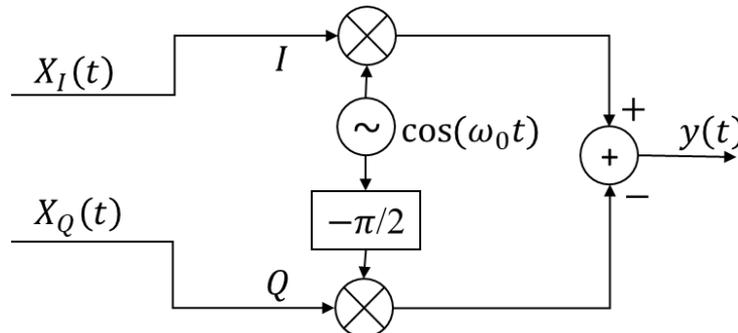


Figura 3. Modulador IQ

Los sistemas SDR llevan un modulador/demodulador IQ mediante el cual generan la señal RF de salida/entrada. La comprensión de este concepto es fundamental para entender el funcionamiento de la tecnología SDR.

2.4 Dispositivos SDR

Existen muchos modelos y marcas [11] incluyendo cada vez dispositivos más asequibles que abarcan diferentes presupuestos y capacidades. Los principales parámetros para determinar la potencia de un dispositivo SDR son:

- Número de antenas
- Rango de frecuencias
- Ancho de banda de RF
- Resolución del ADC/DAC
- Número de canales
- Interfaz de conexión y ancho de banda del bus de conexión
- Chipset RF
- FPGA

Un ejemplo de una las mejores opciones en cuanto a calidad de dispositivos y ecosistema *software* es la empresa National Instruments [12] en los que además de ofrecer muy buenos dispositivos también proporciona el programa de ordenador LabView para controlar sus SDR. El gran problema de estos dispositivos es su elevado precio que está por encima de los 1300€. Otras alternativas en distintos rangos de precio son LimeSDR [13], BladeRF [14] o ADALM-Pluto [15]. Este último modelo será el utilizado en este proyecto fin de grado y será analizado en profundidad más adelante.

Tabla 1. Comparativa de diferentes dispositivos SDR

Dispositivo	NI USRP-2900	LimeSDR	Blade RF	ADALM-Pluto
Número de antenas	2 Rx / 1 Tx	6 Rx / 4 Tx	2 Rx / 2 Tx	1 Rx / 1 Tx
Rango de frecuencias	70 MHz - 6 GHz	30 MHz - 3.8 GHz	300 MHz – 3.8 GHz	325 MHz – 3.8 GHz
Ancho de Banda RF	56 MHz	61.44 MHz	28 MHz	20 MHz limitado por USB 2.0 a 4 MHz
Resolución ADC/DAC	12 bit	12 bit	12 bit	12 bit
Canales	1	1	1	1
Interfaz de conexión	USB 3	USB 3	USB 3	USB 2.0 OTG
Ancho de banda Bus	5 Gbit/s	5 Gbit/s	5 Gbit/s	480 Mbit/s
Chipset RF	-	LMS7002M	LMS6002M	AD9363
FPGA	No	Altera40KLE	Altera40KLE	Xilinx Zynq 7000
Precio	1300 €	330 €	400 €	160 €

2.5 Chipsets de RF

Una vez analizados algunos dispositivos, serán analizados los chipsets de RF que incluyen estos. La función de los chipsets de RF es recibir y/o transmitir señales radio. Cuando actúan como receptor, trasladan la señal RF a IF, mientras que cuando actúan como transmisor realizan justo lo contrario, de IF a RF. En los últimos años, se ha conseguido integrar en un mismo chip las funcionalidades de los chips de RF junto con los demoduladores digitales. Existen varios tipos de chipsets de RF:

- Solo transmisor: poco frecuentes en los dispositivos SDR
- Solo receptor: muy frecuente en dongles de televisión. Son dispositivos baratos que suelen tener el chip RF R820T2 [16] o el E4000 junto a un demodulador RTL2832U [17]. En este caso, el chip RF se encarga de entregar la señal en IF al ADC del demodulador. En la Figura 4 se muestra la placa del RTL-SDR que utiliza este sistema.
- Transceptor: permiten transmitir y recibir en el mismo dispositivo. Estos son la mayoría de los utilizados para SDR: LMS7002M [18], AD9363 [19].

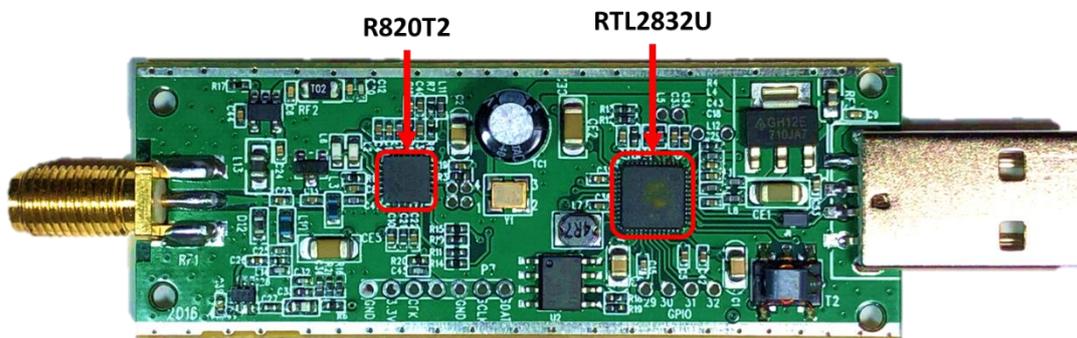


Figura 4. RTL-SDR con el chipset RF R820T2 y demodulador RTL2832U

Las características principales de estos chipsets de RF se recogen en la Tabla 2.

Tabla 2. Características de los chipsets de RF

Modelo	R820T2	LMS7002M	AD9363
Nº de pines	48 QFN	261 aQFN	144 BGA
Nº ADCs	-	4 de 12 bits	2 de 12 bits
Nº DACs	-	4 de 12 bits	2 de 12 bits
Banda de frecuencias Tx	-	30 – 3800 MHz	325 - 3800 MHz
Banda de frecuencias Rx	42 - 1002 MHz	30 – 3800 MHz	325 - 3800 MHz
Ancho de banda del canal	Hasta 8 MHz	Hasta 96 MHz	Hasta 20 MHz
Potencia de tx	-	0 dBm	7.5 dBm
Ganancia de rx	85 dB	89 dB	73 dB
Suelo ruido tx	-	-160 dBm/Hz	-156 dBm/Hz
Fig. ruido rx	3.5 dB	2.5 dB	2.9 dB

2.6 Aplicaciones SDR

Debido a la flexibilidad de esta tecnología existen multitud de aplicaciones y funcionalidades. A continuación, se listan las algunas de las más usuales:

- Uso militar: sistemas radar y de guerra electrónica
- Sistemas de radioastronomía
- Seguimiento de barcos vía transmisiones AIS
- Seguimiento de aviones mediante ADS-B.
- Recibir y transmitir radio y televisión comercial
- Construir redes GSM, UMTS y LTE
- Redes para IoT
- Seguimiento de satélites y sistemas GNSS
- Utilización de redes radio cognitivas¹
- Ciberataques y uso malicioso: *Sniffing, Jamming, Spoofing...*

2.7 Analog Devices ADALM-Pluto SDR

El dispositivo SDR elegido es el ADALM-Pluto de Analog Devices. La elección de este dispositivo es principalmente su bajo precio, la existencia de abundante documentación y gran variedad de *software* compatible. El dispositivo tiene diferentes modos de funcionamiento:

- Puede guardar una señal en su memoria flash y transmitirla en bucle
- Puede transmitir señales generadas en el propio dispositivo usando Síntesis Digital Directa (DDS)
- Puede transmitir y recibir en tiempo real

Puede trabajar en modo half-duplex a velocidad completa o full-duplex a la mitad de velocidad. El dispositivo se divide en dos bloques como puede apreciarse en la Figura 5. En los siguientes apartados se analizan estos dos bloques.

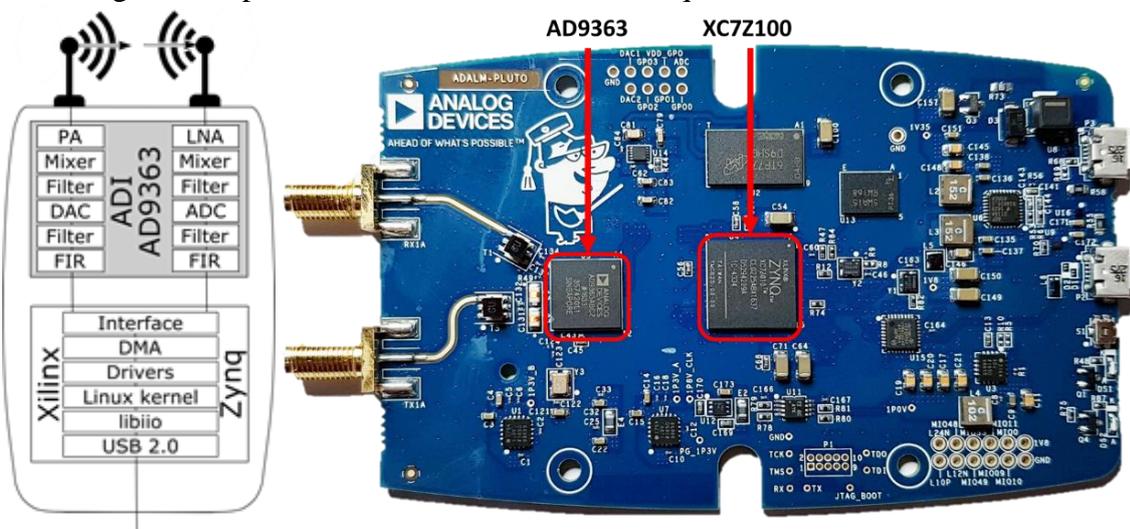


Figura 5. Estructura interna y PCB del ADALM-Pluto

¹ La radio cognitiva es un paradigma de radiocomunicación mediante el cual los equipos adaptan sus parámetros para aprovechar aquellas zonas libres del espectro radioeléctrico consiguiendo una eficiencia espectral mucho mayor que utilizando las técnicas convencionales.

En primer lugar, tenemos el SoC Xilinx Zynq XC7Z010 cuyas especificaciones principales son las siguientes:

- FPGA Artix-7 con 28k celdas lógicas, 2.1Mb de RAM, 80 DSP slices
- Procesador ARM Cortex-A9 MPCore con 1 núcleo a 667 MHz
- Conexión por USB 2.0 con ancho de banda de 4 MSPS

En segundo lugar, tenemos el chipset de RF ADI AD9363 RF Agile Transceiver con las siguientes características:

- 1 canal para transmisión y otro para recepción.
- Rango de frecuencias: 325 MHz - 3.8 GHz²
- Ancho de banda: 200 kHz – 20 MHz
- ADC/DAC de 12 bits
- Velocidad de salida variable entre 61.44 MSPS – 65.1 kSPS
- Precisión de modulación: EVM \leq 40 dB
- Calibración y corrección interna de I/Q
- Potencia máxima de transmisión 7 dBm
- Ruido del transmisor \leq -157 dBm/Hz
- Ganancia de recepción desde 0 a 74.5 dB con diferentes modos de ganancia.
- Figura de ruido del receptor: 2.5 dB utilizando ganancia máxima.

La estructura del chip de RF es la de la Figura 6. Se pueden observar que hay dos ramas con dos canales cada una: en la de transmisión hay 2 DAC y en la de recepción 2 ADC con sus correspondientes filtros. El oscilador local tiene resolución de 2.4 Hz. Para el lado receptor tiene varios amplificadores de bajo nivel de ruido (LNA) y para el transmisor varios amplificadores de potencia (PA).

Cada canal de recepción tiene tres entradas que pueden ser multiplexadas permitiendo sistemas con diversidad teniendo múltiples antenas. Esta funcionalidad no se implementa en el ADALM-Pluto porque solo hay una antena para transmisión y otra para recepción.

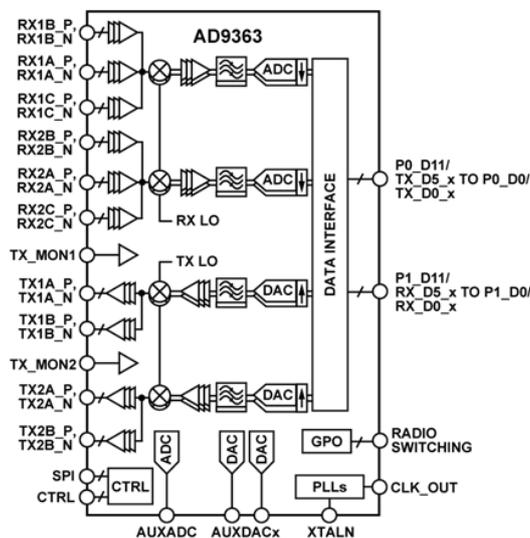


Figura 6. Estructura interna del chip RF AD9363

² Este rango de frecuencias puede aumentarse de forma relativamente sencilla utilizando un *hack* [20] hasta obtener un rango desde 70 MHz a 6 GHz.

2.8 Software

En el laboratorio de la asignatura Teoría de la Comunicación de la Escuela se han utilizado, tradicionalmente, dos programas: MATLAB y WinIQSIM, si bien durante el último curso se dejó de utilizar el segundo, pasando a realizar todas las prácticas sólo con MATLAB.

2.8.1 MATLAB

MATLAB se utiliza para analizar señales y modulaciones analógicas. Los alumnos realizan este análisis utilizando *scripts* desde la ventana de comandos. Esta forma de trabajar es muy poco amigable. Además, los alumnos no son realmente conscientes lo que están haciendo y solo se preocupan en analizar los resultados finales. Esta falta de realidad se pretende solventar con el programa que se desarrolla en este proyecto fin de grado.

2.8.2 WinIQSIM

WinIQSIM es un programa de la empresa Rhode & Schwarz que se utiliza para analizar diferentes modulaciones digitales y los posibles efectos que se producen debido a los componentes de los propios sistemas de comunicación y a los producidos por el canal, algunos de estos efectos son:

- Desfasaje en las componentes IQ
- Ruido de fase y AWGN
- Aparición de bandas laterales
- Distorsión producida por los amplificadores
- Multitrayecto
- Señales interferentes

Permite distinguir las componentes IQ, ver la constelación, diagramas de ojo y ver su espectro. También permite trabajar con sistemas con múltiples portadoras, CDMA2000, IEEE 802.11 WLAN, WCDMA, entre otros. Es un programa muy antiguo que lleva sin soporte desde Windows 2000. El funcionamiento es sencillo, pero visualmente no es agradable. Este programa no es compatible con dispositivos SDR.

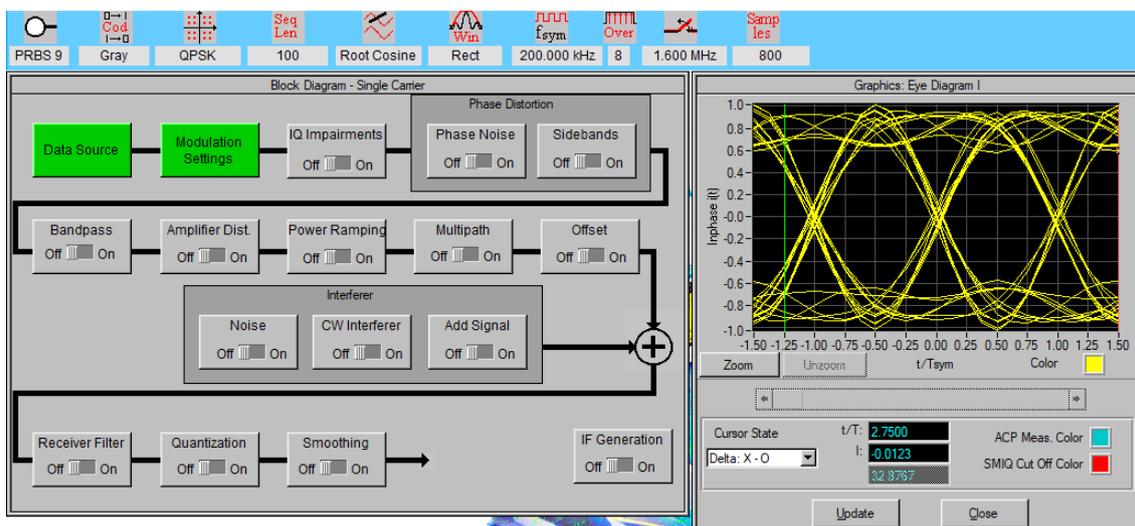


Figura 7. Captura de pantalla de WinIQSIM

En la búsqueda del nuevo *software* se priorizan aquellos programas que puedan integrar las funcionalidades de procesamiento digital de la señal que ofrecen MATLAB y WinIQSIM junto con la capacidad para funcionar utilizando un SDR. Existen multitud de opciones disponibles en el mercado para utilizar junto a un SDR. La mayoría de los programas solo permiten trabajar junto al SDR y no permiten realizar procesamiento digital de la señal en el PC, como SDRangel. En cambio, hay otras opciones, como GNU Radio o Simulink, que permiten realizar las dos funciones.

2.8.3 GNU Radio

Anteriormente se definió de forma breve el programa GNU Radio. En este apartado se profundiza en el funcionamiento del programa y sus características.

GNU Radio es un programa gratuito y de código abierto escrito en lenguaje C++ que es muy utilizado para procesamiento digital de la señal y, sobre todo, para implementar y trabajar con redes de radiocomunicación. En GNU Radio se puede recibir y transmitir datos utilizando *hardware* de todo tipo.

La gran mayoría de dispositivos SDR, por no decir todos, tienen soporte de GNU Radio. Es bastante raro encontrar un dispositivo que no tenga soporte de este programa ya que los propios fabricantes se encargan de proporcionar los bloques necesarios para su manejo y, de no ser así, es la comunidad la que acaba desarrollando estos bloques.

Las señales con las que trabaja GNU Radio son señales digitales. Se utilizan señales en envolvente compleja porque las señales de salida de los receptores y las señales de entrada de los transmisores son de este tipo.

Lo más frecuente es utilizar este programa en distribuciones de Linux, como Ubuntu o Fedora. También tiene una versión para Raspberry Pi y otra para MacOS X. Existe una versión para Windows, pero no tiene soporte oficial por parte de los desarrolladores y su instalación es un poco complicada.

Se puede utilizar GNU Radio mediante la consola de comandos, y también mediante una interfaz conocida como *GNU Radio Companion* (GRC), muy parecida a Simulink. La forma de trabajar es mediante bloques que implementan distintas funcionalidades. Existen más de 470 bloques diferentes en la versión básica. Incluye multitud de bloques de:

- Filtros
- Operaciones matemáticas
- Códigos de canal
- Elementos de sincronización
- Ecuadores
- Moduladores y demoduladores
- Codificadores y decodificadores
- Osciloscopio
- Visor de constelación
- Analizador de espectro

En la interfaz GRC, Figura 8, se pueden distinguir cinco paneles diferentes: en rojo la barra de herramientas; en azul el área de trabajo o *workspace*; en gris la librería en la que se encuentran los bloques; en verde un terminal de comandos y en amarillo las variables que se están utilizando en el proyecto.

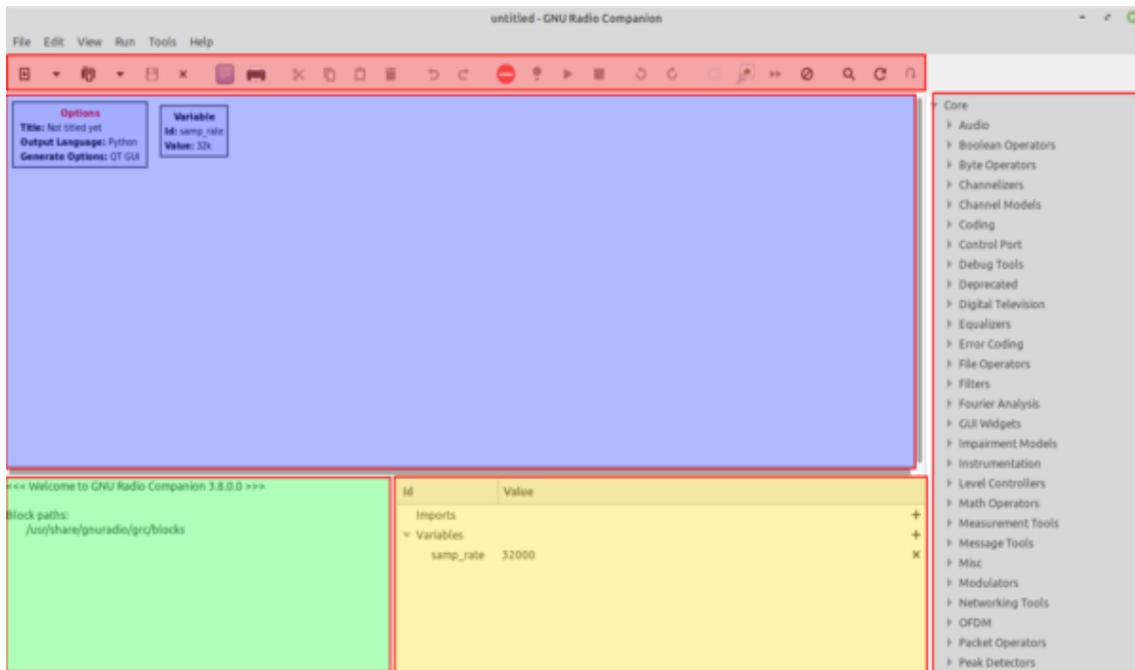


Figura 8. Interfaz de GNU Radio Companion

La gran comunidad que rodea a este programa ha creado multitud de ejemplos que facilitan el aprendizaje y ahorran trabajo a nuevos proyectos. Algunos proyectos son:

- Transmisor DVB-S2
- Decodificadores para satélites
- Receptores y demoduladores de tramas los satélites de la constelación Iridium
- Transceptor FM
- Transceptor IEEE 802.15.4 (ZigBee)
- *Toolbox* para radar
- Receptor para LoRa usando SDR
- Transmisión y recepción inalámbrica de video de drones en 5.8 GHz
- Recepción de señales ADS-B
- Receptores de LTE
- Transmisor DRM

Otra funcionalidad de GNU Radio es compartir de forma rápida y automatizada los datos, recibidos o generados, a programas como MATLAB u Octave utilizando scripts ya implementados. Además, permite utilizar bloques de código en lenguaje Python, aunque no son tan rápidos como los bloques en C/C++.

GNU Radio y Simulink tienen un funcionamiento y apariencia muy similar. Son bastante intuitivos. Ambos utilizan diagramas de flujo y bloques para implementar las diferentes funcionalidades.

En la Figura 9 y en la Figura 10 se implementa un demodulador FM utilizando ambos programas. Como puede observarse, son bastante similares. La diferencia es que en GNU Radio se ven los parámetros en el propio bloque. En cambio, en Simulink hay que pulsar dos veces sobre el bloque para abrir el cuadro de diálogo y desde ahí modificar los parámetros.

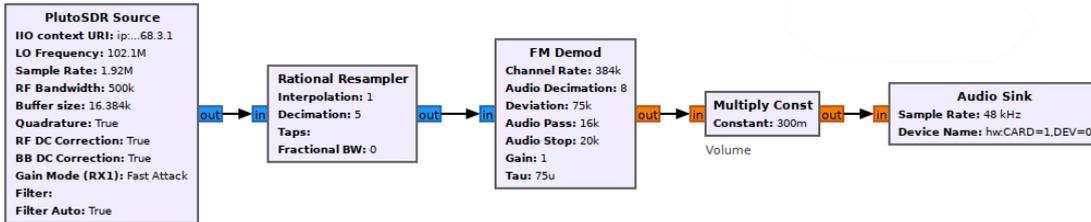


Figura 9. Demodulador FM utilizando GNU Radio y ADALM-Pluto

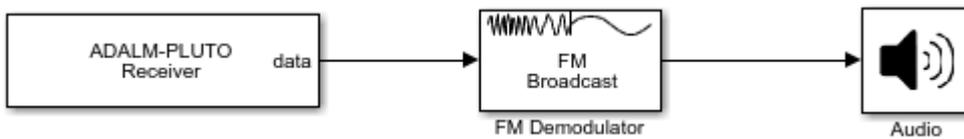


Figura 10. Demodulador FM utilizando Simulink y ADALM-Pluto

El programa que se utilizará en este proyecto de fin de grado será Simulink, que será explicado más adelante, porque el departamento está acostumbrado a trabajar con herramientas MATLAB y también porque la instalación de GNU Radio en Windows no es muy amigable.

2.9 Utilización de SDR en enseñanza

Existen diferentes proyectos que utilizan los SDR como método de enseñanza de diferentes conceptos relacionados con las telecomunicaciones. Desde explicar conceptos generales de comunicaciones a otros más específicos orientados a universitarios de últimos cursos. A continuación, se explican algunos de estos proyectos:

El primero de estos proyectos es de la California State University Northridge [21] (CSUN). En este proyecto se busca demostrar a alumnos de segundo y tercer curso diferentes conceptos generales de comunicaciones y de procesamiento de señal. La implementación del laboratorio se ha hecho utilizando USRP con GNU Radio y *scripts* de PHYTON corriendo sobre Ubuntu. Las funcionalidades son múltiples: tienen la posibilidad de analizar señales en dominio temporal y en frecuencia, también utilizan los SDR para recibir y transmitir señales de diferentes modulaciones. Principalmente trabajan con sistemas que resultan familiares a los alumnos como son la televisión y la radio comercial AM/FM. Los resultados de este laboratorio son, en general, satisfactorios ya que existe un gran interés por parte de los alumnos en este curso.

Otro proyecto es el de la Universidad de Ciencia y Tecnología de Huazhong [22]. En esta ocasión se explica el problema que tenían los profesores de esta universidad para actualizar el laboratorio. Quieren utilizar sistemas SDR y su intención inicial era utilizar el dispositivo USRP-2920, que cuesta más de 3000\$. Debido al elevado precio, finalmente se decidieron por el ADALM-Pluto, por unos 149\$, tras realizar una comparativa entre ambos dispositivos. La prueba que realizaron es probar cómo se comporta en una comunicación punto a punto, utilizando MATLAB con ADALM-Pluto

y LabVIEW con el USRP-2920. La conclusión a la que llegan es que el ADALM-Pluto es la mejor opción para su laboratorio, porque es más asequible para los alumnos, es más simple y funciona con MATLAB, que es muy popular en su universidad. En cuanto a la calidad de las comunicaciones consideran que no existe tanta diferencia para el uso que van a darle. Los resultados de este experimento han sido de gran ayuda para tomar la decisión de utilizar el ADALM-Pluto como SDR de este proyecto fin de grado.

Un proyecto muy similar al que se desarrolla en este proyecto fin de grado es el curso que realiza la Universidad de Strathclyde junto a MathWorks para los alumnos de cuarto curso [23]. En este proyecto el dispositivo elegido es el RTL-SDR que solo puede usarse como receptor. Para generar las señales transmitidas utilizan un dispositivo Zynq SDR o un USRP. El programa que utilizan es Simulink. Los resultados de aprendizaje de este laboratorio son ver cómo funciona y cómo está distribuido el espectro radioeléctrico y ver cómo afecta el canal de comunicación a la señal. Un problema que se encontraron al utilizar este dispositivo es que su ancho de banda de RF es pequeño, 2.8 MHz, por lo que no pudieron ver el espectro completo de UMTS o LTE. Los creadores de este proyecto aprovecharon la ocasión para publicar un libro gratuito [24] que incluye distintos diseños y formas de trabajar con SDR y Simulink.

El último proyecto analizado es el de la Universidad de RWTH Aachen [25]. Tienen cuatro tipos de laboratorios diferenciados según su nivel de dificultad o tipo de estudiante. A continuación, se describen los dos más interesantes:

El curso *Exploring Modular Design of SDR Transceivers* está destinado a estudiantes de segundo curso de sistemas de comunicación. En este se imparten los conceptos básicos de SDR y se explican los conceptos de modulación y demodulación de forma práctica y se experimenta con distintos tipos de modulaciones básicas como QPSK o AM/FM. Se explican conceptos como la sincronización entre símbolos, fase, frecuencia y entre tramas. También se explican e implementan los códigos CRC y de Hamming.

El más avanzado es *Wireless Communications: Software Radio Implementations* y está destinado a alumnos de cuarto curso incluso ya graduados. Comienza con una pequeña introducción a las arquitecturas SDR, después con MATLAB y Simulink aplicado a SDR, lo siguiente es explicar la capa física junto a herramientas de procesamiento digital de la señal y, por último, explican la capa de red protocolos MAC en SDR. Para terminar el laboratorio los alumnos deben realizar un proyecto que debe tratar sobre OFDM, MIMO, protocolos MAC cognitivos o sobre reducción de interferencias y adaptación al canal.

2.10 Conocimientos y trabajos relacionados

En la realización de este proyecto fin de grado ha sido necesario utilizar la gran mayoría de conceptos estudiados durante la carrera desde conceptos sobre teoría de la señal como teorema de muestreo, diseño de filtros, diseño de moduladores y demoduladores, teoría de cuantificación, códigos de canal, así como los diferentes efectos que tiene el canal sobre la señal.

Ha sido necesario aprender a utilizar Simulink y todas las herramientas o *toolbox* que se han utilizado para diseñar el programa de forma autodidacta. Para aprender estos

conceptos ha sido de gran ayuda la documentación y manuales de este programa, así como la información de foros como StackOverflow, DSP Related, MATLAB Central y en la Wiki de Analog Devices.

Otro concepto aprendido durante la realización de este proyecto fin de grado ha sido el funcionamiento de la tecnología SDR, la cual apenas se conoce entre los alumnos de la escuela ya que solo se utiliza en una práctica de laboratorio de la asignatura Sistemas de Telecomunicación. Para aprender todo sobre esta tecnología se han utilizado diferentes libros como [7] y [24].

En cuanto a otros proyectos fin de carrera similares a este, la realidad es que no hay muchos realizados utilizando esta tecnología. El más parecido a este es el proyecto de Mario Mateo Hernández, “Diseño de un sistema basado en radio definida por software para realizar medidas de canal radio” [26]. La diferencia es que en este se hace principal hincapié en el modelado del canal y los modelos de *path-loss*. Este proyecto está más orientado a la planificación de redes de radiocomunicación, como radioenlaces o estaciones base para telefonía. Se incluyen más de seis tipos de modelos de *path-loss* y de modelo de rayos. Además, se tienen en cuenta las ganancias de las antenas transmisora y receptora, así como la altura a la que se encuentran. Se utiliza la herramienta App Designer de MATLAB. En cambio, este proyecto tiene muy pocos tipos de modulaciones ya que no dispone de modulaciones analógicas y las digitales que tiene son solo QAM y PSK.

Otro proyecto fin de grado utilizando SDR es el de Guillermo Bartolomé Herrador, “Diseño de una estación remota de comunicaciones basada en radio software” [27] que se centra en la utilización de SDR junto a una Raspberry Pi para crear una estación remota de comunicaciones tipo PMR446, más conocida como *walkie talkie*. El proyecto de Guillermo tiene dos partes: el centro de control y la estación remota. El centro de control se conecta con la estación remota mediante un cable Ethernet conectado a la Raspberry Pi. La Raspberry Pi se utiliza para comunicarse con el centro de control utilizando protocolos UDP o TCP, y para procesado de la señal recibida y transmitida por el SDR, junto al programa GNU Radio. En este proyecto se utiliza siempre el mismo tipo de modulación que es FM de banda estrecha. Existen grandes diferencias entre el proyecto de Guillermo y el que se desarrolla en estas páginas, por ejemplo: las señales radio que se envían y se reciben utilizando el SDR son siempre del mismo tipo y no pueden modificarse, además, se centra sobre todo en la parte de protocolos de comunicaciones en la capa de transporte IP y no en la interfaz radio.

3. Especificaciones y restricciones

La elección de los componentes *software* y *hardware* viene condicionada especialmente por el bajo presupuesto y la necesidad de que el *software* sea sencillo de utilizar. En los siguientes apartados se explica el motivo de la elección de estos componentes, así como las especificaciones de los bloques y requisitos.

3.1 Especificaciones

3.1.1 Rango de frecuencias

En España existen diferentes bandas de frecuencias destinadas a aplicaciones industriales, científicas y médicas (ICM). Estas frecuencias pueden utilizarse también por los radioaficionados [28]. Algunas de las bandas de frecuencias ICM son:

- 433.050 – 434.790 MHz
- 2400 – 2450 MHz
- 5725 – 5800 MHz
- 24 – 24.25 GHz

Se utilizará la banda de frecuencias de 433.05 - 434.79 MHz. Esta banda de frecuencias es la que mejor se adapta a las características del ADALM-Pluto y además no se encuentra demasiado saturada, cosa que sí ocurre en la banda de 2400 MHz.

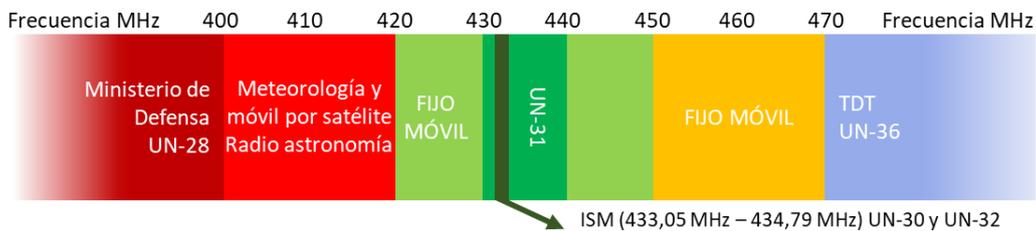


Figura 11. Organización del espectro en España entre 400 y 470 MHz

3.1.2 Señales analógicas

Nuestro prototipo debe generar diferentes tipos de señales analógicas de utilidad en el ámbito de la enseñanza. Estas señales analógicas solo pueden utilizarse con las modulaciones analógicas, que se analizarán más adelante. Las señales analógicas que se pueden generar se muestran en la Tabla 3 junto a sus parámetros.

Tabla 3. Parámetros del generador de señales analógicas

Tipo de generador	Parámetros
Tonos 1 y 2	Amplitud de cada tono A_1, A_2 . La salida está normalizada.
	Frecuencia de cada tono en Hz f_1, f_2
	Fase inicial en rad φ_1, φ_2
Aleatoria Uniforme	<i>Seed</i> : semilla generadora.
	Amplitud normalizada
Gaussiana	<i>Seed</i> : semilla generadora.
	Amplitud normalizada
	Media = 0; Desviación típica = 1
Fichero de audio	Ruta del fichero.

3.1.3 Cuantificador

Las señales analógicas pueden pasar por un cuantificador en el que se toman los valores de amplitud de estas señales de entrada. Se deben diseñar cuantificadores de tipo uniforme y de tipo logarítmico. En la Tabla 4 se muestran los parámetros que pueden modificarse para cada uno de ellos.

Tabla 4. Parámetros de los cuantificadores

Tipo de cuantificador	Parámetros
Uniforme: Midrise o Midtreat	Intervalo de cuantificación
Ley A	Valor de A
Ley Mu	Valor de Mu (μ)
Sin Cuantificador	Sin parámetros

3.1.4 Señales digitales

En este bloque deben generarse seis tipos de señales digitales las cuales se muestran en la Tabla 5 junto a sus parámetros. La longitud de estas señales se pone de forma automática según el número de símbolos a la salida del modulador y de si se usa o no un codificador de canal. Para la sincronización de las tramas se utiliza una cabecera con un código de tipo Barker que se incluye después de la codificación de canal.

Tabla 5. Parámetros del generador de señales digitales

Tipo de generador	Parámetros
Todo 1 / Todo 0	Sin parámetros
Secuencia (<i>pattern</i>)	Introducir una secuencia personalizada
PRBS	Coeficientes 7, 9, 11, 15, 16, 20, 21, 23
Bit aleatorio uniforme	Sin parámetros
Fichero binario	Variable que contiene el fichero en binario

3.1.5 Codificación de canal

Las señales digitales pueden pasar por el bloque de codificación para añadir redundancia y detectar y corregir errores. También puede añadirse entrelazado utilizando el *Random Interleaver* para evitar ráfagas de bits erróneos. Se han implementado dos tipos de codificadores de canal. En la Tabla 6 se muestran los parámetros de ambos codificadores.

Tabla 6. Parámetros del codificador

Tipo de codificador	Parámetros
BCH	M: entero entre 3 y 16
	R: redundancia añadida por el código
Reed-Solomon	K: longitud del mensaje de entrada
	N: longitud del mensaje de salida
Random Interleaver	Seed: semilla generadora del entrelazado

3.1.6 Modulación

El bloque de modulación es el bloque más complejo del sistema. En este bloque se implementan las diferentes modulaciones, tanto analógicas como digitales, y también los distintos filtros.

Las modulaciones analógicas tienen un parámetro en común que es la frecuencia de muestreo f_s . También puede añadirse un filtro de pre-énfasis a este tipo de modulación. En la Tabla 7 se muestran los parámetros individuales de cada modulación analógica.

Tabla 7. Parámetros de las modulaciones analógicas

Tipo de modulación	Parámetros
AM	m : índice de modulación
	φ : fase en radianes
DSB	φ : fase en radianes
SSB	Banda superior o inferior
	φ : fase en radianes
FM	Δf : desviación de frecuencia en Hz

Para las modulaciones digitales hay cuatro parámetros generales que comparten todas ellas, y son:

- Orden de la modulación M .
- Número de símbolos a la salida del modulador.
- Régimen simbólico a la salida del modulador.
- Factor de sobremuestreo o de interpolación.

En la Tabla 8 se muestran los parámetros individuales de las modulaciones digitales.

Tabla 8. Parámetros de las modulaciones digitales

Tipo de modulación	Parámetros
ASK, PSK, MSK, QAM	φ : fase en radianes
FSK	φ : fase en radianes
	Índice de modulación

En el proyecto se ha implementado un modulador IQ que convierte la señal paso bajo a señal paso banda. Hay dos bloques diferentes: uno para las modulaciones analógicas y otro para las modulaciones digitales. Están identificados con su correspondiente nombre y se encuentran dentro del bloque *Other Blocks*. Los parámetros de este modulador se muestran en la Tabla 9.

Tabla 9. Parámetros del modulador IQ

	Parámetros
Modulador IQ	A_c : amplitud de salida de la señal modulada
	f_c : frecuencia del modulador IQ

La amplitud de salida de la señal modulada es la que modifica la potencia de salida de todo el sistema.

Una vez modulada la señal puede utilizarse el bloque de OFDM para observar su espectro y modificar los parámetros mostrados en la Tabla 10. Los pilotos llevan una señal aleatoria modulada en BPSK.

Tabla 10. Parámetros de OFDM

	Parámetros
OFDM	Longitud de la FFT
	Bandas de guarda
	Longitud del prefijo cíclico
	Número de símbolos OFDM
	Insertar nulo en DC
	Posiciones de los pilotos
	Ganancia de los pilotos
	Enventanado coseno alzado

Además de los moduladores, el bloque modulador también incluye los filtros típicos de comunicaciones y pueden modificarse los parámetros de la Tabla 11.

Tabla 11. Parámetros de los filtros

Tipo de filtro	Parámetros
Paso Bajo	Frecuencia de la banda de paso en Hz
	Frecuencia de corte en Hz
	Máximo rizado en banda de paso en dB
	Atenuación mínima de la banda de corte en dB
Raíz de Coseno Alzado, Coseno Alzado	α : factor de <i>roll-off</i>
	<i>Span</i> del filtro en número de símbolos
	Ganancia del filtro
Gaussiano	<i>BT</i> : ancho de banda -3dB por tiempo de símbolo
	<i>Span</i> del filtro en número de símbolos

3.1.7 Efectos del canal

Después de la modulación, la señal pasa por el bloque de efectos (“impairments”) producidos por el canal. Estos efectos y sus parámetros se recogen en la Tabla 12.

Tabla 12. Parámetros de los efectos del canal

Tipo de efecto	Parámetros
AWGN	E_b/N_0 en dB
Distorsión lineal	Atenuación en dB
	Desfasaje en grados
Distorsión no lineal	k_2 y k_3
Onda continua interferente	C/I en dB
	Frecuencia en Hz
	φ : fase en radianes
Multitrayecto: 4 rayos	Relación de amplitud respecto al rayo directo en dB
	Retardo de tiempo con el que llega el rayo
	Desfase en grados

3.2 Requisitos

Todas las pruebas realizadas en este proyecto fin de grado han sido realizadas en un ordenador con las siguientes características:

- Procesador Intel i7-4500U a 2.7 GHz de dos núcleos y cuatro hilos.
- 16 GB de memoria RAM DDR3 a 1600 MHz.
- SSD de 500 GB.

El *hardware* de este ordenador es relativamente antiguo (2013) por lo que el programa debería correr sin problema en ordenadores más modernos. A continuación, se describen los requisitos para que funcione el programa de forma correcta.

3.2.1 Capacidad

Los requisitos mínimos para la utilización de MATLAB y Simulink junto con las *toolbox* en este proyecto son:

- Espacio de almacenamiento de 18 GB, recomendable tener SSD
- Memoria RAM 6 GB, recomendable 8 GB

3.2.2 Procesado

Las especificaciones del ADALM-Pluto indican que existe una limitación en cuanto al ancho de banda disponible para hacer retransmisiones en directo mediante un ordenador utilizando la interfaz USB 2.0.

El estándar USB 2.0 [29] permite una velocidad teórica de transmisión máxima de 480 Mbit/s (60 MB/s). En pruebas reales, la velocidad máxima que puede conseguirse es de 280 Mbit/s (35 MB/s). Además, el SDR requiere de un número mínimo de muestras para poder mantener la transmisión en tiempo real. Por tanto, tenemos una limitación en el ancho de banda máximo disponible y en la velocidad de transmisión, cuyos valores se han obtenido mediante pruebas experimentales:

- Para modulaciones y fuentes digitales: el régimen simbólico máximo es de 400 kbauds y el mínimo es de 20 kbauds.
- Para modulaciones y fuentes analógicas: la frecuencia de muestreo máxima es de 2 MHz y la mínima de 50 kHz.

Esta limitación puede observarse utilizando un analizador de espectros, lo que sucede es que el SDR no es capaz de tener todas las tramas a la vez y se envían por ráfagas.

Respecto a las frecuencias de emisión no existe ningún limitante ya que permite elegir un amplio conjunto de frecuencias.

3.2.3 Fiabilidad

El sistema está preparado para funcionar de forma correcta si se utiliza en un ordenador potente o si solo se ejecuta este programa. Es posible que se llene el buffer del dispositivo Adalm-Pluto tras un uso continuado y requiera reiniciar la transmisión. En

caso de utilizar el programa para análisis de las componentes IQ o medida de la BER no se han encontrado problemas.

Para utilizar el modo de transmisión se recomienda cerrar las tareas en segundo plano para quitar carga al procesador, ya que es posible que se pierdan algunas muestras de la señal transmitida al SDR. Para comprobar si se han perdido muestras, se ha implementado en el programa un indicador de muestras perdidas. El número que aparece indica las muestras perdidas.

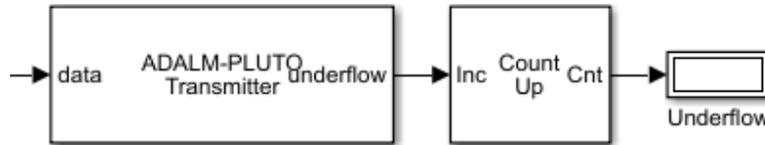


Figura 12. Detector de muestras perdidas

3.2.4 Disponibilidad

Este programa estará disponible en el fichero adjunto a este proyecto. Las instalaciones necesarias son: MATLAB, Simulink y las correspondientes *toolbox* que son: *Communications Toolbox*, *DSP System Toolbox* y *SerDes Toolbox*.

3.2.5 Accesibilidad

El programa se considera accesible para todas las personas excepto para personas ciegas ya que no puede utilizarse mediante dictado por voz.

3.2.6 Usabilidad

Para manejar este programa hay que saber utilizar Simulink de forma básica. Es necesario entender qué se está haciendo en cada bloque para conseguir aprovechar al máximo el programa, por tanto, es necesario tener sobre el funcionamiento general de los sistemas de telecomunicación.

3.2.7 Coste

Para cumplir con estos requisitos *hardware* y *software* es necesario realizar una pequeña inversión en la que se incluye todo lo necesario para ejecutar el programa con normalidad. El total de la inversión son unos 800€, desglosados en:

- SDR ADALM-Pluto por 160 €
- Ordenador personal básico (procesador de 4 núcleos + 8 GB RAM) por 500 €
- Periféricos: monitor, teclado y ratón: 130 €
- MATLAB y Simulink: gratuito con la licencia universitaria.

Más detalles pueden encontrarse en el capítulo 7, donde se incluye el presupuesto total general del proyecto.

3.3 Herramientas necesarias

3.3.1 MATLAB y Simulink

MATLAB es un programa de cálculo y programación con lenguaje propio. Este programa es muy utilizado para la representación de datos y funciones, cálculo matricial, implementación de algoritmos, comunicación con dispositivos hardware y con programas en otros lenguajes.

Simulink es un entorno de diseño y simulación de sistemas incluido en MATLAB, en el que no es necesario escribir código. Simulink permite modelar sistemas de gran complejidad, realizar multitud de pruebas de sistemas forma rápida e incluso generar código automáticamente en diferentes lenguajes como C y HDL.

La utilización del ecosistema MATLAB-Simulink permite aprovechar al máximo las ventajas que este ofrece. Una de ellas es la posibilidad de aumentar las capacidades de ambos entornos añadiendo paquetes de herramientas o *toolboxes* que pueden ser proporcionados por la empresa que desarrolla MATLAB y Simulink, MathWorks, o por otros desarrolladores.

Las *toolboxes* permiten obtener compatibilidad con *hardware* específico, así como utilizar funciones y *System Objects* ya implementados para acelerar el flujo de trabajo y centrarse solo en las dificultades que presenta el diseño del sistema.

Los *System Objects* [30] son objetos en lenguaje MATLAB que sirven para implementar y simular sistemas dinámicos con entradas que cambian con el tiempo, sistemas con diferentes estados y/o sistemas en tiempo real. Estos sistemas son muy útiles para sistemas de comunicaciones y de procesamiento digital de la señal.

La principal diferencia entre las funciones tradicionales y los *System Object* es que las funciones requieren un código más complejo porque necesitan definir la longitud de los datos que se procesan mientras que en los *System Object* esto no es necesario definirlo porque se realiza de forma dinámica.

Cada uno de los bloques ha sido programado mediante *System Objects* o bien ha sido creado utilizando los bloques proporcionados por las *toolboxes*. Cada uno de estos bloques tiene un cuadro de diálogo o GUI propio que ha sido creado usando el editor de máscaras de Simulink [31].

Durante el desarrollo del proyecto ha sido necesaria la utilización de estas máscaras para mostrar al usuario únicamente los parámetros que puede modificar al utilizar el programa. La utilidad de estas máscaras es que permiten interconectar los parámetros de los diferentes componentes internos de cada bloque, así como modificar la apariencia del cuadro de diálogo dependiendo de la opción elegida.

Estos son los motivos por los cuales el combo MATLAB-Simulink es una gran elección para realizar este proyecto fin de grado. En el siguiente apartado se explican las diferentes *toolbox* utilizadas en el proyecto y el motivo de su elección.

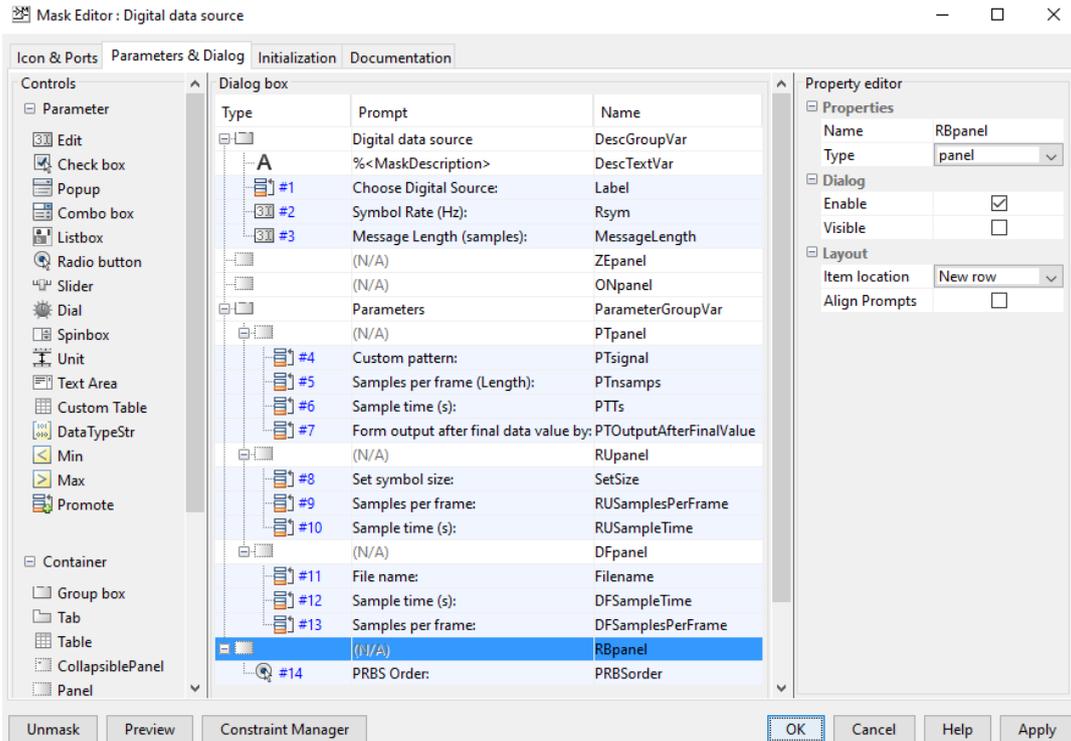


Figura 13. Editor de Máscaras de Simulink para el bloque de señales digitales

3.3.2 Simulink Toolboxes

La primera *toolbox* del proyecto es *Communications Toolbox* [32] que añade una gran cantidad de elementos de telecomunicaciones como pueden ser:

- Fuentes de información y generadores de señal
- Visores de constelación y de diagrama de ojo
- Cuantificadores
- Códigos de detección y corrección de errores
- Moduladores y demoduladores
- Filtros de comunicaciones
- Pérdidas del medio

Para que funcione esta *toolbox*, es necesario instalar también el paquete de soporte para ADALM-Pluto [33].

La segunda *toolbox* es *DSP System Toolbox* [34]. Como su nombre indica, sirve para procesado digital de la señal. Necesaria para introducir filtros, añadir archivos como fuente de información, interpolación, realizar cambios de frecuencia de muestreo, añadir retardo, etc. También incluye un analizador de espectro y un osciloscopio.

La tercera y última *toolbox* se llama *SerDesToolbox* [35] que se utilizará para generar señales del tipo PRBS.

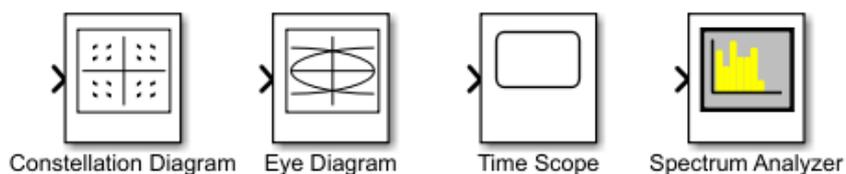


Figura 14. Cuatro bloques introducidos por las toolbox

4. Descripción de la solución propuesta

La solución propuesta es el diseño de un programa, utilizando Simulink, que abarca las necesidades del laboratorio de Teoría de la Comunicación de la carrera. El programa tiene tres modos distintos de funcionamiento seleccionables de forma sencilla mediante un conmutador:

- Transmisión de señales utilizando el SDR.
- Análisis de las componentes IQ: diagrama de ojo, analizador de espectros y osciloscopio y constelación.
- Cálculo de la BER.

El sistema está diseñado de tal forma que se pueda elegir entre señales analógicas y señales digitales, utilizando otro conmutador. El programa está formado por diferentes bloques que implementan los diferentes componentes. A continuación, se detallan las características de los bloques de este proyecto.

4.1 Bloque de señales analógicas

En este bloque se generan cuatro tipos de señales analógicas las cuales se analizan en los siguientes apartados:

4.1.1 Tonos 1 y 2

Se generan dos sinusoides x_i que siguen la siguiente expresión:

$$x_1(t) = A_1 \sin(\omega t + \varphi_1) \quad (5)$$

$$x_2(t) = A_2 \sin(\omega t + \varphi_2) \quad (6)$$

La salida del sistema es la suma de ambas expresiones normalizadas en amplitud para el caso de coincidencia de fases. Es decir, cuando $\varphi_1 = \varphi_2$:

$$y(t) = [A_1 \sin(\omega t + \varphi_1) + A_2 \sin(\omega t + \varphi_2)] \cdot \frac{1}{A_1 + A_2} \quad (7)$$

4.1.2 Aleatoria Uniforme

Se generan muestras de una distribución aleatoria uniforme entre -1 y 1 a partir de un número generador o semilla. La función de probabilidad de una variable aleatoria uniforme es la de la expresión (8), donde $a = -1$ y $b = 1$:

$$f(x|a, b) \begin{cases} \left(\frac{1}{b-a}\right); & a \leq x \leq b \\ 0 & ; resto \end{cases} \quad (8)$$

4.1.3 Gaussiana

Se genera una función gaussiana de media $\mu = 0$ y desviación típica $\sigma = 1$. La función densidad de probabilidad de la gaussiana que se implementa en el proyecto es:

$$f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \rightarrow f(x|0,1) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \text{ para } x \in \mathbb{R} \quad (9)$$

4.2 Cuantificador

El bloque de cuantificación implementa tres tipos de cuantificadores. La cuantificación consiste en redondear los valores de amplitud de una señal analógica respecto a un determinado valor de referencia. Los cuantificadores introducen un error de cuantificación que depende del nivel de cuantificación. En general, los cuantificadores no uniformes son mejores que los uniformes, ya que utilizan los niveles apropiados para el tipo de señal que se les presente a la entrada. Sin embargo, los cuantificadores no uniformes necesitan un compresor logarítmico antes de cuantificar para reducir el ruido de cuantificación.

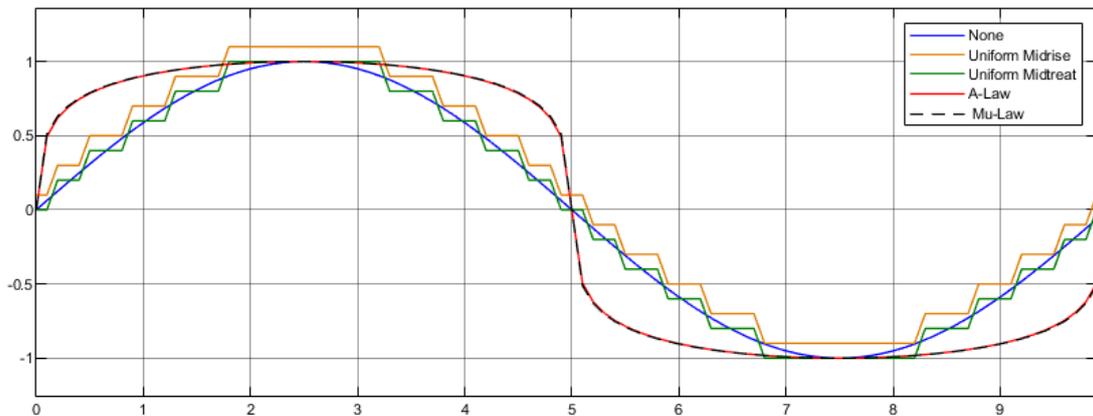


Figura 15. Efecto de los cuantificadores sobre una senoide

4.2.1 Uniforme

Los cuantificadores uniformes tienen un intervalo de cuantificación fijo Δ y equiespaciado. También se les conoce como cuantificadores lineales. La implementación en este proyecto se realiza utilizando el bloque *Quantizer* de Simulink. Para el cuantificador uniforme de tipo *midrise* se utiliza la ecuación (10) y para el de tipo *midtread* se utiliza la ecuación (11).

$$y = \Delta * \text{sign}(x) * \text{round}(x/\Delta) + \Delta/2 \quad (10)$$

$$y = \Delta * \text{sign}(x) * \text{round}(x/\Delta) \quad (11)$$

4.2.2 Ley A

La cuantificación Ley A es de tipo logarítmico y, por tanto, no uniforme. Su expresión sigue la ecuación (12). La cuantificación Ley A es una de las dos versiones disponibles en el estándar G.711 [36]. Se utiliza en sistemas de comunicaciones europeos, especialmente en telefonía, para mejorar el rango dinámico de las señales analógicas. Para la función inversa se utiliza la expresión (13). En ambas ecuaciones se utiliza $A = 87.6$.

$$F(x) = \text{sign}(x) \begin{cases} \frac{A|x|}{1+\ln(A)}, & |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \leq |x| \leq 1 \end{cases} \quad (12)$$

$$F^{-1}(y) = \text{sign}(y) \begin{cases} \frac{|y|(1+\ln A)}{A}, & |y| < \frac{1}{1+\ln A} \\ \frac{\exp(|y|(1+\ln A)-1)}{A}, & \frac{1}{1+\ln A} \leq |y| < 1 \end{cases} \quad (13)$$

4.2.3 Ley Mu

La cuantificación tipo Ley Mu es logarítmica, luego no uniforme y sigue la ecuación (14). Es la otra versión disponible del estándar G.711 [36] y se utiliza en sistemas de comunicaciones en América del Norte y Japón. La función inversa del cuantificador Ley Mu es la de la ecuación (15). El valor de $\mu = 255$.

$$F(x) = \text{sign}(x) \frac{\ln(1+\mu|x|)}{\ln(1+\mu)}, \quad -1 \leq x \leq 1 \quad (14)$$

$$F^{-1}(y) = \text{sign}(y) \left(\frac{1}{\mu}\right) \left((1 + \mu)^{|y|} - 1\right), \quad -1 \leq y \leq 1 \quad (15)$$

4.3 Señales digitales

Los generadores de todo 1, todo 0, y bit aleatorio uniforme generan señales de este tipo y se utilizan bloques básicos de Simulink para implementarlos.

4.3.1 Secuencia personalizada

La secuencia o patrón personalizado pide al usuario que introduzca un vector de bits y permite que el usuario elija si la secuencia se transmite en bucle “*Cyclic repetition*”, se mantiene el último bit “*Holding final value*” o si se envían ceros tras finalizar la secuencia “*Setting to zero*”.

El usuario también puede definir en MATLAB una variable global como un vector con valores binarios y utilizar esa variable como secuencia poniendo en “*Custom pattern*” el nombre de la variable.

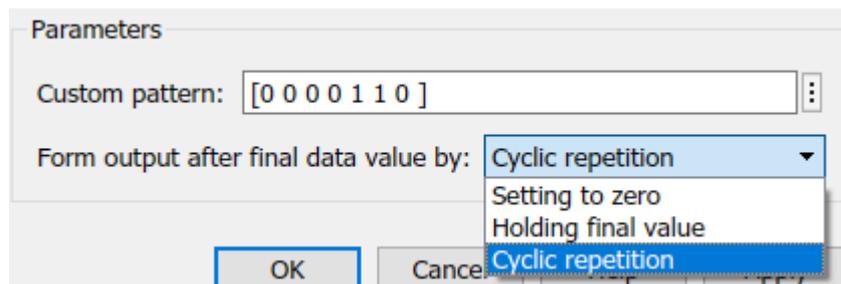


Figura 16. Interfaz del bloque de secuencia binaria personalizada

4.3.2 PRBS

Los códigos PRBS son secuencias pseudoaleatorias que se comportan a nivel estadístico como una secuencia aleatoria. Estas secuencias son periódicas, donde la longitud máxima de cada ciclo es $N = 2^n - 1$ siendo n el orden de la secuencia.

Estos códigos son muy útiles para probar el correcto funcionamiento de muchos sistemas. En este caso se utilizará como una fuente de información. Para la implementación de los códigos se utiliza la función “prbs” de *SerDesToolbox*.

Los códigos que se utilizan en este proyecto son los que se muestran en la Tabla 13 junto a sus polinomios generadores obtenidos de [37].

Tabla 13. Polinomios generadores de PRBS

Orden n del PRBS	Polinomio generador
7	$x^7 + x^6 + 1$
9	$x^9 + x^5 + 1$
11	$x^{11} + x^9 + 1$
15	$x^{15} + x^{14} + 1$
16	$x^{16} + x^{14} + x^{13} + x^{11} + 1$
20	$x^{20} + x^{17} + 1$
21	$x^{21} + x^{19} + 1$
23	$x^{23} + x^{18} + 1$

4.3.3 Archivo binario

Para utilizar un archivo binario es necesario cargarlo en el espacio de trabajo de MATLAB. Otra condición es que el archivo sea binario, es decir, ceros y unos (formato .bin). Como no es usual tener archivos .bin se ha implementado una función con nombre `jpg2bit.m` que pasa de formato .jpg a .bin y otra que convierte de nuevo a .jpg cuyo nombre es `bit2jpg.m`.

A la función `jpg2bit.m` hay que pasarle la imagen que se desea convertir a bits y devuelve la imagen en bits y las dimensiones de la imagen.

Los argumentos de entrada de la función `bit2jpg.m` son: la señal y que sale del proyecto de Simulink y las dimensiones de la imagen a reconstruir. En el manual de usuario se muestra un ejemplo de esta función del programa.

4.4 Codificación de canal

La codificación de canal es un proceso por el cual se añade redundancia a los datos transmitidos con el objetivo de reducir, detectar y corregir los errores que puedan producirse en la transmisión [38]. La inclusión de redundancia hace que aumente el régimen binario y, por tanto, el ancho de banda. Esto provoca que la eficiencia espectral de la transmisión sea baja en situaciones de alta SNR, pero para situaciones con baja SNR es muy conveniente porque mantiene la BER en valores adecuados.

Los códigos implementados son códigos de bloque y son capaces de corregir un número finito de errores sin necesidad de retransmisión. Los bits de paridad se añaden a bloques de la secuencia de entrada para formar palabras (*codewords*). En los codificadores de bloque, la entrada de k bits se convierte en una palabra de n bits. Por tanto, se tienen $(n - k)$ bits de redundancia. Para su implementación, se utiliza la *Communications Toolbox*.

Para mejorar todavía más la resistencia frente a los errores producidos por el medio se utiliza el entrelazado que consiste en redistribuir de los bits para evitar tener ráfagas enteras con errores.

4.4.1 BCH

Los códigos $BCH(r, m)$ son códigos cíclicos de gran importancia y utilidad en el mundo de las telecomunicaciones por tener buena ganancia de código y porque se pueden utilizar con altas velocidades. Tienen los siguientes parámetros:

- Longitud de bloque: $n = 2^m - 1$ para $m \geq 3$
- Longitud de la entrada: $k \geq 2^m - mr - 1$
- Distancia del código: $d = 2r + 1$
- Máximo número de errores que puede corregir $t < \frac{2^m - 1}{2}$.

4.4.2 Reed–Solomon

Los códigos $RS[n, k, n - k + 1]$ son códigos no binarios. Sus parámetros son:

- Distancia del código: $d = n - k + 1$
- Máximo número de errores que puede corregir $t < \frac{n - k}{2}$.

Como en el proyecto se trabaja únicamente con señales binarias y el codificador Reed-Solomon de la *Communications Toolbox* trabaja con números enteros, la implementación del codificador RS es la siguiente: se convierten los bits de entrada en enteros, se realiza la codificación y, finalmente, se convierten los números enteros de la salida en bits para el correcto funcionamiento del sistema.

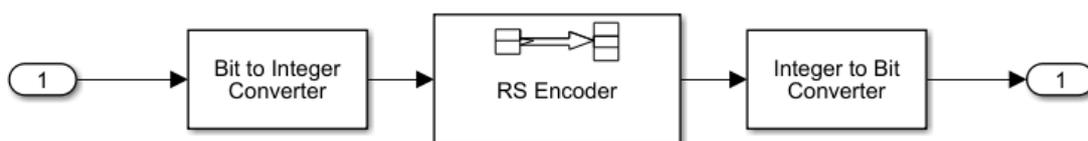


Figura 17. Implementación del codificador Reed-Solomon

4.5 Cabeceras

Una vez se genera la señal digital y ha sido codificada y entrelazada se le añade una cabecera para facilitar la sincronización a la hora de recibir la señal.

Al principio de cada trama se añade una cabecera que contiene un código de Barker. Los códigos de Barker son códigos bipolares con una autocorrelación muy alta y permiten sincronizar las tramas en el receptor. Existen pocos códigos de Barker binarios, se ha elegido el más largo de todos ellos para no estar tan limitados en relación señal a ruido.

Tabla 14. Códigos de Barker

Longitud del código	Secuencia Barker Bipolar
1	+1
2	+1 +1
3	+1 +1 -1
4	+1 +1 +1 -1
5	+1 +1 +1 -1 +1
7	+1 +1 +1 -1 -1 +1 -1
11	+1 +1 +1 -1 -1 -1 +1 -1 -1 +1 -1
13	+1 +1 +1 +1 +1 -1 -1 +1 +1 -1 +1 -1 +1

Se elige el código de longitud 13 y se convierte a unipolar obteniendo la secuencia [1 1 1 1 1 0 0 1 1 0 1 0 1]. Estos bits se repiten M veces según el número de bits por símbolo de la modulación. Al pasar por el modulador se convierten en 13 símbolos. El bloque que añade las cabeceras tiene la estructura de la Figura 18.

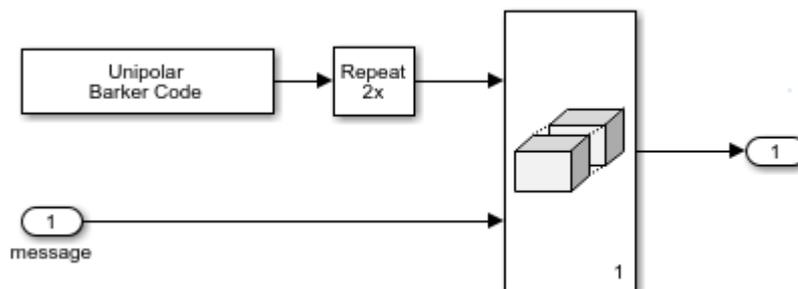


Figura 18. Estructura del bloque que añade las cabeceras

Por ejemplo, con modulación QPSK (2 bits por símbolo) la secuencia a la entrada del modulador es el código con cada bit repetido el número de bits por símbolo de la modulación, en este caso 2 veces: [1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 0 0 1 1 0 0 1 1], de esta manera, a la salida del modulador se tiene siempre una cabecera de longitud 13 símbolos. La estructura de la trama a la salida del modulador es la de la Figura 19.

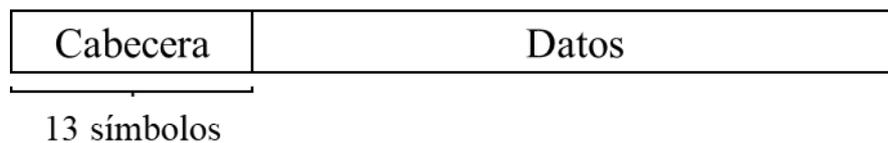


Figura 19. Estructura de las tramas a la salida del modulador.

4.6 Modulación

El bloque de modulación es el bloque más complejo del sistema. En este bloque se implementan las diferentes modulaciones, tanto analógicas como digitales, y también los distintos filtros. Como el SDR trabaja con un modulador IQ, se le deben entregar estas componentes. Para ello, se implementan los moduladores banda base de cada modulación los cuales se estudian en los siguientes apartados.

4.6.1 AM

La modulación AM modifica la amplitud de la señal portadora para transmitir la información. Para la modulación AM se ha obtenido a partir de la expresión paso banda (16), la expresión banda base (17). El parámetro principal de la modulación AM es el índice de modulación m que relaciona la amplitud del mensaje y la amplitud de la señal portadora. Para $m > 1$ se conoce como sobremodulación. Otro parámetro que puede modificarse es la fase φ_0 de la modulación. El desarrollo matemático es el siguiente:

Sea la expresión paso banda:

$$y(t) = A_C \cdot [1 + mx(t)] \cos(\omega_C t + \varphi_0) \quad (16)$$

La envolvente compleja banda base es la siguiente:

$$X_C = [1 + mx(t)]e^{j\varphi_0} \quad (17)$$

En la Figura 20 se observa la forma de onda un tono de 10 kHz modulado en AM con fase $\varphi_0 = \pi/3$ (para poder ver las componentes IQ de la modulación) junto con un modulador IQ con frecuencia central de 100 kHz. En la Figura 21 están las componentes IQ de la modulación AM.

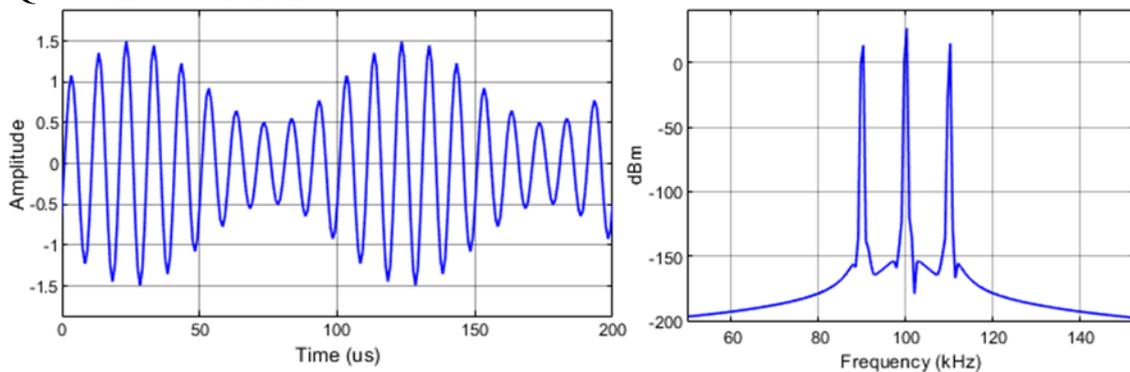


Figura 20. Forma de onda (izquierda) y espectro (derecha) de AM

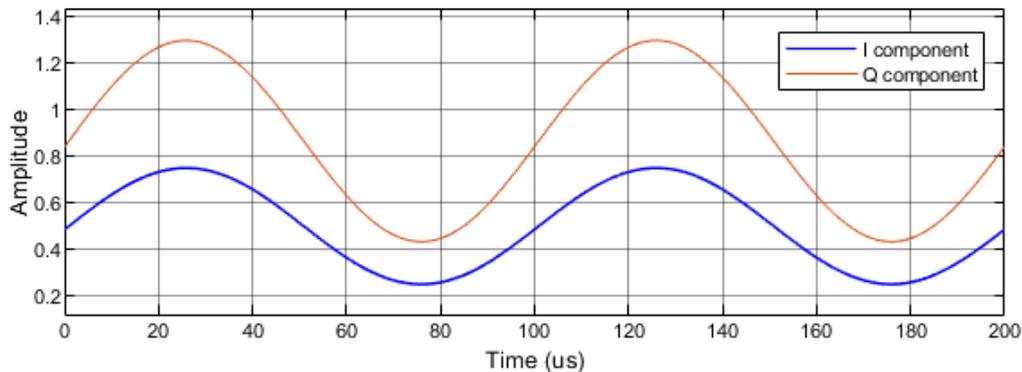


Figura 21. Componentes IQ para un tono modulado en AM

4.6.2 DSB

La modulación DSB o doble banda lateral es muy similar a la AM. La principal diferencia es que no tiene la portadora en el espectro de la señal, tan solo se queda con las bandas laterales, como podemos observar en la Figura 22. Al eliminar la portadora, se tiene una mejora de la eficiencia en potencia.

Debido a la ausencia de portadora, aparecen problemas en el receptor para sincronizar correctamente la frecuencia central de la señal al no tener una señal de referencia.

Al igual que en AM, para utilizar esta modulación con el SDR se necesitan las expresiones banda base. Partimos de la expresión general paso banda que es:

$$y(t) = A_C \cdot x(t) \cos(\omega_C t + \varphi_0) \quad (18)$$

De la cual podemos obtener fácilmente la expresión de la envolvente compleja en banda base que es:

$$X_C(t) = x(t)e^{j\varphi_0} \quad (19)$$

En la Figura 22 se observa la forma de onda un tono de 10 kHz modulado en DSB con fase $\varphi_0 = \pi/3$ (para poder ver las componentes IQ de la modulación) junto con un modulador IQ con frecuencia central de 100 kHz. En la Figura 23 están las componentes IQ la modulación DSB.

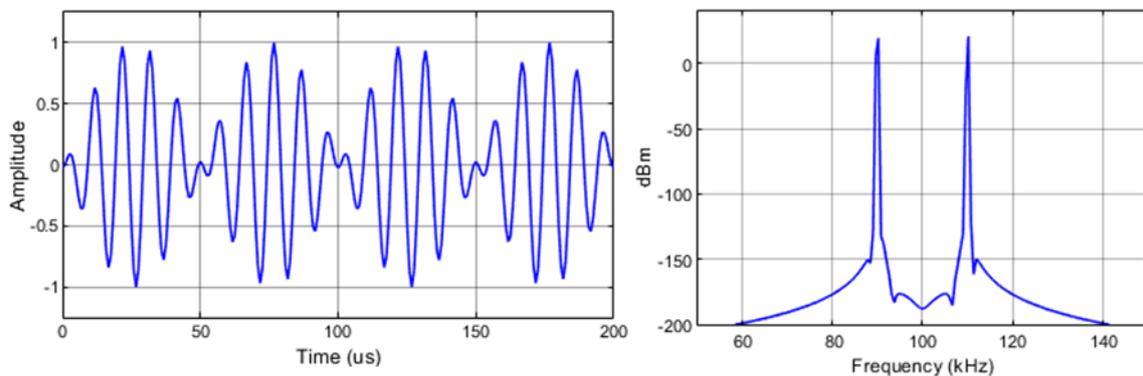


Figura 22. Forma de onda (izquierda) y espectro (derecha) de DSB

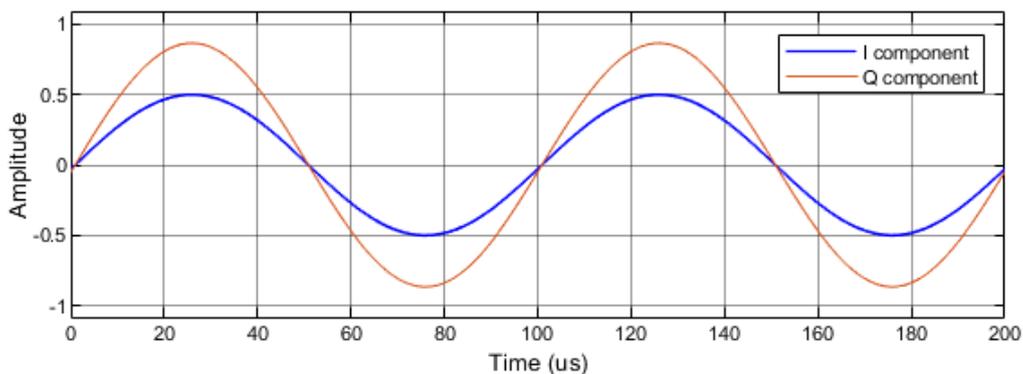


Figura 23. Componentes IQ de un tono modulado en DSB

4.6.3 SSB

La modulación SSB o banda lateral única consiste en transmitir solo una de las bandas laterales de la DSB. La banda lateral superior se llama *Upper Side Band* (USB) y la inferior *Lower Side Band* (LSB). Se consigue una eficiencia teórica del 100% porque toda la potencia se utiliza para transmitir la información sin repetición de banda. Esta ventaja trae consigo un problema que se manifiesta a la hora de sincronizar la señal en recepción.

Generar esta modulación de forma práctica no es del todo sencillo y hay varios métodos diferentes, todos ellos orientados a la generación de la señal paso banda y no banda base. A continuación, se mencionan estos métodos.

En primer lugar, está el método de filtrado paso banda, que consiste en generar una AM y filtrar la banda lateral no deseada y la portadora. De esta manera se reduce el ancho de banda y toda la potencia del amplificador va destinada a la información enviada.

En segundo lugar, está el modulador de Hartley que se basa en utilizar la fase para suprimir la banda no deseada. Se generan dos señales iguales en la que una de ellas está desfasada $\pi/2$ y desplazada en frecuencia de tal forma que, al sumarlas en contrafase, solo se elimina una de las bandas.

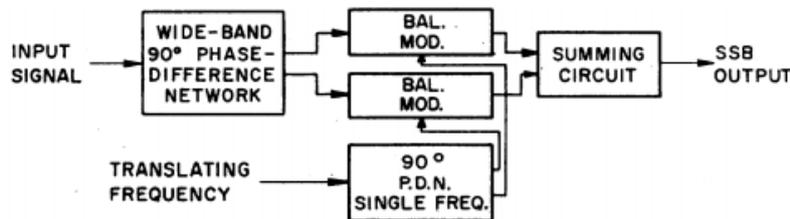


Figura 24. Modulador SSB de Hartley

En tercer y último lugar, se encuentra el modulador de Weaver [39] que solo utiliza filtros paso bajo y mezcladores en cuadratura. Primero se traslada la banda deseada a frecuencia cero utilizando un mezclador en cuadratura. Después se filtra con un filtro paso bajo, que elimina la banda no deseada. Por último, se devuelve la señal compleja al dominio real utilizando otro mezclador en cuadratura y la centra en la frecuencia deseada.

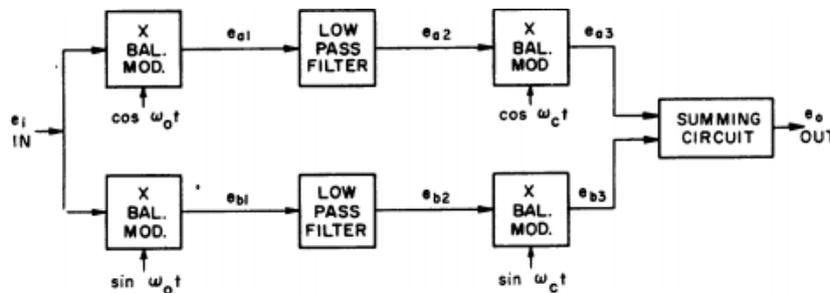


Figura 25. Modulador SSB de Weaver

La implementación en Simulink del modulador de Weaver no es sencilla. En este proyecto se ha utilizado el método que utiliza la transformada de Hilbert. La transformada de Hilbert es un procedimiento matemático con múltiples propiedades.

La propiedad que se aprovecha para los moduladores SSB es que aplicando la transformada de Hilbert a una señal, se obtiene una señal analítica.

Una señal analítica $y_{analitica}(t)$, (20) es una señal compleja creada tomando la propia señal $x(t)$ y sumando en cuadratura su transformada de Hilbert $\hat{x}(t)$.

$$y_{analitica}(t) = x(t) + j\hat{x}(t) \quad (20)$$

La expresión de la envolvente compleja banda base para la modulación SSB coincide con la señal analítica. Para elegir entre la banda superior y la inferior solo hay que cambiar el signo de la suma.

$$Y_{USB}(t) = A_C \cdot \frac{1}{2}(x(t) + j\hat{x}(t)) \quad (21)$$

$$Y_{LSB}(t) = A_C \cdot \frac{1}{2}(x(t) - j\hat{x}(t)) \quad (22)$$

Para implementar el modulador SSB se ha utilizado un filtro FIR que realiza la transformada de Hilbert. El orden del filtro es de 200. Se utiliza un orden alto para conseguir mejor atenuación en la banda no utilizada. Como el filtro es FIR, es necesario añadir un retardo a la otra rama para que ambas estén sincronizadas. Se recomienda utilizar una frecuencia de muestreo lo más baja posible para mejorar la atenuación de la banda eliminada.

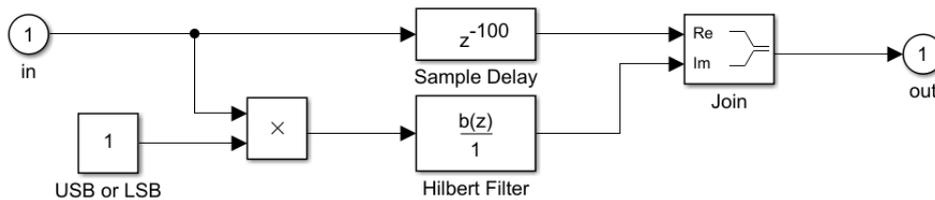


Figura 26. Modulador SSB aplicando filtro de Hilbert

En la Figura 27 tenemos el mismo ejemplo que en AM y DSB (tono 10 kHz con un modulador IQ con frecuencia central de 100 kHz). En este caso se utiliza SSB-USB.

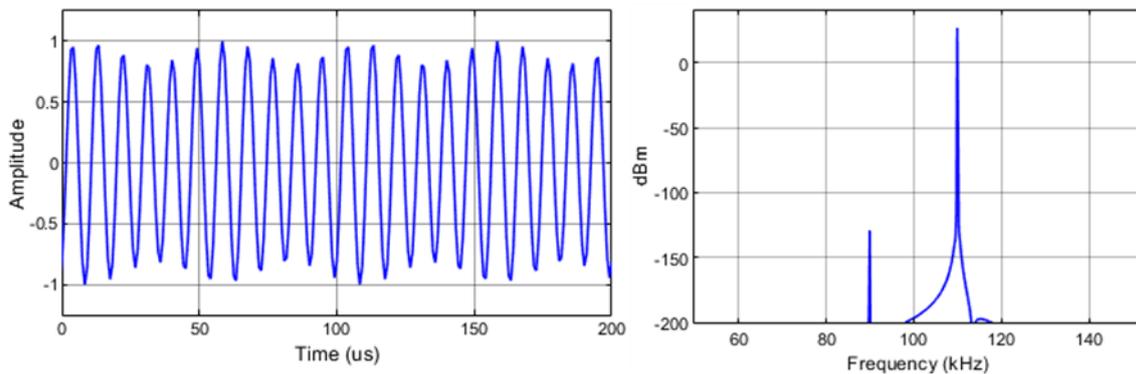


Figura 27. Forma de onda (izquierda) y espectro (derecha) de SSB

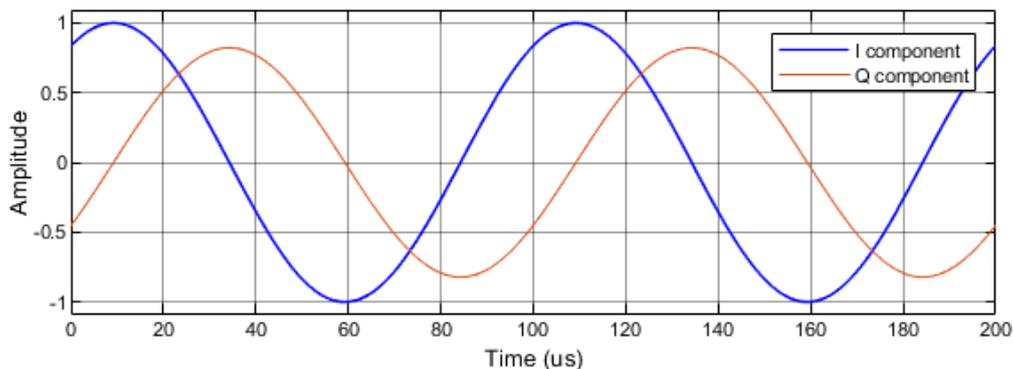


Figura 28. Componentes IQ de un tono modulado en SSB

4.6.4 FM

La modulación FM modifica la frecuencia instantánea de la portadora para enviar información. En este caso se utiliza el modulador implementado por la *Communications Toolbox*. La expresión paso banda de la modulación FM es (23), donde Δf es la desviación de frecuencia en Hz.

$$y(t) = A_c \cdot \cos\left(\omega_c t + 2\pi\Delta f \int_0^t x(\tau) d\tau\right) \quad (23)$$

Realizando el desarrollo de la ecuación anterior se llega a la expresión banda base:

$$y(t) = \frac{A}{2} e^{j\phi(t)} \text{ donde } \phi(t) = 2\pi\Delta f \int_0^t x(\tau) d\tau \quad (24)$$

El ancho de banda de la modulación FM es el ancho de banda de Carson B_C . Este ancho de banda se define como aquel que incluye aproximadamente el 98% de toda la potencia de una señal modulada en FM. Por ejemplo, modulando con un tono de frecuencia f_m , el ancho de banda de Carson resulta:

$$B_C = 2(\Delta f + f_m) \quad (25)$$

En la Figura 29 se observa la forma de onda y el espectro de un tono de 10 kHz modulado en FM con una desviación en frecuencia Δf de 40 kHz. En este caso, el modulador IQ tiene una frecuencia central de 300 kHz para poder observar correctamente el espectro. En la Figura 30 están las componentes IQ de la señal.

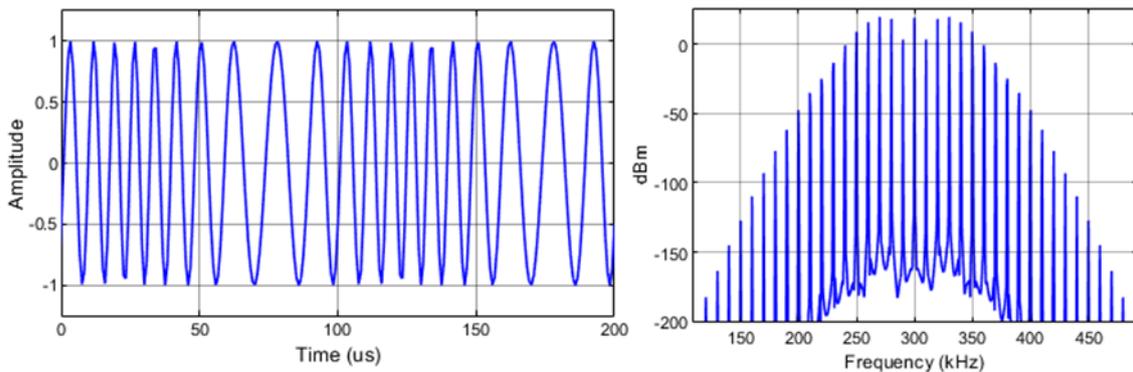


Figura 29. Forma de onda (derecha) y espectro (izquierda) de FM

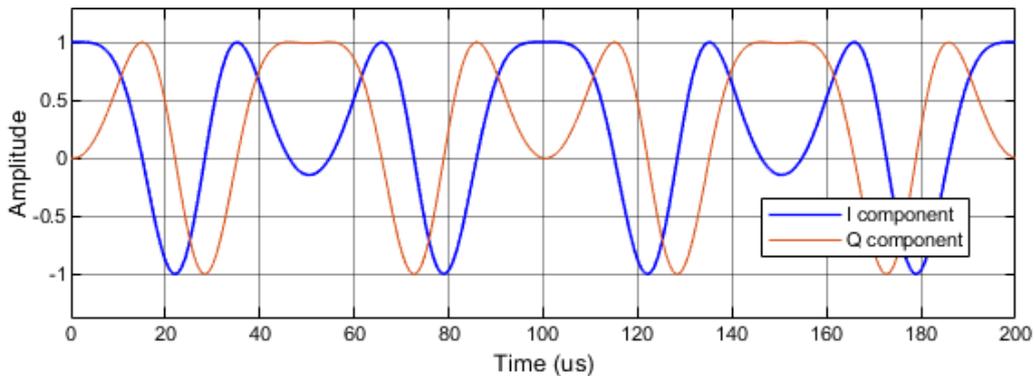


Figura 30. Componentes IQ de un tono modulado en FM

4.6.5 ASK

La modulación por desplazamiento de amplitud realiza variaciones en la amplitud de la portadora para transmitir información. La amplitud máxima es 1 y el número de símbolos diferentes es M . Las amplitudes A_i para cada símbolo se obtienen como:

$$A_i = \frac{2^{i-(M-1)}}{M-1}; \text{ donde } i = 0,1,2, \dots, M-1 \quad (26)$$

El diseño de este modulador se ha implementado de la siguiente manera: en primer lugar, se convierten los bits de entrada a números enteros i de longitud $\log_2 M$. Después se divide entre $M-1$ tal y como indica la expresión (26). Por último, se añade la fase, a elegir por el usuario.

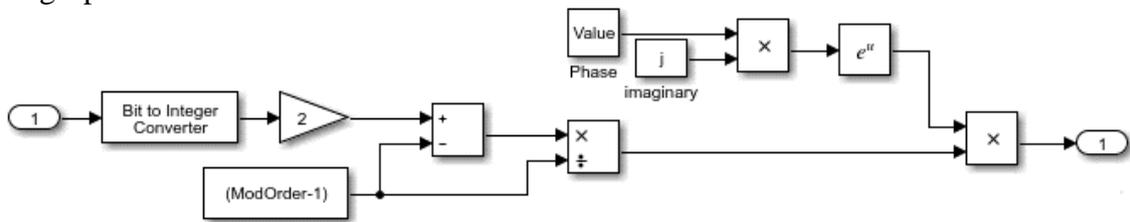


Figura 31. Esquema del modulador ASK

Las componentes IQ de esta modulación son las de la Figura 32. En este caso, la componente en cuadratura Q es cero porque la fase del modulador es cero. Su constelación se muestra en la Figura 33.

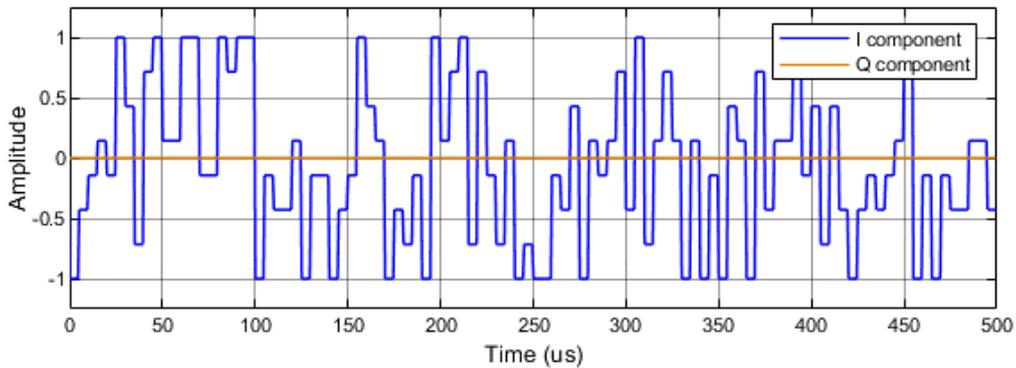


Figura 32. Componentes IQ de 8-ASK

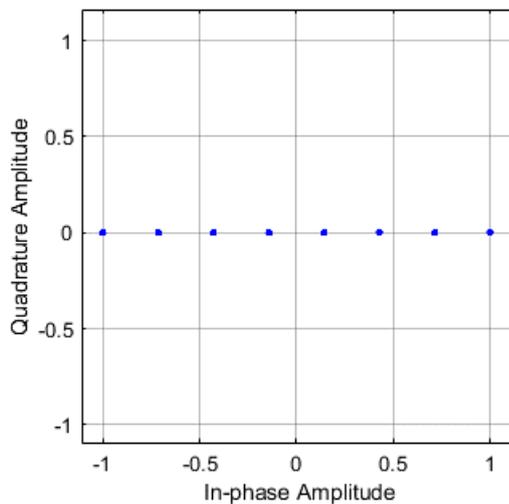


Figura 33. Constelación de 8-ASK

En la Figura 34 se muestra el diagrama de ojo utilizando filtrado de tipo coseno alzado con factor de *roll-off* $\alpha = 0.7$.

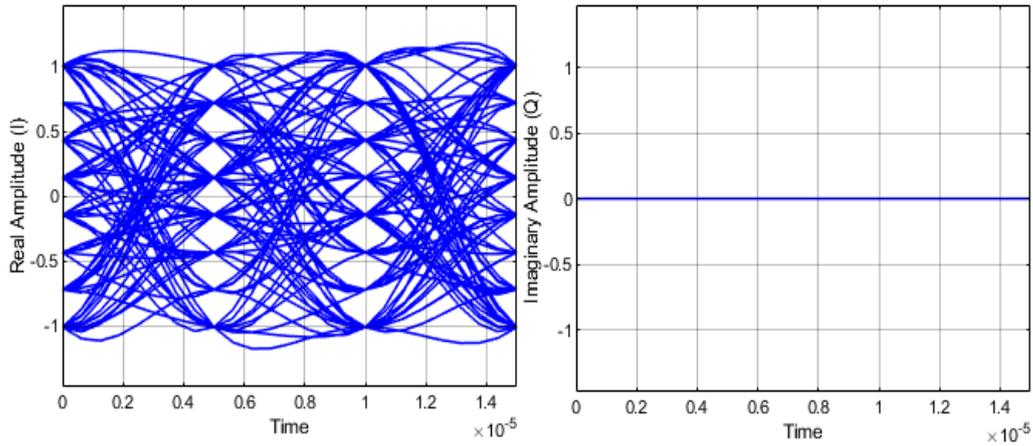


Figura 34. Diagrama de ojo de 8-ASK con $\alpha = 0.7$

En la Figura 35 se muestra, en negro, la forma de onda que tiene la modulación ASK una vez se modulan las componentes IQ (en azul la componente I, y en naranja la Q).

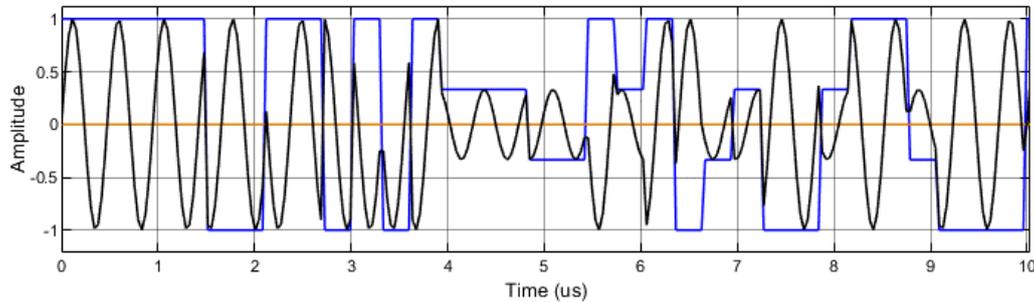


Figura 35. Forma de onda de ASK (en negro) a la salida del modulador IQ

Para el diseño del demodulador se utiliza el diagrama de bloques de la Figura 36. Primero se toma el valor absoluto de la señal de entrada para quedarse con el módulo y se multiplica por el signo de la parte real. Después se suma 1 y se multiplica por el factor $(M - 1)/2$, tal y como se indica en la ecuación de demodulación (27).

$$y = (x + 1) \cdot \frac{(M-1)}{2} \quad (27)$$

Se cuantifica en pasos de 1 y se aproxima al entero más cercano. Tras esto se recupera la secuencia recibida y se convierte a bits.

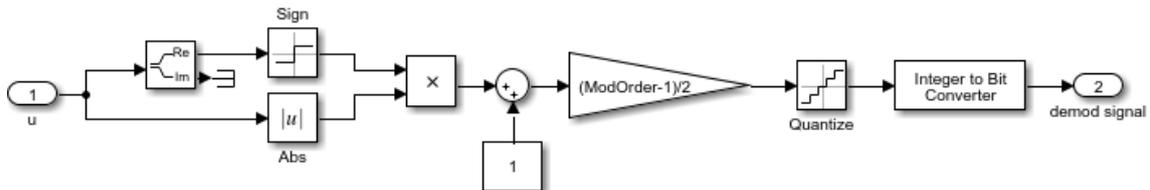


Figura 36. Demodulador ASK

4.6.6 PSK

La modulación por desplazamiento de fase realiza variaciones en la fase de la portadora para transmitir información. La amplitud máxima es 1 y el número de símbolos está determinado por el orden M de la modulación.

Para implementar esta modulación se ha utilizado el bloque “PSK Modulator Baseband” que incluye la *Communications Toolbox*. Este bloque entrega a la salida las componentes IQ siguiendo la ecuación (28).

$$s_n(t) = \exp\left(j\pi\left(\frac{2n+1}{M}\right)\right); n \in \{0,1, \dots, M-1\} \quad (28)$$

Los usos de la modulación PSK son variados: desde transmisores pasivos y de bajo coste, para identificación usando radiofrecuencia (RFID) como tarjetas de crédito o pasaportes biométricos, que utilizan BPSK, con $M = 2$, hasta comunicaciones de radiodifusión por satélite utilizando QPSK y 8PSK.

En las siguientes capturas se utiliza una modulación 8PSK con fase inicial de $\pi/8$. Además, se utiliza codificación de Gray.

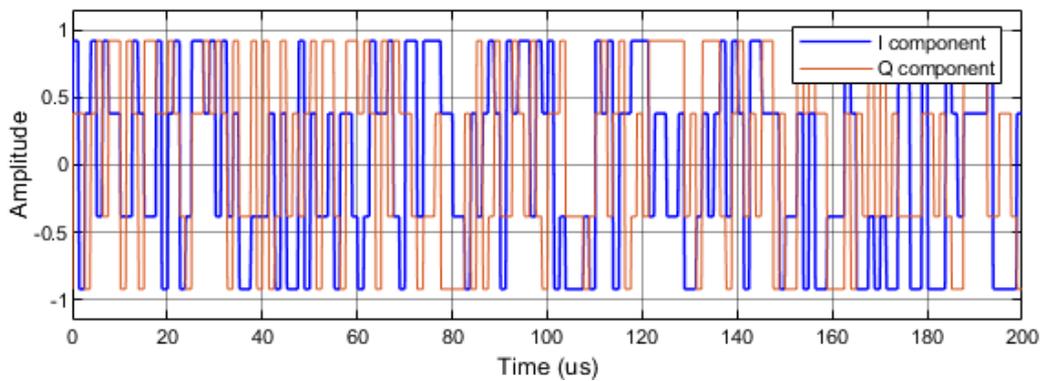


Figura 37. Componentes IQ de 8PSK

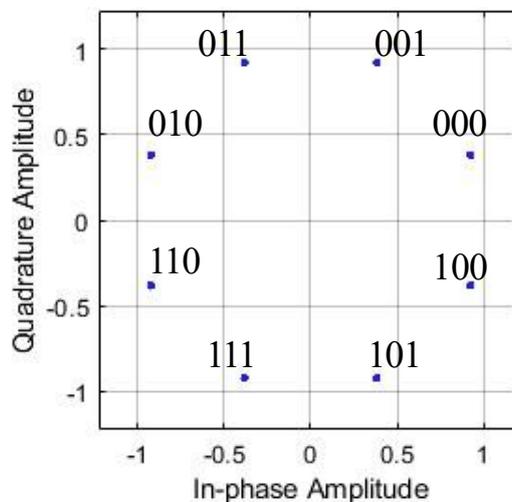


Figura 38. Constelación de 8PSK con fase $\pi/8$ y codificación de Gray

En la Figura 39 se muestra el diagrama de ojo utilizando filtrado de tipo coseno alzado con factor de *roll-off* $\alpha = 0.7$.

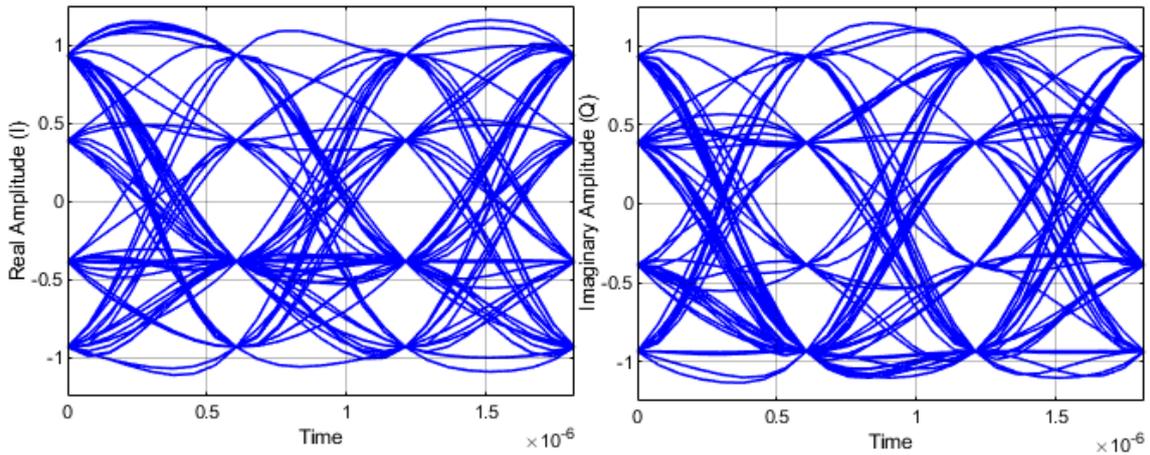


Figura 39. Diagrama de ojo de 8PSK con $\alpha = 0.7$

En la Figura 40 se muestra, en negro, la forma de onda que tiene la modulación 8PSK una vez se modulan las componentes IQ (en azul la componente I, y en naranja la Q). Pueden apreciarse los cambios de fase en la señal de salida del modulador IQ.

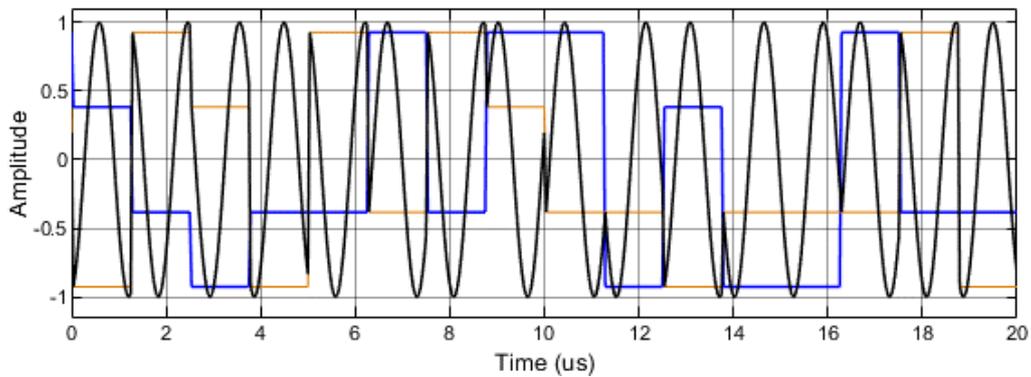


Figura 40. Forma de onda de PSK (en negro) a la salida del modulador IQ

4.6.7 FSK

La modulación por desplazamiento en frecuencia varía la frecuencia de la señal portadora según la información transmitida. Existen dos formas de generar señales FSK:

En primer lugar, tener M osciladores independientes de distinta frecuencia y que cada uno represente un símbolo. Esta forma provoca que la fase de la señal sea discontinua; a este tipo se le denomina FSK discontinuo. Esta forma no suele utilizarse porque se generan espurios en la señal de salida y ensanchamiento del espectro.

La segunda forma, es modular directamente la señal portadora usando la forma de onda de la información transmitida. Esta forma se conoce como modulación FSK de fase continua o CPFSK. Esta forma es la más utilizada puesto que la fase continua reduce la interferencia con los canales vecinos y reduce la potencia de los lóbulos laterales.

En el proyecto se implementa FSK de fase continua utilizando el bloque que introduce la *Communications Toolbox*. La ecuación que sigue esta modulación es

$$(29s(t) = A_c \cos\left(2\pi f_c t + D_f \int_{-\infty}^t x(\tau) d\tau\right) \quad (29).$$

$$s(t) = A_c \cos\left(2\pi f_c t + D_f \int_{-\infty}^t x(\tau) d\tau\right) \quad (29)$$

Donde D_f es el índice de modulación normalizado con $\frac{\pi}{R_S}$ que debe ser mayor de 0.5 y $x(\tau)$ es la señal moduladora normalizada con la información.

FSK se utiliza para aplicaciones como, por ejemplo: estaciones de telemetría, radiosondas meteorológicas y llaves de puertas de garaje.

En las siguientes figuras se utiliza una 4-FSK con $D_f = 0.7$. En la Figura 42 se observan las distintas frecuencias correspondientes a cada símbolo.

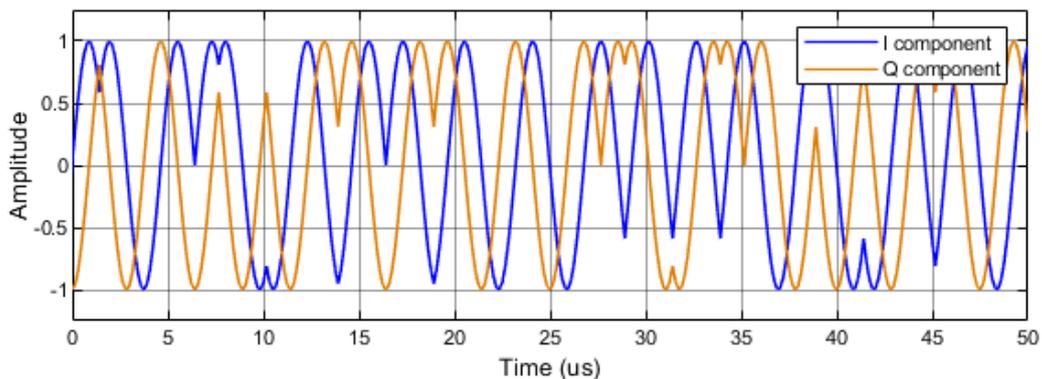


Figura 41. Componentes IQ de 4-FSK

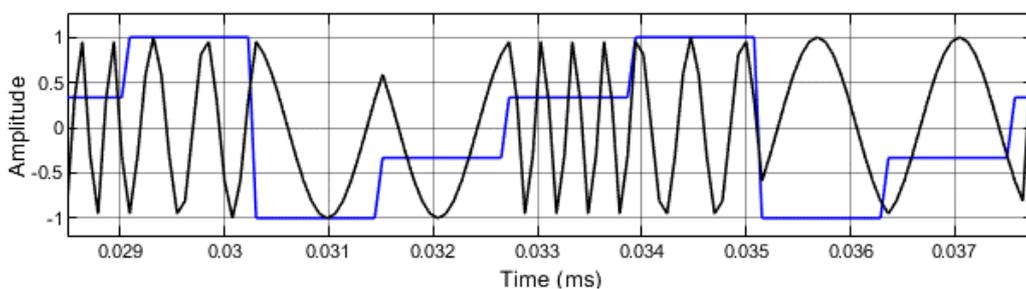


Figura 42. Forma de onda de 4-FSK (en negro) y señal enviada (en azul)

4.6.8 MSK

La modulación de desplazamiento mínimo es un tipo especial de modulación CPFSK en el que las dos frecuencias que se utilizan están desfasadas una cantidad igual a la mitad del tiempo de bit. Se puede interpretar como una modulación 2-FSK con índice de modulación $D_f = 0.5$. Se utiliza el bloque ya implementado por la *Communications Toolbox*. La ecuación a la salida del sistema es:

$$s(t) = \sqrt{\frac{2E_b}{T_b}} \cos \left[2\pi f_c t - m_I(t)m_Q(t) \frac{\pi t}{2T_b} + \phi_K \right] \quad (30)$$

Donde el producto m_I y m_Q puede ser -1 o 1, dependiendo de la entrada, y ϕ_K es la fase de la modulación. Para que los valores de entrada sean ± 1 , los bits de entrada se modifican de tal forma que si es un 0 pasa a ser un -1, y en el caso de 1 se queda en 1.

Las aplicaciones de esta modulación son variadas. La más popular y conocida es la variante GMSK que es gran interés en el mundo de la telefonía puesto que las redes 2G utilizan este tipo de modulación. Esta modulación puede verse también como una modulación de 4 fases o QPSK, la comparación entre ambas señales se realiza en el apartado 5.3 Práctica 5. Modulaciones digitales.

En la Figura 43 se muestran las componentes IQ de la modulación MSK y en la Figura 44 la constelación de la modulación.

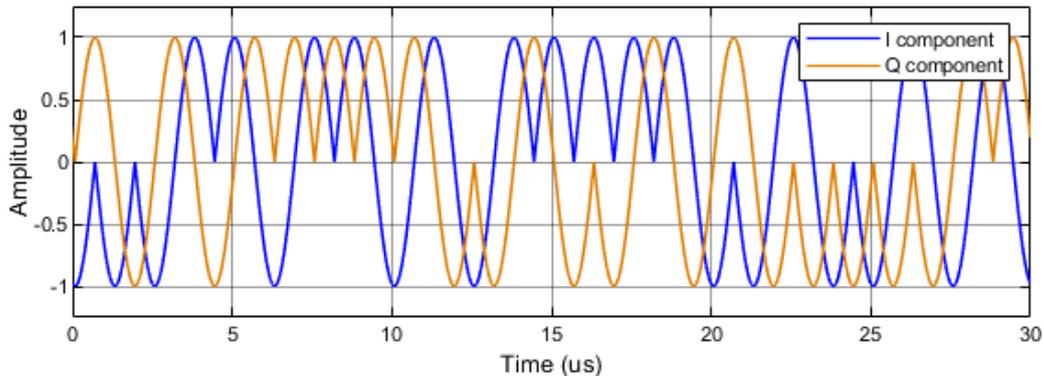


Figura 43. Componentes IQ de MSK

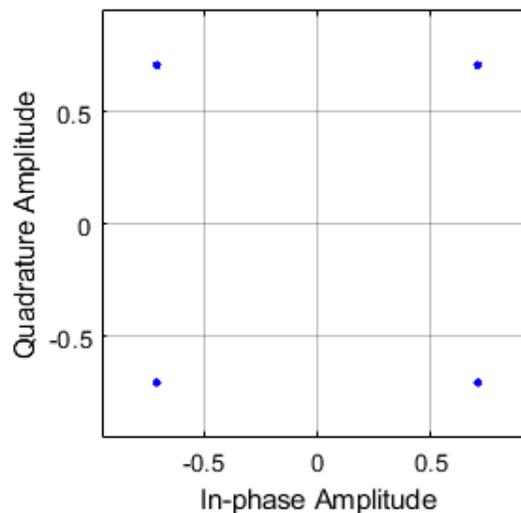


Figura 44. Constelación de MSK desfasada $\pi/4$

En el diagrama de ojo de la Figura 45 se puede comprobar que existe un desfase entre las componentes IQ.

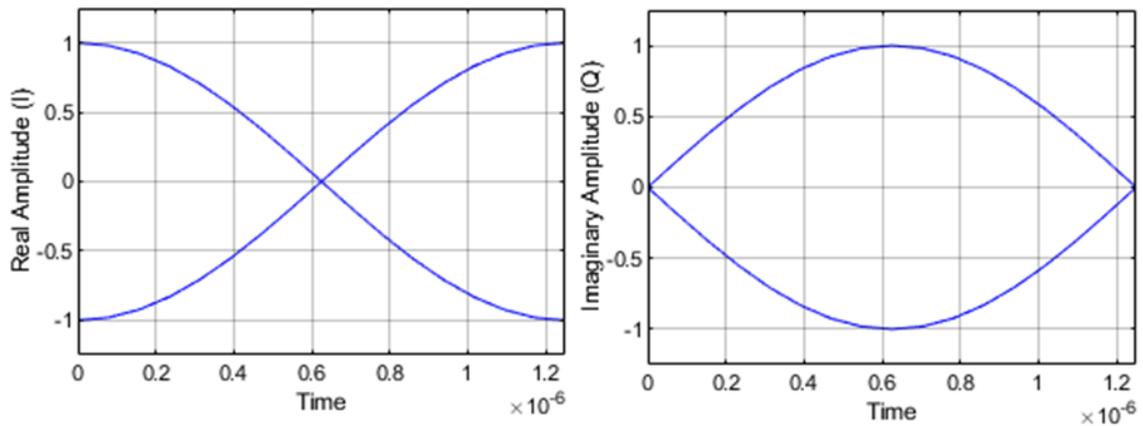


Figura 45. Diagrama de ojo de MSK

En la Figura 46 se puede observar el cambio de frecuencia en la señal de salida del modulador IQ, en negro, y la señal de información original, en azul, antes de ser modulada.

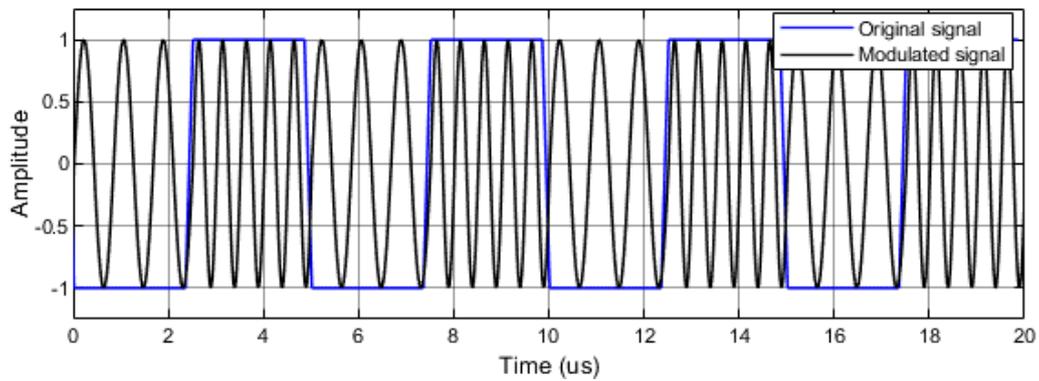


Figura 46. Señal original y señal modulada en MSK

4.6.9 QAM

La modulación de amplitud en cuadratura es un tipo de modulación en el que se envían dos señales independientes al modificar la fase y la amplitud de la portadora. La forma general de una M-QAM se define como (31), donde $i = 1, 2, \dots, M$ y $0 \leq t \leq T_s$.

$$s_i(t) = \sqrt{\frac{2E_{min}}{T_s}} a_i \cos(2\pi f_c t) + \sqrt{\frac{2E_{min}}{T_s}} b_i \sin(2\pi f_c t) \quad (31)$$

Se define E_{min} como la energía de la señal con menor amplitud y a_i, b_i como dos números enteros independientes que corresponden con la posición de un punto de la constelación. Se pueden escribir las componentes IQ como:

$$\phi_I(t) = \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t), 0 \leq t \leq T_s \quad (32)$$

$$\phi_Q(t) = \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t), 0 \leq t \leq T_s \quad (33)$$

La utilización de esta modulación se extiende por diferentes campos de las comunicaciones, por ejemplo: en España se utiliza 64-QAM para la televisión digital terrestre, también se utiliza en el estándar IEEE 802.11, que puede llegar hasta 256-QAM en Wi-Fi 5 (802.11ac) y hasta 1024-QAM en Wi-Fi 6 (802.11ax).

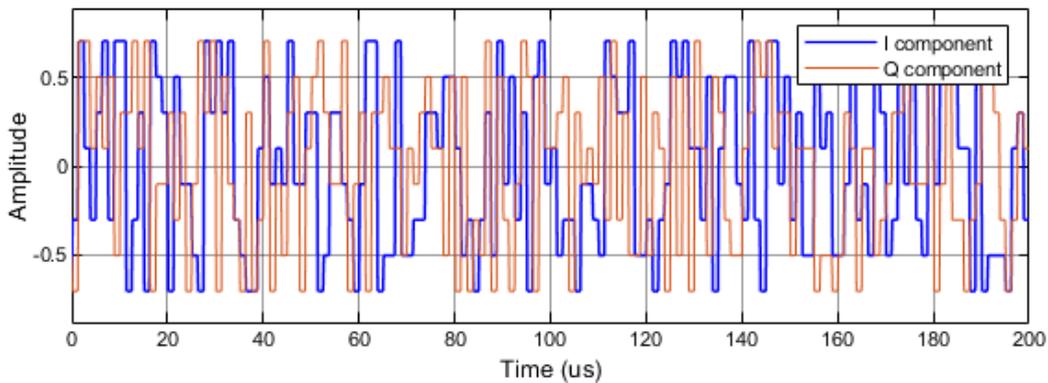


Figura 47. Componentes IQ de 64-QAM

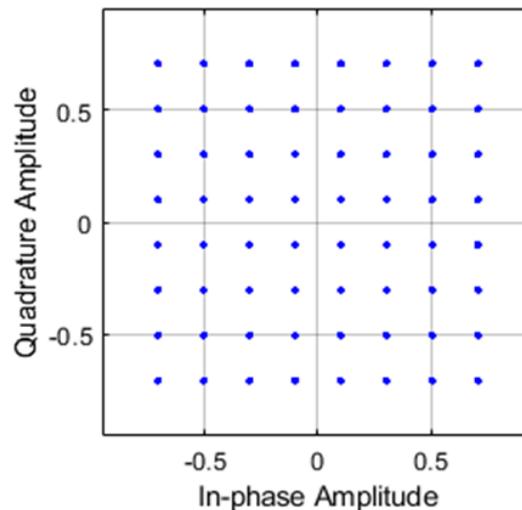


Figura 48. Constelación de 64-QAM

En la Figura 49 se muestra el diagrama de ojo utilizando filtrado de tipo coseno alzado con factor de *roll-off* $\alpha = 0.7$.

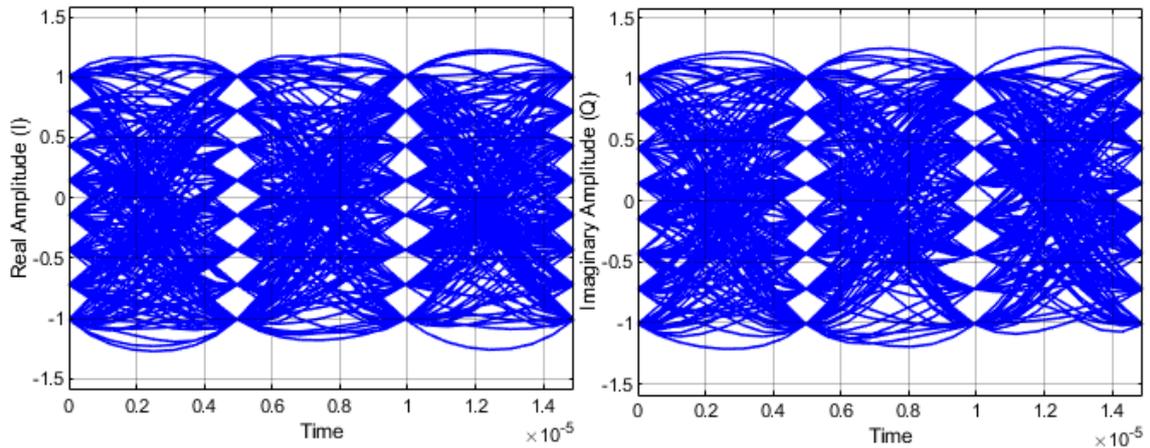


Figura 49. Diagrama de ojo de 64-QAM con $\alpha = 0.7$

En la Figura 50 se muestra, en negro, la forma de onda que tiene la modulación QAM una vez se modulan las componentes IQ (en azul la componente I, y en naranja la Q). Pueden apreciarse los cambios de fase y de amplitud en la señal de salida del modulador IQ.

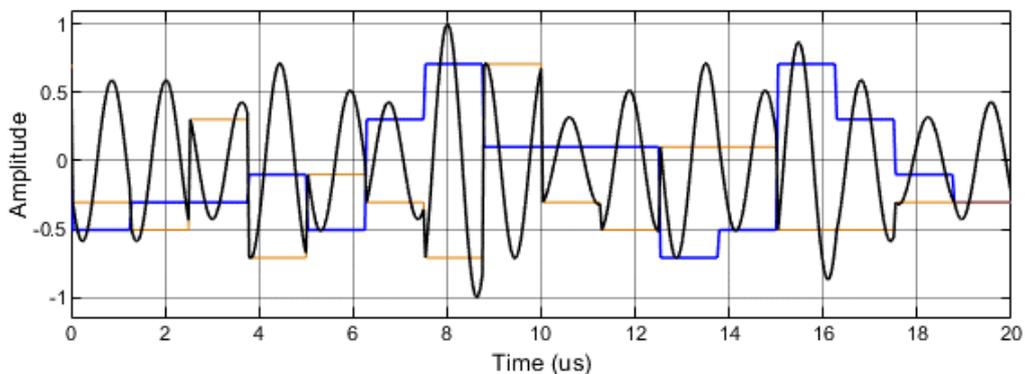


Figura 50. Forma de onda de 64-QAM (en negro) a la salida del modulador IQ

4.6.10 OFDM

La multiplexación por división de frecuencias ortogonales o, en inglés, *Orthogonal Frequency Division Multiplexing* (OFDM), es una técnica de transmisión multi portadora donde el flujo de información de alto ancho de banda se divide en varias subportadoras de banda estrecha. Estas subportadoras suelen llevar información modulada en QAM o PSK.

La división del flujo de datos en subportadoras hace que la modulación sea muy robusta a los desvanecimientos selectivos en frecuencia y simplifica la ecualización en recepción.

Para la generación de OFDM se sigue los pasos del siguiente esquema:

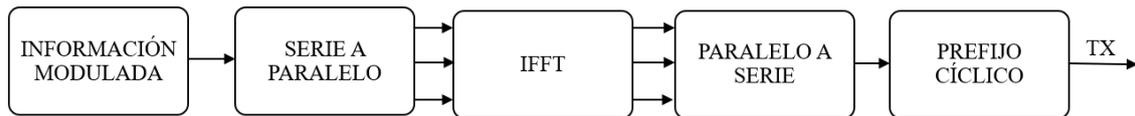


Figura 51. Esquema de generación de OFDM

La expresión matemática en banda base de la señal OFDM a la salida de la IFFT es (34) donde N es el número de subportadoras, i es el índice de la subportadora, X_i es la amplitud de la subportadora y $n = 0, 1, \dots, N - 1$

$$s_n(t) = \frac{1}{N} \sum_{i=0}^{N-1} X_i e^{j \frac{2\pi}{N} ni} \quad (34)$$

Para generar OFDM en el proyecto se ha utilizado el bloque de la *Communications Toolbox*. El bloque de OFDM realiza una modulación compleja en banda base. Los parámetros que pueden configurarse en este bloque son:

- Orden de la FFT (que es igual al número de subportadoras).
- Número de bandas de guarda a la derecha y a la izquierda del espectro.
- Longitud del prefijo cíclico.
- Número de símbolos OFDM.
- Posición de las señales piloto.
- Nulo en DC.

El prefijo cíclico [40] es una extensión del símbolo OFDM en el que se copian las últimas muestras del símbolo al principio de este, Figura 52, para evitar la interferencia intersimbólica. Las señales piloto se utilizan para mantener la sincronización, están moduladas en BPSK y no transmiten información; se puede modificar su ganancia y posición en el espectro.

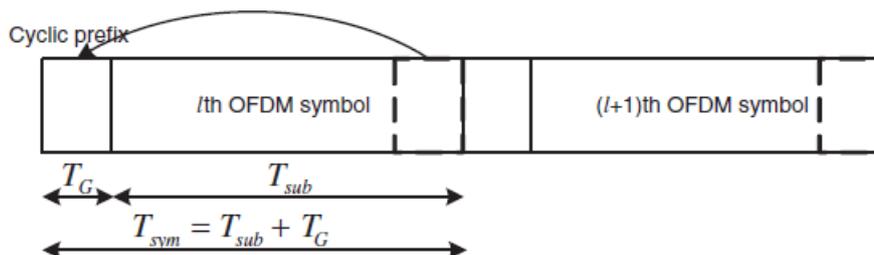


Figura 52. Utilización del prefijo cíclico en OFDM

Existen multitud de estándares que utilizan esta técnica de transmisión, por ejemplo: televisión digital terrestre DVB-T, radio digital DAB, redes WLAN o 802.11 (Wi-Fi), redes WiMAX, telefonía móvil 4G LTE y las nuevas redes 5G NR.

Para que OFDM funcione correctamente en el proyecto es necesario poner bien el número exacto de muestras a la entrada del modulador. Este número de muestras sigue la expresión (35):

$$N^{\circ}Simb_{OFDM} \cdot (LongFFT - N^{\circ}BandasGuarda - DC - N^{\circ}Pilotos) \quad (35)$$

Para facilitar el cálculo de muestras a la entrada del bloque OFDM, Figura 53, se ha incluido en la interfaz del bloque el número de muestras que deben entrar al bloque, *Input Length*, y, por tanto, el número de muestras que deben salir del bloque de modulación. Este valor obedece a (36) y debe ser el mismo que (34):

$$N^{\circ}Simb_{PSK/QAM} \cdot Sobremuestreo \quad (36)$$

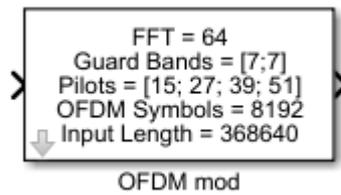


Figura 53. Bloque de OFDM

En el siguiente ejemplo, Figura 54, se utiliza un sistema OFDM que realiza una FFT de longitud 64 con 7 bandas de guarda a cada lado del espectro, nulo en DC y con pilotos en las posiciones 15, 27, 39 y 51, por tanto, 45 bandas (o subportadoras) llevan información. Se generan 8192 símbolos. Cada subportadora lleva datos modulados en 8PSK. Donde $N^{\circ}Simb_{PSK o QAM} = 46080$ y $Sobremuestreo = 8$.

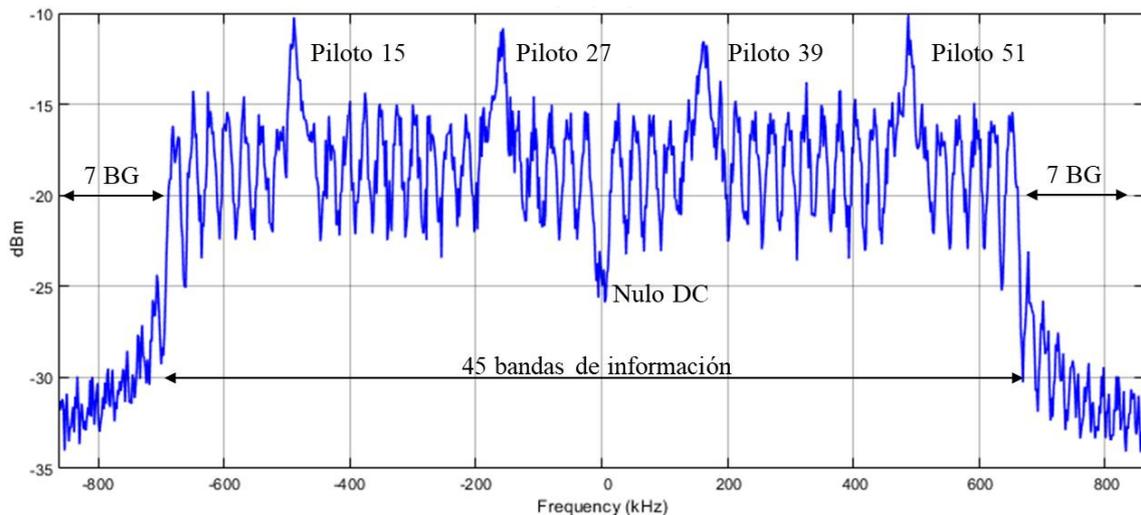


Figura 54. Espectro de OFDM

4.7 Filtros

En el bloque de modulación se han implementado cuatro filtros distintos que son los más habituales. Estos filtros están diseñados en banda base para filtrar la señal compleja en banda base y son:

4.7.1 Filtro Paso Bajo

Para la implementación de un filtro paso bajo se ha utilizado la *DSP System Toolbox*. Se utiliza un filtro FIR donde puede elegirse la frecuencia de corte y el rizado de la banda de paso.

4.7.2 Coseno alzado y raíz de coseno alzado

Los filtros en coseno alzado son filtros digitales que se utilizan para reducir la interferencia intersimbólica. Su parámetro principal es el factor de *roll-off* α . Para implementar estos dos filtros se ha utilizado el bloque de la *Communications Toolbox*.

La respuesta al impulso de este tipo de filtros es (37), donde T es el tiempo de símbolo.

$$h_{rc}(t) = \begin{cases} \frac{\pi}{4T} \operatorname{sinc}\left(\frac{1}{2\alpha}\right), & t = \pm \frac{T}{2\alpha} \\ \frac{1}{T} \operatorname{sinc}\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\pi\alpha t}{T}\right)}{1 - \left(\frac{2\alpha t}{T}\right)^2}, & \text{resto} \end{cases} \quad (37)$$

El factor de *roll-off* es una medida del exceso de ancho de banda del filtro respecto al ancho de banda de Nyquist. El ancho de banda del coseno alzado en banda base es (38):

$$BW = \frac{R_s}{2} (1 + \alpha) \text{ Hz} \quad (38)$$

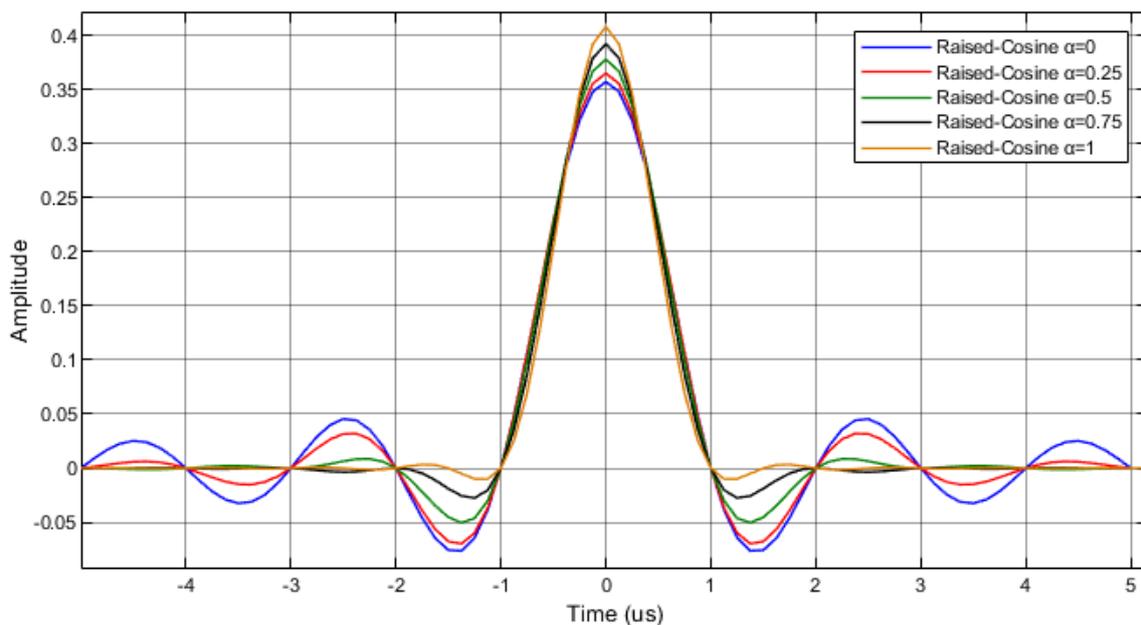


Figura 55. Respuesta al impulso del filtro coseno alzado

Los filtros raíz de coseno alzado, como su nombre indica, tienen la respuesta en frecuencia del filtro coseno alzado, pero realizando la raíz cuadrada:

$$|H_{rrc}(f)| = \sqrt{|H_{rc}(f)|} \quad (39)$$

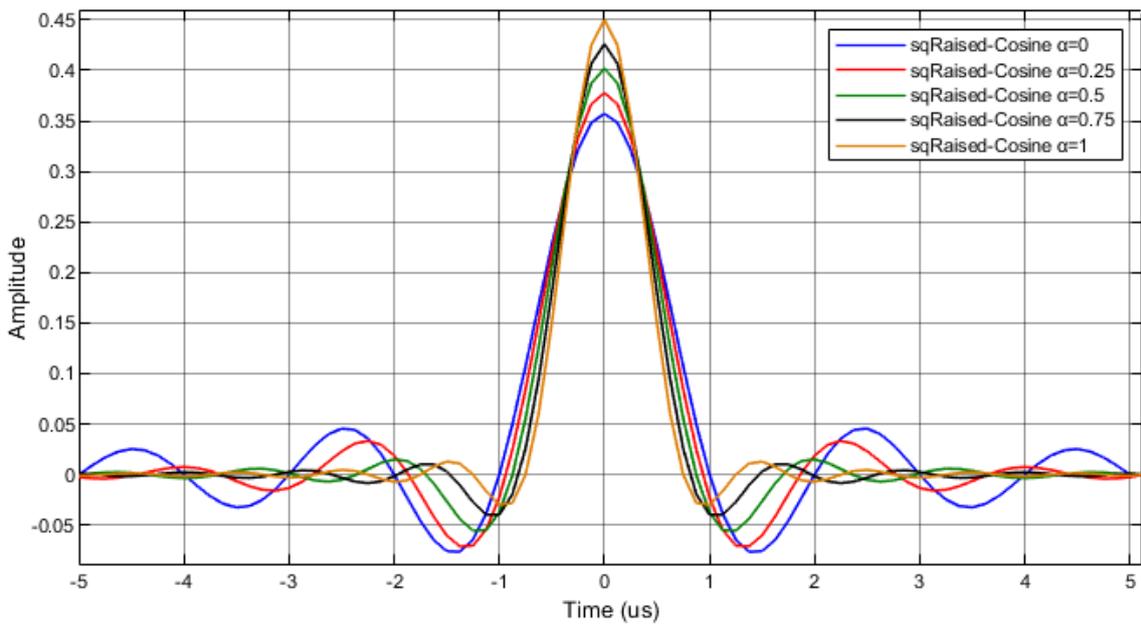


Figura 56. Respuesta al impulso del filtro raíz de coseno alzado

En la Figura 57 se muestra una comparativa entre los filtros coseno alzado y raíz de coseno alzado para valores de *roll-off* de 0.25 y 0.7. Se puede observar que el coseno alzado pasa por cero en los mismos puntos independientemente del valor de α , no ocurre así en el caso del raíz de coseno alzado.

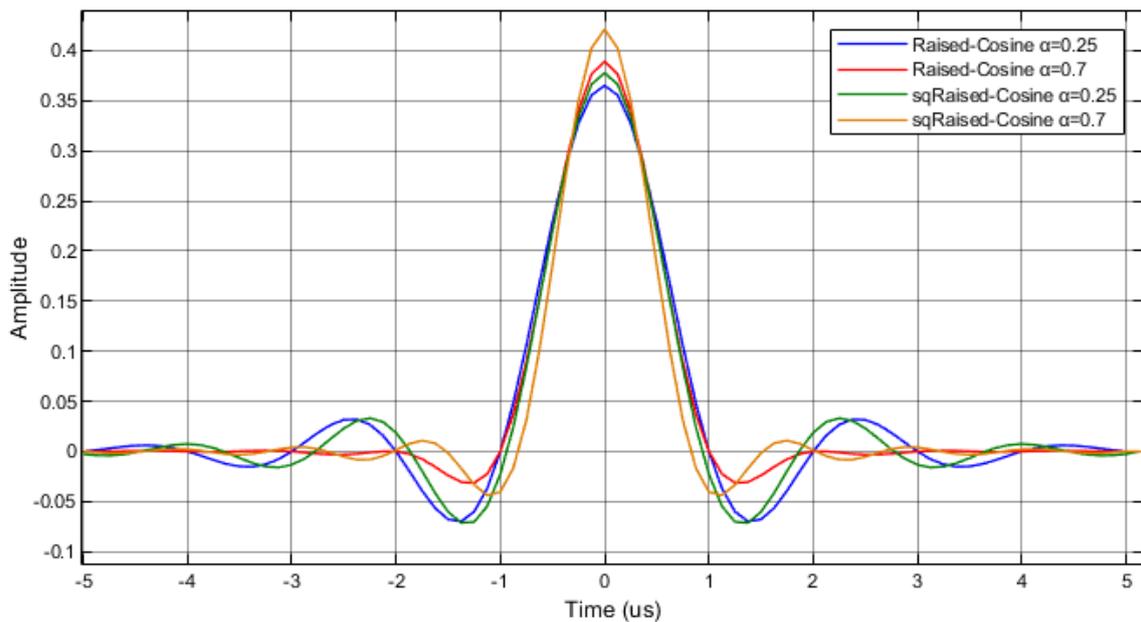


Figura 57. Respuesta al impulso de coseno alzado y raíz coseno alzado

4.7.3 Gaussiano

Se implementa un filtro Gaussiano con la *DSP System Toolbox*. Estos filtros tienen como respuesta al impulso la función de Gauss o una aproximación de ella. En este caso, la función “gaussdesing” implementa la siguiente respuesta al impulso:

$$h(t) = \frac{\exp\left(-\frac{t^2}{2\delta^2}\right)}{\sqrt{2\pi}\cdot\delta} \text{ donde } \delta = \frac{\sqrt{\log 2}}{2\pi BT} \quad (40)$$

Donde el parámetro BT es el producto entre el ancho de banda B a -3dB y el tiempo de símbolo T .

En la Figura 58 se muestra la respuesta al impulso del filtro implementado para distintos valores de BT .

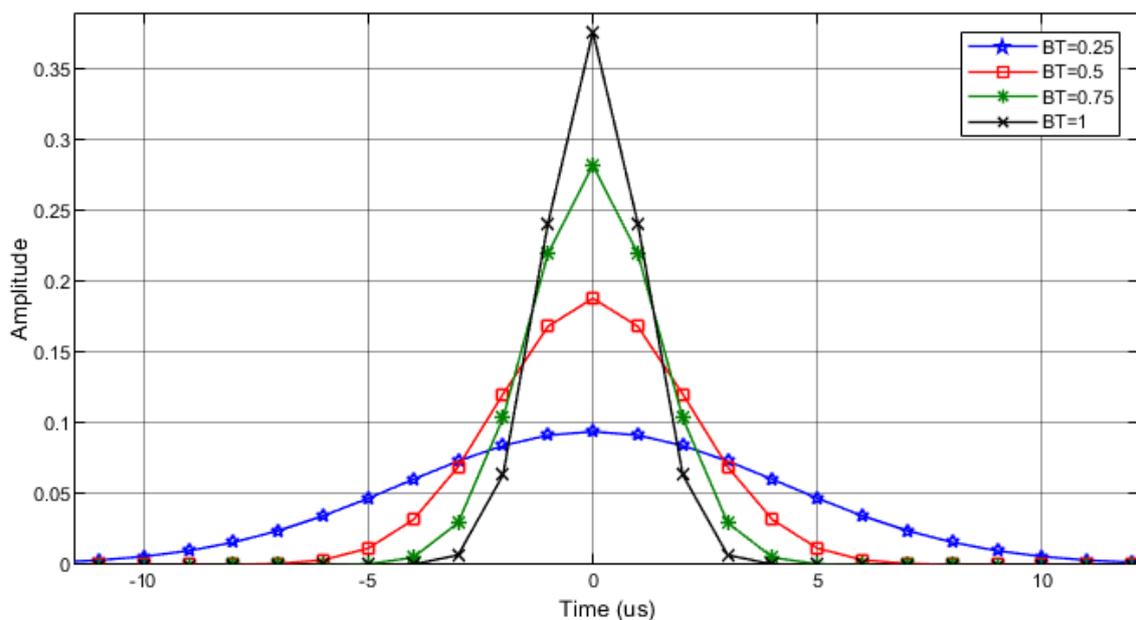


Figura 58. Respuesta al impulso del filtro gaussiano

4.7.4 Filtro de preénfasis

El filtro de preénfasis es un tipo de filtro que acentúa las frecuencias altas para mejorar la resistencia de la señal al ruido y a las interferencias producidas por el canal. Este filtro se usa antes de la modulación y solo con modulaciones analógicas. En recepción se realiza el proceso inverso, deénfasis, para recuperar la señal original. Su uso se da generalmente en aplicaciones de radio comercial como puede ser FM.

El filtro implementado en este proyecto es un filtro FIR que tiene la siguiente respuesta al impulso:

$$H(Z) = 1 - 0.95z^{-1} \quad (41)$$

En la Figura 59 se muestra la respuesta en frecuencia normalizada del filtro que se ha implementado en el proyecto. Para convertir la frecuencia normalizada a hercios, hay que multiplicar por la mitad de la frecuencia de muestreo.

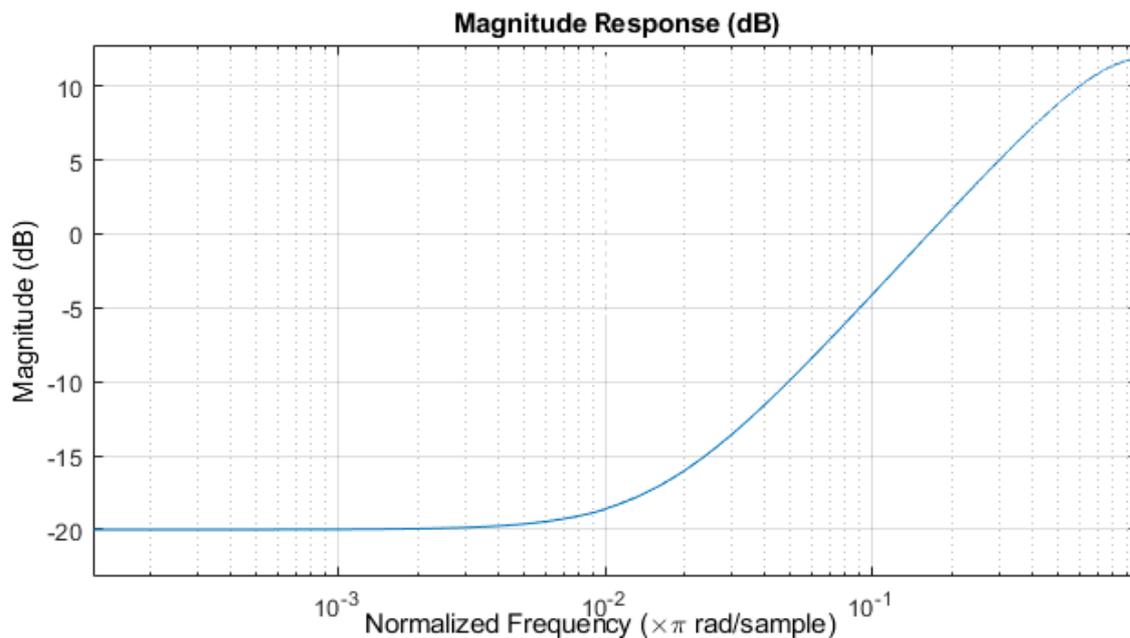


Figura 59. Respuesta en frecuencia del filtro de preénfasis

4.8 Efectos del canal

4.8.1 AWGN

Se implementa un canal que añade ruido blanco Gaussiano a la señal de entrada. Para ello se utiliza el bloque “AWGN Channel” de la *Communications Toolbox*. El parámetro de entrada es la relación entre la energía de bit y la densidad de potencia de ruido E_B/N_0 en dB. En la Figura 60 se muestra un ejemplo para 16-QAM con E_B/N_0 igual a 28 dB. En rojo se muestra la señal sin ruido, mientras que en azul se muestra la señal después de pasar por el bloque AWGN.

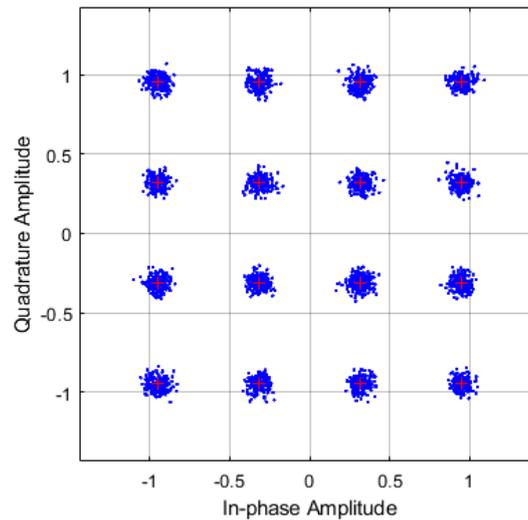


Figura 60. Efecto del AWGN sobre 16-QAM

4.8.2 Distorsión lineal

En este apartado se producen los efectos de atenuación y modificación de fase. La atenuación se mide en dB, pero para implementarlo se multiplica la señal de entrada por el valor de atenuación en veces. Para la fase se utiliza el bloque “Phase Offset” de la *Communications Toolbox*. Se introduce la fase en grados. En la Figura 61 se puede ver un ejemplo de cómo actúan estos efectos sobre una 16-QAM.

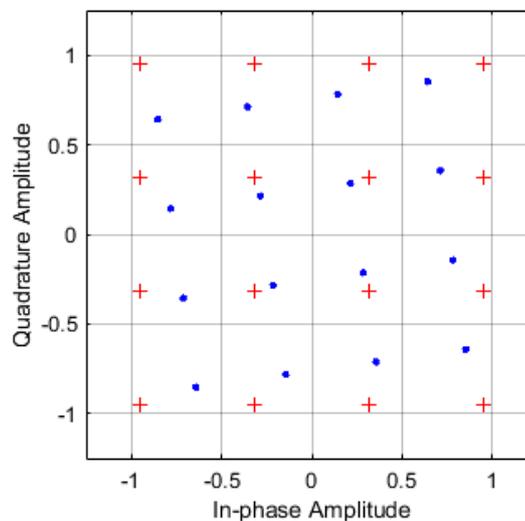


Figura 61. Efecto de 2 dB atenuación y de 8° desplazamiento fase para 16-QAM

4.8.3 Distorsión no lineal

La distorsión no lineal se produce en los sistemas en los que la señal de entrada no es proporcional a la de salida. Se generan armónicos y productos de intermodulación. En el proyecto se implementa siguiendo la ecuación (42). Los parámetros de entrada son los coeficientes k_2 y k_3 .

$$y(t) = x(t) + k_2 \cdot x^2(t) + k_3 \cdot x^3(t) \quad (42)$$

En la Figura 62 se muestra el efecto de la distorsión no lineal, en azul, sobre una señal 16-QAM. Los valores de los coeficientes son: $k_2 = 0.05$ y $k_3 = -0.05$

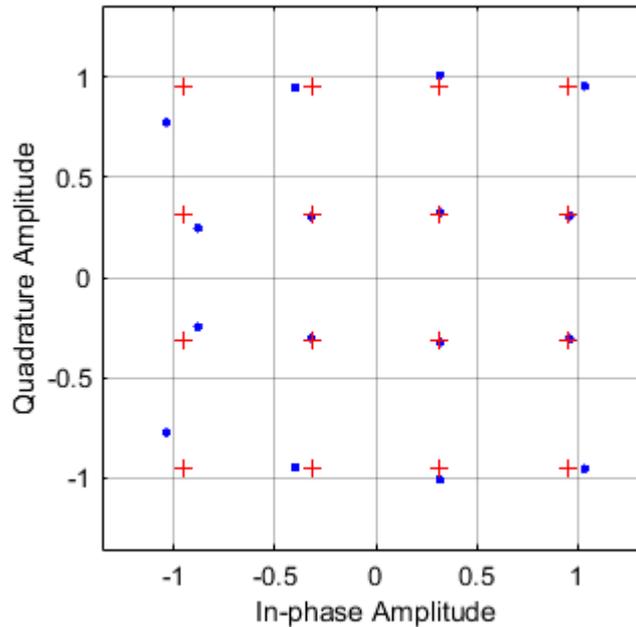


Figura 62. Efecto de la distorsión no lineal sobre 16-QAM

En la Figura 63 se muestra un tono de 20 kHz modulado en AM con los mismos valores de k_2 y k_3 . Para la traza en color negro se muestra la AM original sin distorsión. En azul, se pueden ver efectos de la distorsión no lineal creando armónicos a 40 y 60 kHz.

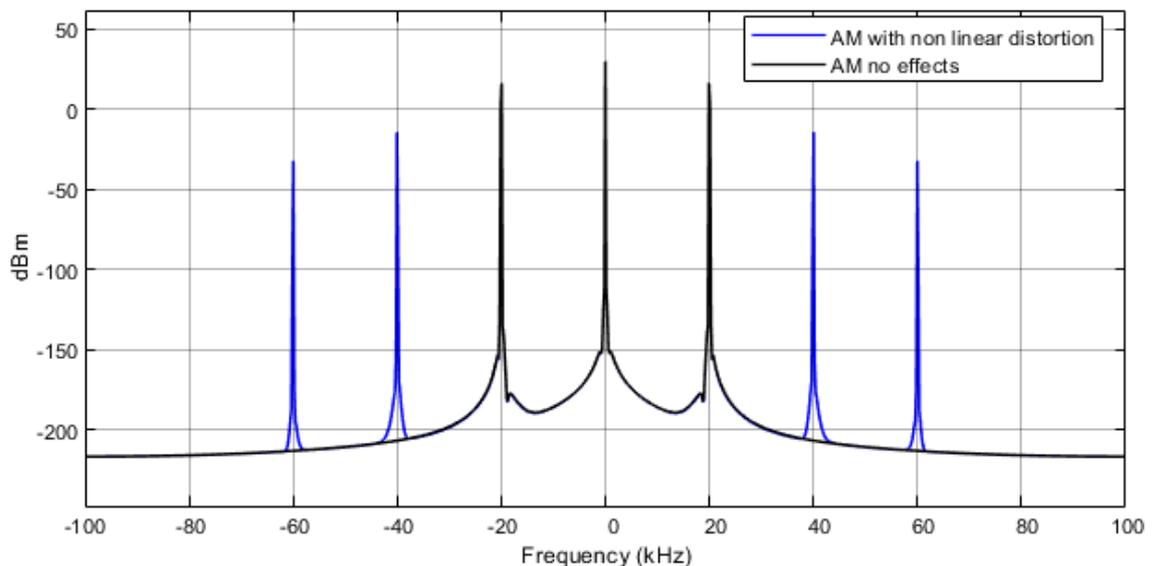


Figura 63. Efecto de la distorsión no lineal sobre tono en AM

4.8.4 Onda continua

Este efecto de canal añade una onda continua (CW) interferente de tipo sinusoidal. Se puede modificar la frecuencia f , el desfase ϕ_0 y la relación entre la señal portadora y la interferente C/I en dB. La ecuación de la onda interferente es (43).

$$y_{interferente} = 10^{\frac{C/I(dB)}{20}} \cdot \sin(2\pi f + \phi_0) \quad (43)$$

La señal (43) se suma directamente a la señal de entrada y produce los efectos que se observan en las siguientes figuras. Para la Figura 64 se ve el efecto de una CW de frecuencia 2.5 kHz y relación C/I de 20 dB sobre una modulación 8-PSK.

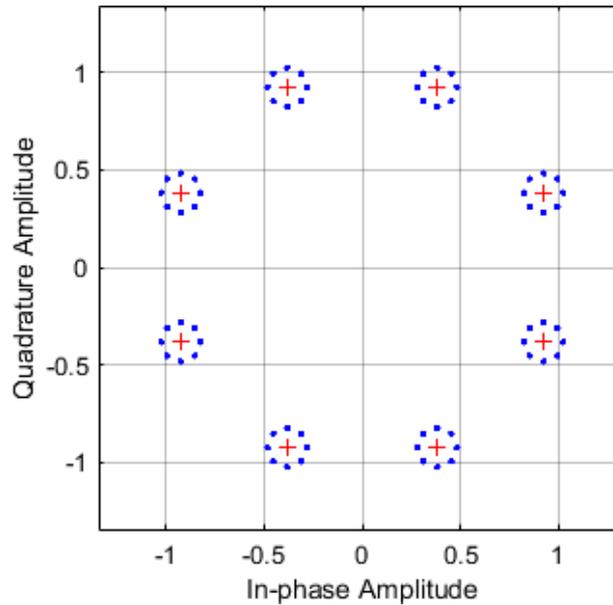


Figura 64. Efecto de la CW sobre una modulación 8-PSK.

En la Figura 65 se muestra un tono de 20 kHz modulado en AM junto con una señal interferente en 10 kHz y relación C/I de 30 dB como puede apreciarse en los marcadores.

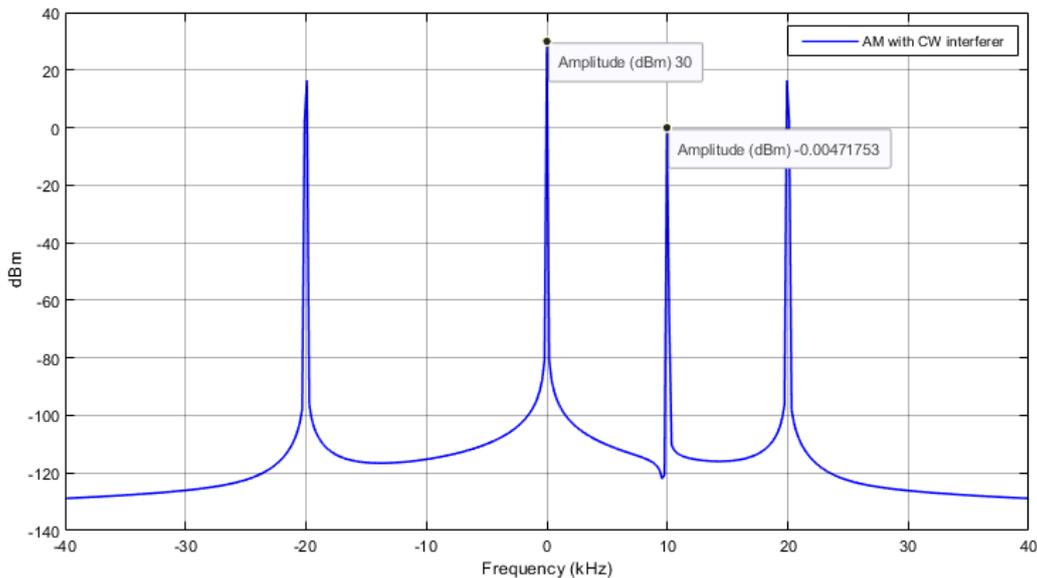


Figura 65. Efecto de la CW sobre un tono modulado en AM

4.8.5 Multitrayecto

El fenómeno de multitrayecto se produce cuando la señal de radio llega a la antena receptora por dos o más caminos y en instantes diferentes. La causa de la división en diferentes caminos viene dada por sucesos como son la reflexión en obstáculos y la acción de la atmósfera. Los efectos que provoca son un retardo en la señal, interferencia constructiva o destructiva y cambio de fase.

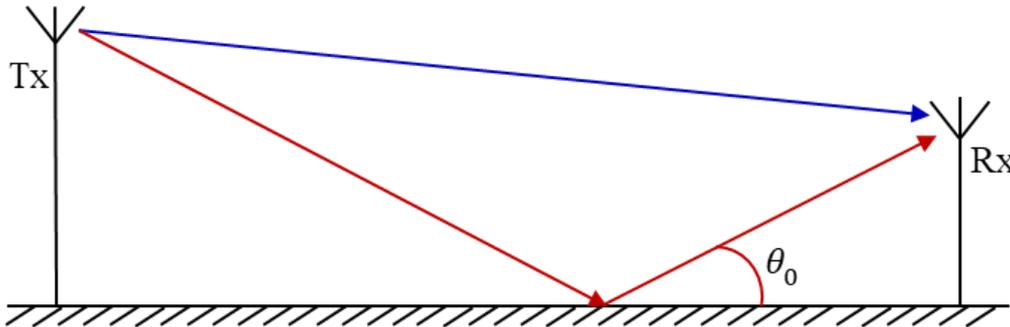


Figura 66. Modelo de multitrayecto para dos rayos.

En el proyecto se tienen hasta cuatro rayos diferentes en los que se puede modificar la relación ρ , en dB, respecto a la señal original, el retardo τ y la fase θ . Siendo $x(t)$ la señal original, la ecuación de salida del bloque es la ecuación (44).

$$y(t) = x(t) + \sum_{i=1}^4 [\rho_i \cdot x(t - \tau_i) \cdot e^{j\theta_i}] \quad (44)$$

La implementación del multitrayecto, Figura 67, se ha realizado de la siguiente manera: un primer bloque de atenuación de la señal (Att_i), tras esto se añade el retardo temporal (varDelay_i) Figura 67. Implementación del sistema de multitrayecto y la diferencia de fase (Phase_i) con el que llega la señal respecto al rayo directo. Por último, se suman todas los rayos reflejados junto al rayo directo. Los bloques iniciales (Mult_i) sirven para seleccionar el número de rayos, esto es transparente al usuario. No es necesario utilizar todos los rayos.

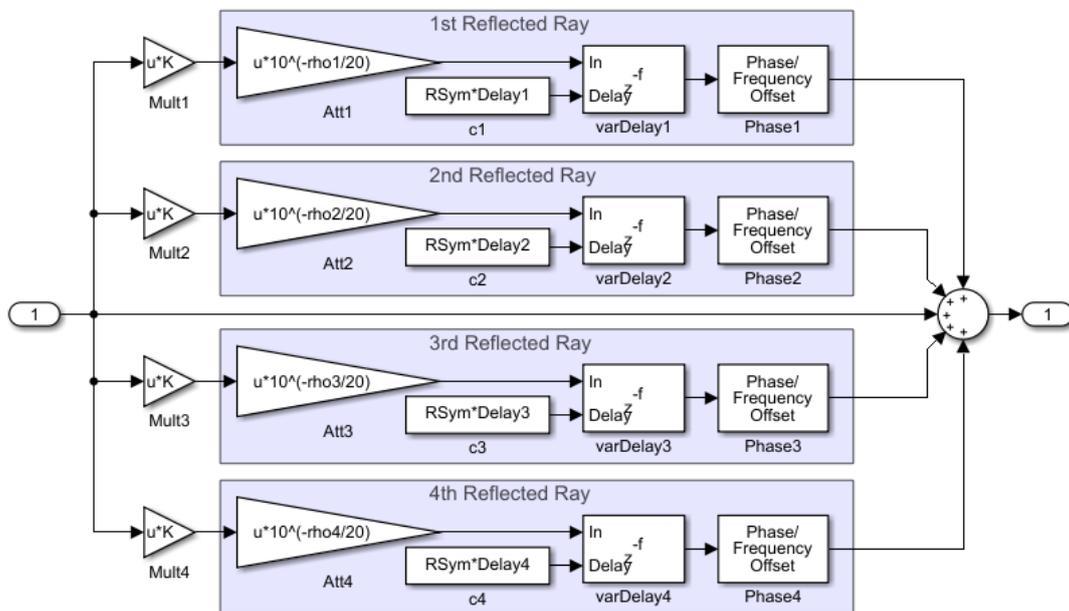


Figura 67. Implementación del sistema de multitrayecto con 4 rayos

En la Figura 68 se ve el efecto del multitrayecto para una modulación 16-QAM con dos rayos reflejados, los cuales tienen los parámetros de la Figura 69. Los efectos que se pueden observar es que se produce distorsión haciendo que aparezca en el lugar de cada símbolo otra 16-QAM que a su vez tiene otra 16-QAM.

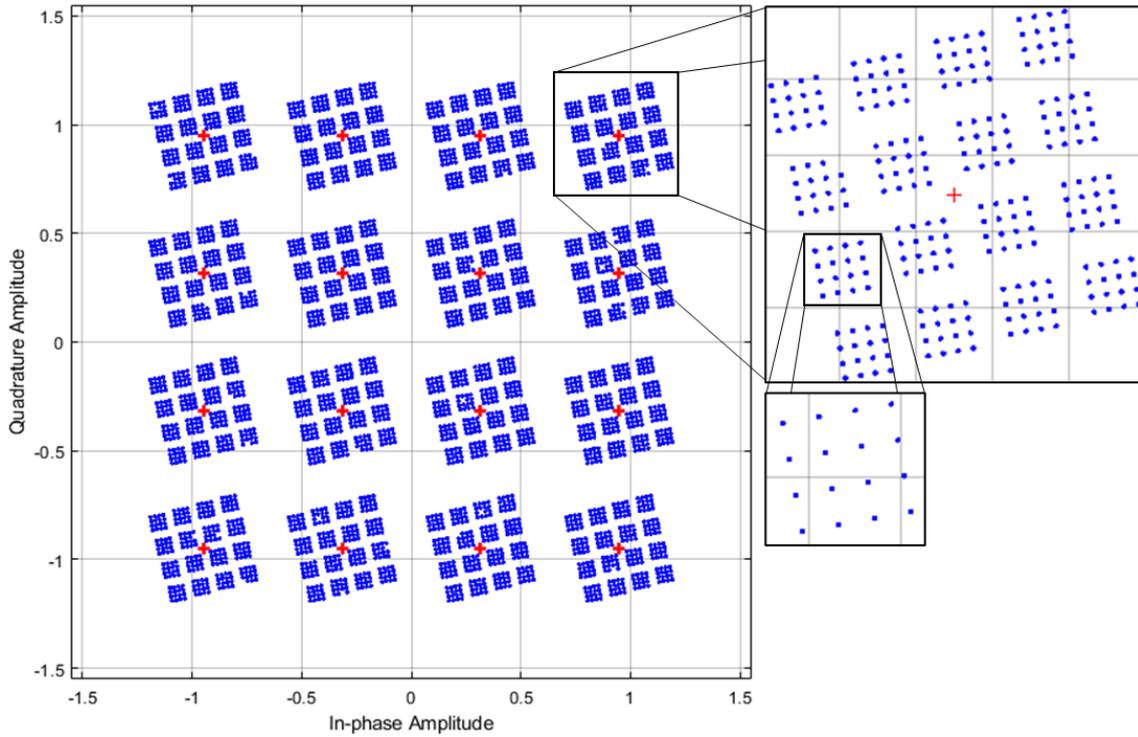


Figura 68. Efectos del multitrayecto sobre una modulación 16-QAM

Ray 1			
<input checked="" type="checkbox"/>	Level (dB)	30	:
	Delay (s)	0.5	:
	Phase (deg)	10	:
Ray 2			
<input checked="" type="checkbox"/>	Level (dB)	15	:
	Delay (s)	0.01	:
	Phase (deg)	15	:

Figura 69. Parámetros de los rayos utilizados en el ejemplo

4.9 Receptor

El receptor implementado sirve para calcular la tasa de error de bit (BER) que muestra a la salida del bloque. Este bloque debe estar desactivado cuando no se esté utilizando. El receptor se compone de varios bloques internos, Figura 70, que se explican en los siguientes párrafos.

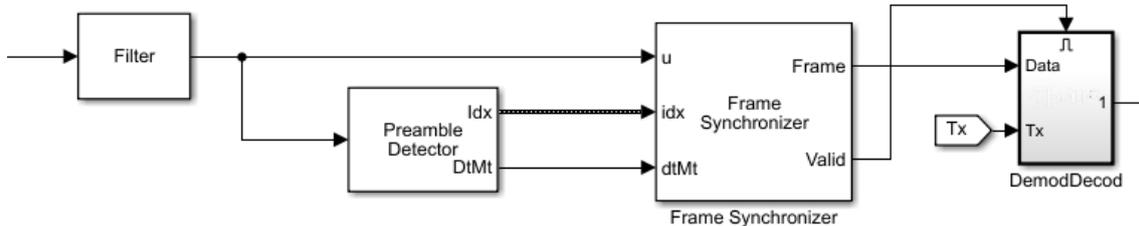


Figura 70. Bloques internos del receptor

En primer lugar, se realiza un filtrado de la señal recibida con los mismos filtros que en el apartado 4.7, configurables según las preferencias del usuario.

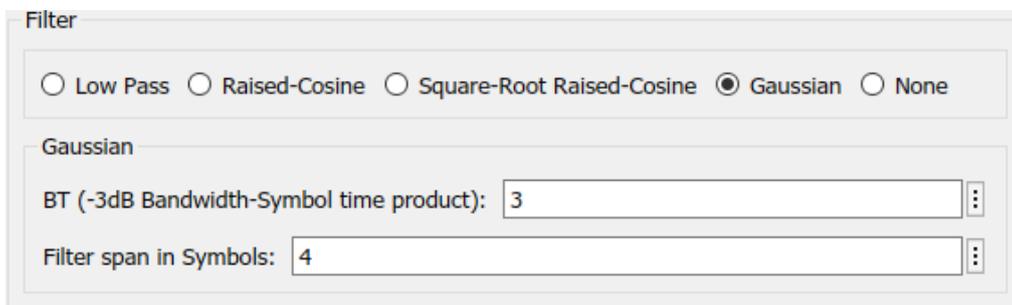


Figura 71. Captura del bloque de filtrado en el receptor

Después se realiza la sincronización de trama mediante un detector de la cabecera o de preámbulo en el que podemos configurar el umbral de detección. Este detector de preámbulo envía una señal al sincronizador de trama cuando encuentra un preámbulo válido.

El sincronizador de trama se encarga de quedarse solo con la parte correcta de la trama. De esta forma se consigue eliminar el retardo introducido por los filtros o los efectos del canal. Si no es capaz de sincronizar la trama no entrega señal de salida. Cuando la señal de salida es válida, se envía una señal de disparo o *trigger* que habilita el bloque demodulador-decodificador.

El bloque demodulador-decodificador se encarga de realizar los pasos del proceso de transmisión, pero a la inversa, estos son: demodular la señal recibida, quitar la cabecera, eliminar el entrelazado y decodificar la trama. Finalmente, se compara la señal recibida con la señal original, sin cabeceras para calcular la BER, que se muestra a la salida del bloque receptor, utilizando el bloque *Error Rate Calculation*.

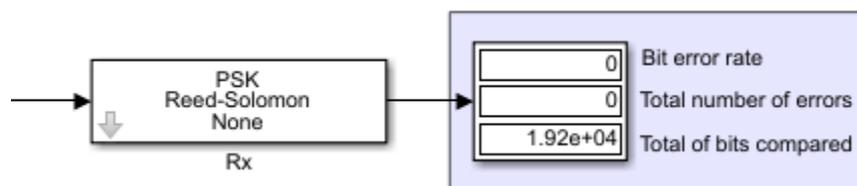


Figura 72. Bloque receptor y display con la BER

5. Resultados

Los resultados de este proyecto fin de grado son seis prácticas destinadas a laboratorios de sistemas de telecomunicación. Para su elaboración, se parte de las prácticas actuales que se hacen en el laboratorio de la asignatura de cuarto semestre Teoría de la Comunicación.

Se proponen dos prácticas para la parte de señal y ruido, dos prácticas para la parte de modulaciones analógicas y dos prácticas para la parte de modulaciones digitales.

5.1 Práctica 1. Visualización de señales en Simulink

Para la realización de esta práctica se ha simplificado el circuito utilizado con el objetivo de facilitar el aprendizaje a los alumnos. De esta manera, empiezan a conocer el funcionamiento de Simulink y de los distintos bloques de forma progresiva mejorando la curva de aprendizaje.

El objetivo de la práctica es enseñar a los alumnos cómo visualizar las señales en Simulink, realizar medidas de potencia y amplitud y practicar la conversión de unidades logarítmicas a unidades naturales y viceversa. Los parámetros que se pueden modificar son los mostrados en la Tabla 15.

Tabla 15. Parámetros de la práctica 1

Bloque	Parámetros
Sistema	Frecuencia de muestreo, f_s en Hz.
	Impedancia, Z en Ω
Tono	Frecuencia de la sinusoide, f_0 en Hz.
	Valor de pico de la sinusoide, A en V.
	Fase inicial de la sinusoide, θ_0 en rad.
Amplificador	Ganancia de potencia del amplificador G en dB.
	Figura de ruido del amplificador F_n en dB.

En la

Figura 73 se muestra la interfaz del sistema implementado en la que se pueden ver los bloques generador de señal, amplificador y visores (Scopes) sobre los cuales se realizarán las medidas.

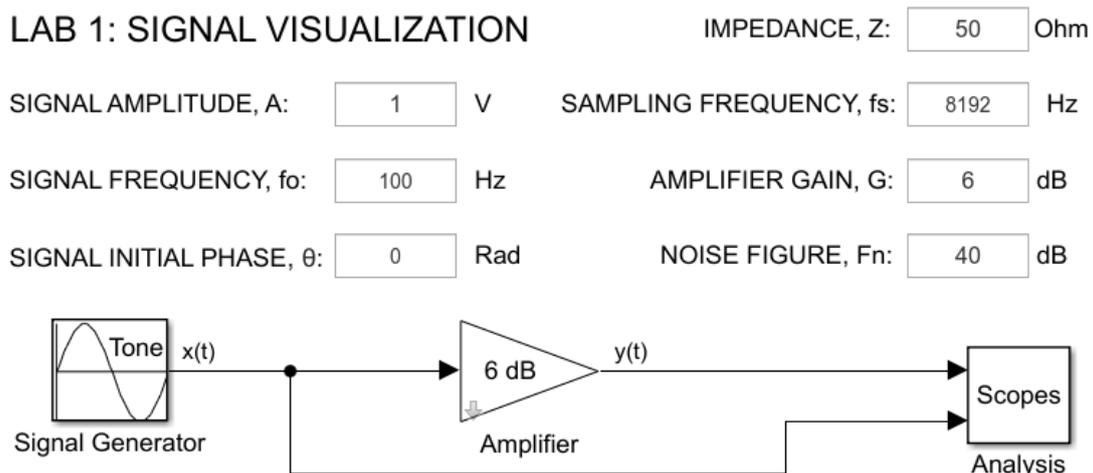


Figura 73. Interfaz del sistema de la práctica 1

Los bloques de medida son cuatro analizadores de espectro uno para potencia en W, otro para potencia en dBm, otro para densidad espectral de potencia y el último para calcular el valor cuadrático medio o RMS. Estos medidores son los bloques *Spectrum Analyzer* de la *DSP System Toolbox*. Se puede modificar el ancho de banda de resolución y el ancho de banda para medir potencia, entre otros parámetros. Para más información sobre su funcionamiento se recomienda acudir a la documentación oficial en la página de MathWorks [43].

Existe otro bloque de medida, en este caso, un osciloscopio que permite ver la señal en el dominio del tiempo para sacar información de amplitud, fase y frecuencia. Estas medidas se pueden realizar para la señal original $x(t)$ y para la señal amplificada $y(t)$ de forma simultánea. Este bloque pertenece a la *DSP System Toolbox* y se llama *Time Scope*, se puede acceder a toda la documentación desde [48].

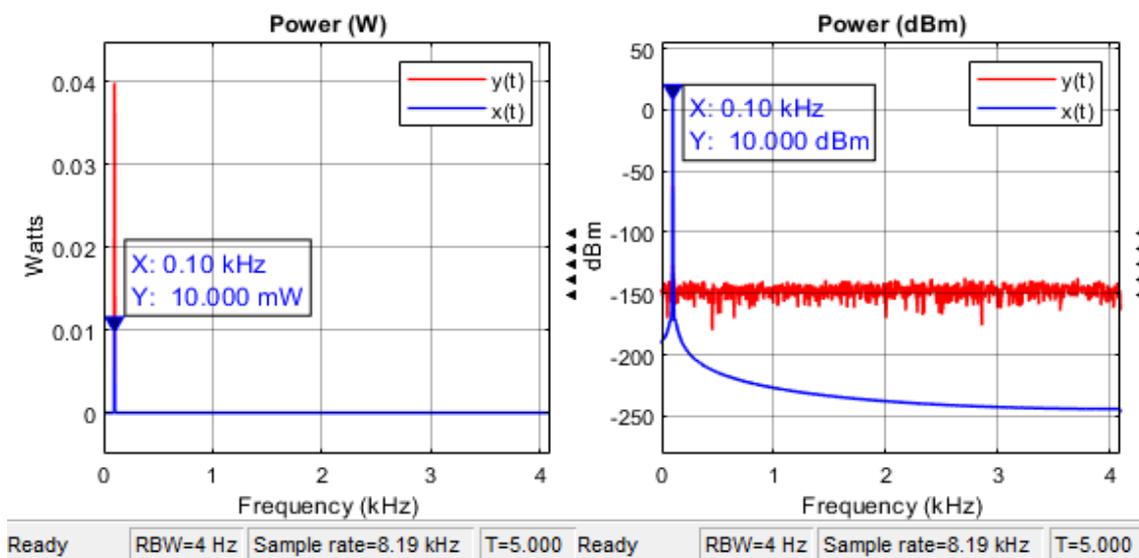
A continuación, se realizan los diferentes apartados que deben completar los alumnos para superar este laboratorio. Se utilizan los datos de la

Figura 73 y un tiempo de simulación de 5 segundos.

1) Anotar los valores de potencia de la señal $x(t)$ e $y(t)$ en W y dBm. Nota: utilizando la función *Trace Selection* del analizador de espectros se puede seleccionar la señal medida.

En la Figura 74 se muestran las dos gráficas correspondientes a los medidores de potencia para la señal $x(t)$. El ancho de banda de resolución RBW es de 4 Hz y el ancho de banda en el que se mide la potencia es de $f_s/2$. Estos valores son configurables desde las configuraciones de los bloques.

El valor de potencia para la señal $x(t)$ en este ejemplo es $P_x = 10 \text{ dBm}$, $10 \text{ mW} \rightarrow 0.01 \text{ W}$. Para calcular la potencia de la señal $y(t)$ hay que cambiar la traza en la que se realizan las medidas. Se obtiene: $P_y = 16 \text{ dBm}$, $39.8 \text{ mW} \rightarrow 0.0398 \text{ W}$.

Figura 74. Capturas de los medidores de potencia en dBm y W para $y(t)$

2) Anotar el valor de pico (V) de ambas sinusoides. Compruebe si el valor de pico de $y(t)$ coincide con el esperado, dada la ganancia del amplificador. Anote el periodo de la senoide (ms).

En la Figura 75 aparece la representación temporal de las señales. Para este apartado se obtiene que la amplitud de $y(t)$ es $A_y = 2 V$. Para realizar la comprobación hay que realizar la operación (45). El periodo se obtiene de los cursores, en este ejemplo el periodo es 10 ms.

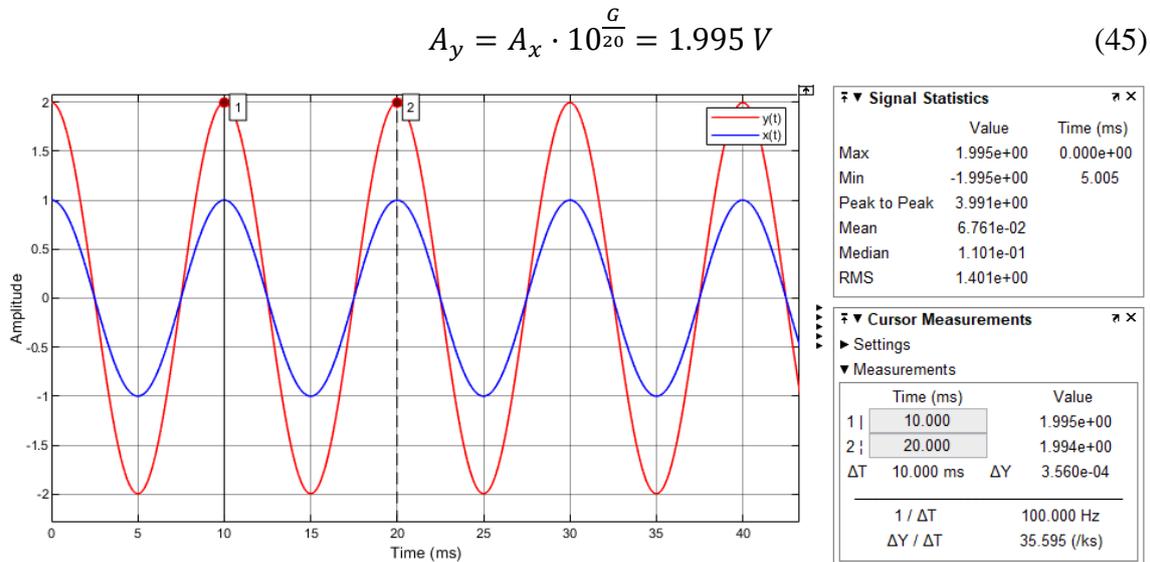


Figura 75. Osciloscopio con ambas señales y diferentes medidas

3) Anotar los valores de frecuencia y potencia de la delta que aparece en la gráfica de densidad espectral de potencia.

Se obtiene una frecuencia de 100 Hz y una densidad espectral de potencia de 3.979 dBm/Hz para $x(t)$ y de 9.979 dBm/Hz para $y(t)$ con un ancho de banda de resolución de 4 Hz y un ancho de banda de $f_s/2$.

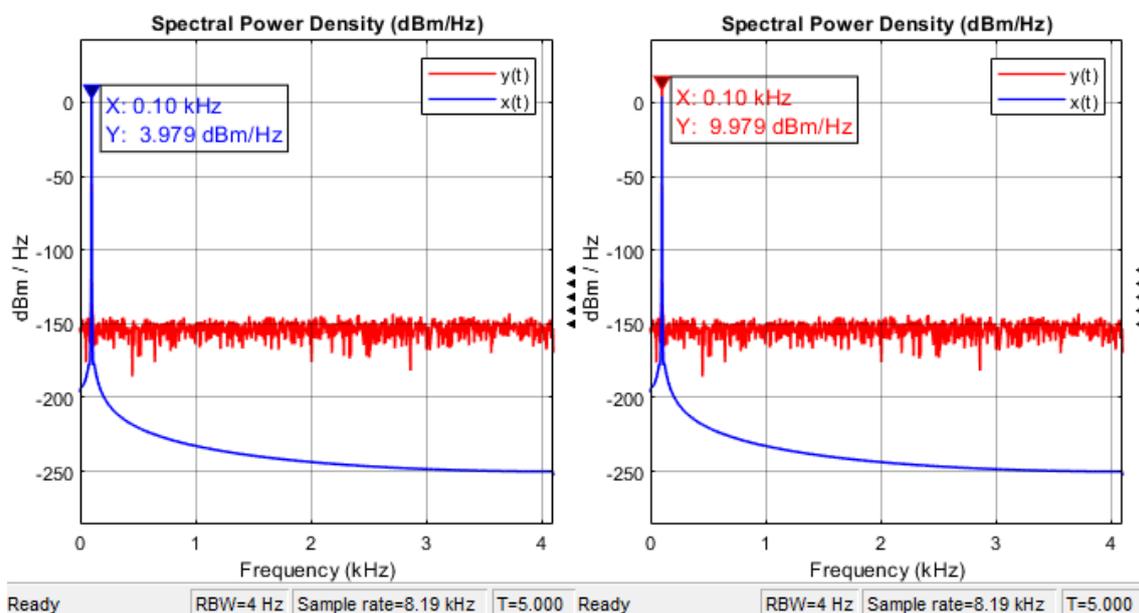


Figura 76. Gráfica de densidad espectral de potencia en dBm/Hz

4) En la práctica original se pide utilizar la línea de comandos de MATLAB para calcular distintos parámetros y que así el alumno se familiarice con las funciones que existen en MATLAB. Los parámetros que se piden son: valor de pico de la señal $y(t)$ en V, valor medio de la señal $y(t)$ en V, valor eficaz de $x(t)$ e $y(t)$ en V, valor cuadrático medio de $x(t)$ e $y(t)$ en V^2 e indicar cómo se obtiene la ganancia del amplificador a partir de este valor. Obtener la potencia calculandola de forma manual utilizando la impedancia. Por tanto, para obtener los valores descritos a continuación se puede hacer de dos maneras a elección del profesor:

En primer lugar, mantener la práctica ya existente en la que se pide utilizar las funciones las funciones `max`, `mean`, `std` y `var` para realizar cálculos de estos parámetros elementales y se pide al alumno indicar el código empleado y el valor resultante. Para trabajar con las señales $x(t)$ e $y(t)$ en la línea de comandos hay que utilizar `out.x` o `out.y`.

La segunda forma consiste en obtener los parámetros directamente de las gráficas mostradas en Simulink.

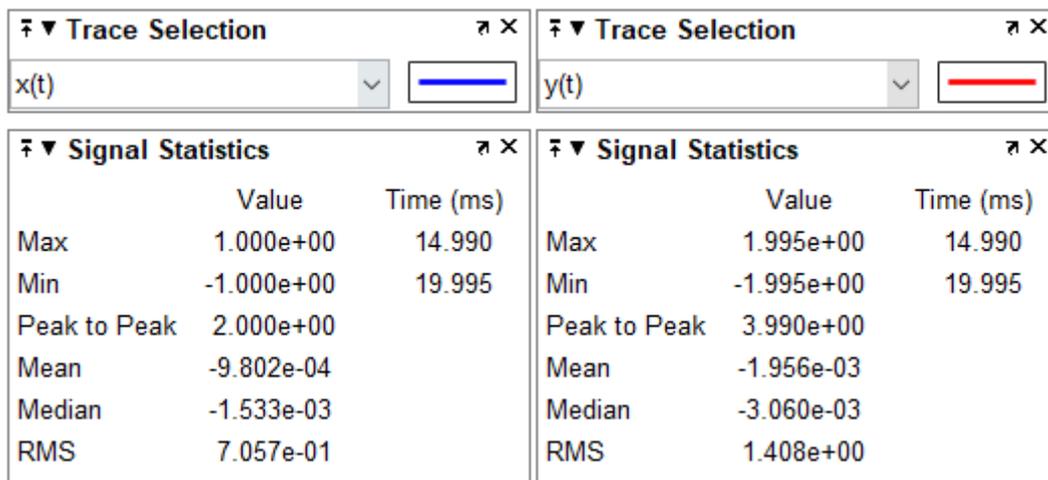


Figura 77. Parámetros obtenidos en Simulink utilizando Time Scope

Puede comprobarse que la potencia de las señales es igual a $P(W) = \frac{RMS^2}{Z}$

5.2 Práctica 2. Análisis de Señal y Ruido

Para la segunda práctica se amplía el proyecto de Simulink de la práctica 1 para aumentar la complejidad del sistema de forma progresiva. El objetivo de esta práctica es comprender cómo afectan ciertos fenómenos como son el ruido y la atenuación a un sistema de comunicaciones y afianzar conceptos como la densidad espectral de potencia de ruido y la potencia de las señales.

En esta práctica se necesita introducir los parámetros de la Tabla 16. A diferencia de la práctica 1, en este caso es necesario poner los parámetros en cada bloque por separado a excepción de los parámetros generales del sistema (impedancia y frecuencia de muestreo).

Tabla 16. Parámetros de la práctica 2

Bloque	Parámetros	Valores en el ejemplo	
Sistema	Frecuencia de muestreo, f_s en Hz.	8192 Hz	
	Valor de la impedancia Z del sistema.	50 Ω	
Tonos	Frecuencia de la senoide, f_0 en Hz.	700 Hz	1200 Hz
	Valor de pico de la senoide, A en V.	1 V	1 V
	Fase inicial de la senoide, θ en rad.	0 rad	0 rad
Atenuador	Valor de la atenuación ATN en dB.	9 dB	
AWGN	Densidad de potencia de ruido n_0 en W/Hz.	5e-8 W/Hz	

El circuito implementado es el de la Figura 78 en el que se tienen dos generadores de señales sinusoidales, una de las cuales es atenuada. Tras esto se suman ambas señales y se le añade ruido para estudiar cómo afecta éste a la señal. Para el apartado 3 se necesitará definir un ancho de banda en el que calcular la potencia de ruido alrededor de una frecuencia central. Estos valores son B_AWGN y fo_AWGN, en el ejemplo se utilizan los valores de la Figura 78.

LAB 2: SIGNAL AND NOISE ANALYSIS

IMPEDANCE, Z: Ohm

fo_AWGN=2000 Hz B_AWGN=800 Hz

SAMPLING FREQUENCY, fs: Hz

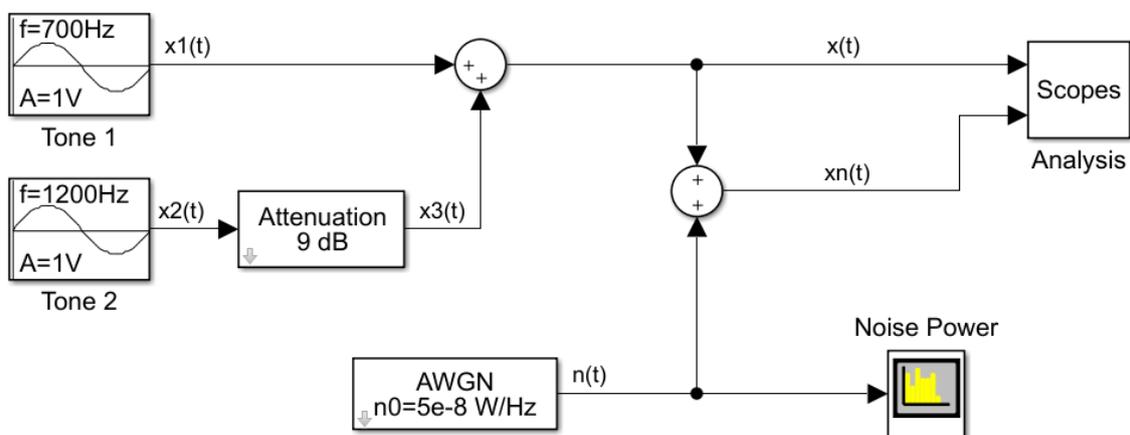


Figura 78. Sistema de la práctica 2

1) Calcule teóricamente (a partir de la amplitud A y la atenuación ATN) la potencia de ambas señales sinusoidales $x_1(t)$ y $x_3(t)$, así como del conjunto $x(t)$, en dBm. Compruebe que los valores calculados coinciden con los de la gráfica Power (dBm), Figura 79.

$$p_{x1} = \frac{A^2}{2Z} = 0.010 \text{ W} \rightarrow P_{x1} = 10 \log(p_{x1}(mW)) = 10 \text{ dBm}$$

$$p_{x2} = \frac{A^2}{2Z} = 0.010 \text{ W} \rightarrow P_{x2} = 10 \log(p_{x1}(mW)) = 10 \text{ dBm}$$

$$P_{x3} = P_{x2} - ATN = 10 - 9 = 1 \text{ dBm}$$

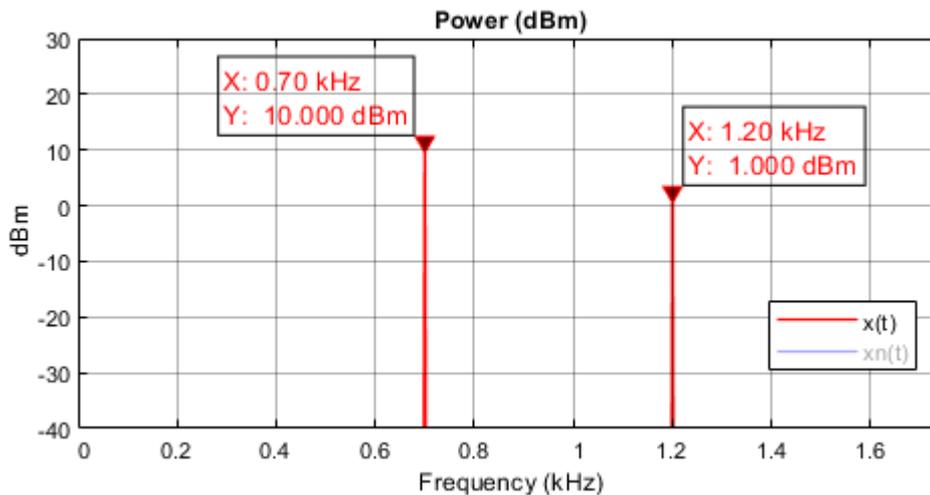


Figura 79. Potencia de cada senoide

La potencia total se puede obtener de la Figura 80. De esta manera se calcula la potencia en un ancho de banda que puede definir el usuario.

$$p_x = p_{x1} + p_{x2} = 0.0113 \text{ W}$$

$$P_x(\text{dBm}) = 10 \log(p_x(mW)) = 10.515 \text{ dBm}$$

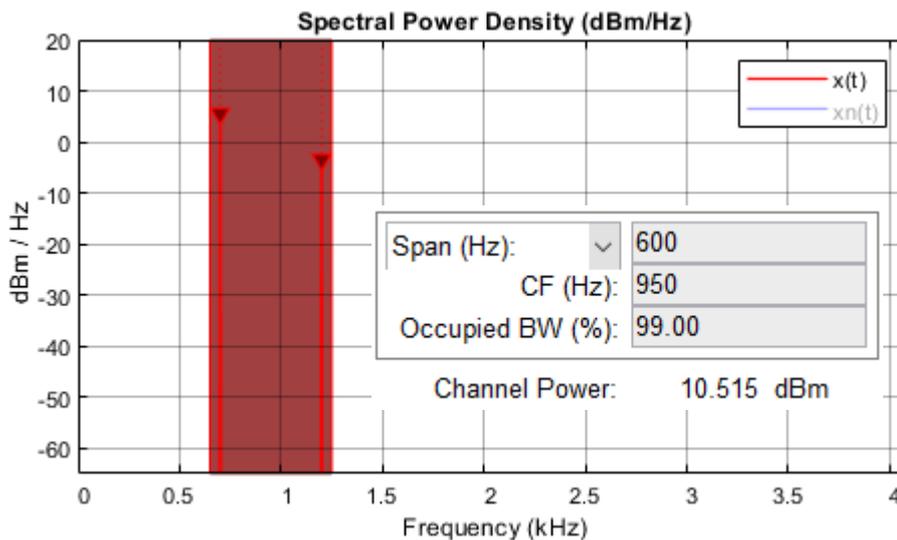


Figura 80. Potencia de las sinusoides

2) A partir del valor de la densidad espectral de ruido, n_0 , determine la potencia total de ruido (en toda la banda, desde 0 a $f_s/2$).

$$n(W) = n_0 \left(\frac{W}{\text{Hz}} \right) \cdot \frac{f_s}{2} (\text{Hz}) = 5 \cdot 10^{-8} \cdot 4096 = 2.048 \cdot 10^{-4} W$$

$$N(\text{dBm}) = 10 \log(n(\text{mW})) = -6.89 \text{ dBm}$$

3) Calcule teóricamente la potencia de ruido en un ancho de banda B_{AWGN} alrededor de la frecuencia f_{AWGN} . Compare ese valor con lo obtenido en la Figura 81.

En este ejemplo se utiliza un ancho de banda B_{AWGN} de 800 Hz y una frecuencia central f_{AWGN} de 2000 Hz.

$$n(W) = n_0 \left(\frac{W}{\text{Hz}} \right) \cdot \frac{f_s}{2} (\text{Hz}) = 5 \cdot 10^{-8} \cdot 800 = 4 \cdot 10^{-5} W$$

$$N(\text{dBm}) = 10 \log(n(\text{mW})) = -13.97 \text{ dBm}$$

El valor teórico puede no coincidir con el obtenido en la simulación debido a que en cada ejecución del programa sale un resultado diferente. La causa de esto es la aleatoriedad del ruido. En el siguiente apartado de la práctica se realiza un promediado, en la banda completa, para obtener un resultado más preciso.

En la Figura 81 se muestra la densidad espectral de potencia en el ancho de banda B_{AWGN} de 800 Hz y de frecuencia central f_{AWGN} de 2000 Hz. Debajo de la gráfica se muestran dos capturas de los valores obtenidos en dos ejecuciones diferentes. Como puede observarse, este valor se modifica en cada ejecución.

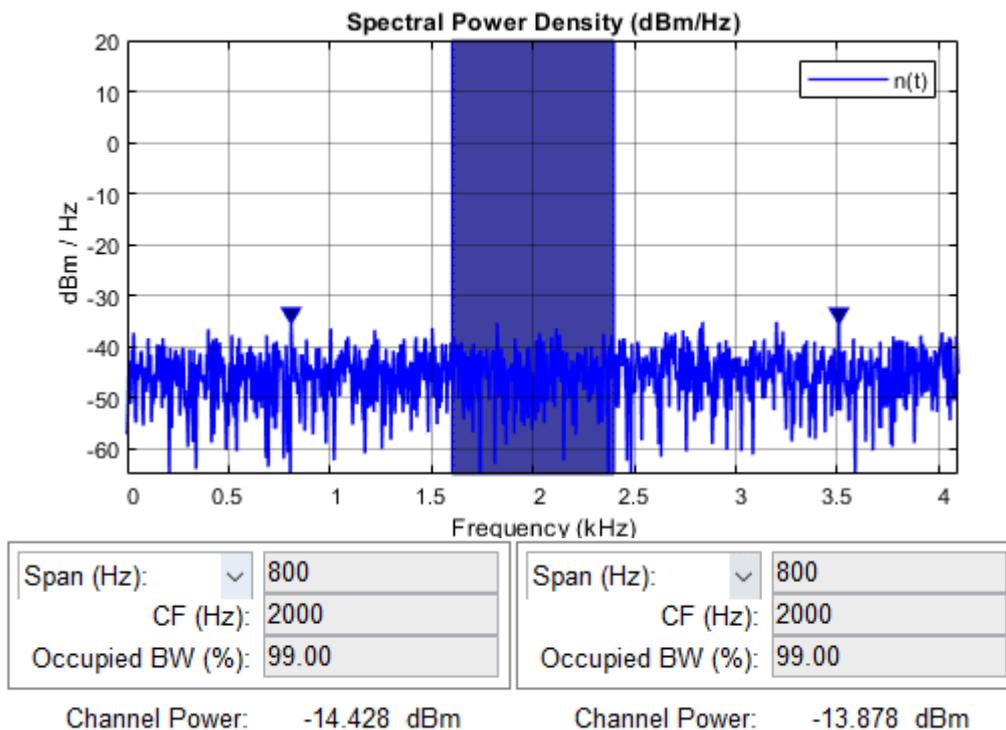


Figura 81. Potencia de ruido en un ancho de banda de 800 Hz

4) Al promediar bastantes espectros (inicialmente 500 espectros promediados, pero puede incrementar este número) el ruido presentará una densidad espectral de potencia razonablemente plana. El tono no se ve afectado por el promediado. Anote el valor aproximado de la densidad espectral de ruido (suelo de ruido). Compruebe que coincide aproximadamente con n_0 .

El cálculo teórico indica que $n_0 = -43.01 \text{ dBm/Hz}$. Se puede obtener de la gráfica el valor que sale para esta simulación después de 500 espectros promediados y comprobar que este resultado se aproxima mucho al valor anterior.

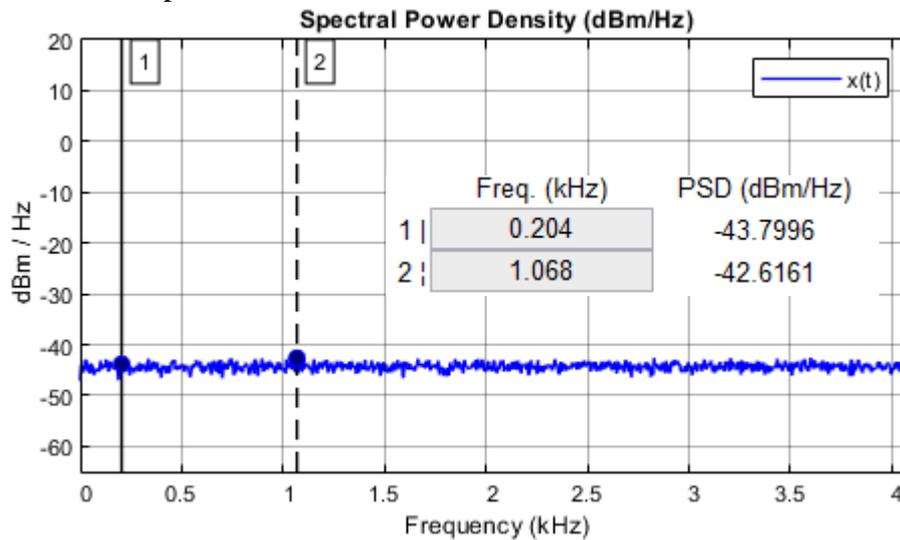


Figura 82. Densidad espectral de potencia de ruido

5) A partir del vector ruido n del entorno Matlab, calcule utilizando comandos de MATLAB similares a los utilizados en la práctica 1:

- Valor medio del ruido (debería salir un valor próximo, pero no igual, a cero)
- Valor eficaz del ruido (compruebe que es similar, pero no igual, a n_{ef})
- Potencia total de ruido (debería coincidir con el valor del apartado 2)

Al igual que en el apartado 4 de la práctica 1, se pueden obtener estos valores de los analizadores de Simulink o a partir de comandos de MATLAB. Las señales generadas se pueden utilizar desde la variable `out`. Las señales se llaman `out.n`, `out.x`, `out.xn`.

5.3 Práctica 3. Modulaciones AM, DSB y SSB

El objetivo de esta práctica es generar modulaciones analógicas de tipo AM, DSB y SSB y realizar medidas de diferentes parámetros. En el proyecto se generan estas modulaciones en banda base con las expresiones ya explicadas en el apartado 4.6 de este documento. En esta práctica se trabaja con el programa completo y no con versiones reducidas de éste, como sucede en las prácticas 1 y 2. Los bloques necesarios para la realización de esta práctica son los mostrados en el diagrama de bloques de la Figura 83.

La fuente de señal es de un solo tono. En este caso se utiliza un bloque que se ha creado para este proyecto, Analog Data Source, explicado en el apartado 4.1. El bloque se ha programado para que solo entregue un tono a la salida.

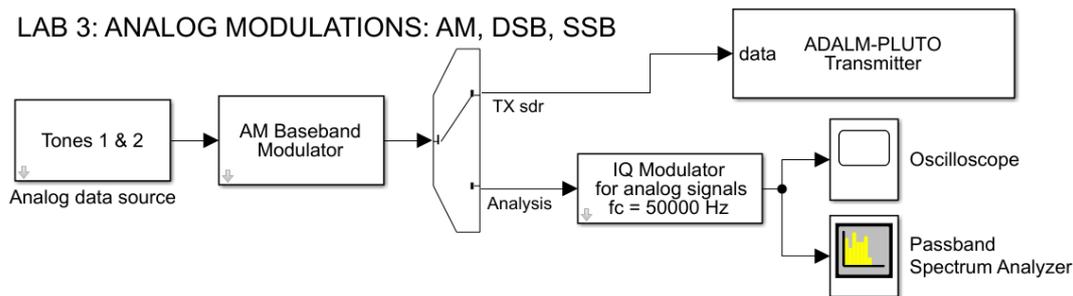


Figura 83. Diagrama de bloques de la práctica 3

La forma de realizar esta práctica es que el profesor utilice el proyecto y genere las señales en banda base y las envíe utilizando el SDR Adalm-Pluto. Tras esto los alumnos reciben la señal en sus correspondientes SDR y analizan los parámetros pedidos. Esta práctica también se puede realizar sin SDR. En este caso, se sustituye el SDR utilizando un modulador IQ implementado en el proyecto. La amplitud de la señal portadora se puede modificar desde la configuración del bloque modulador IQ. Esto permite cambiar la potencia de salida del sistema. De esta forma, el alumno podrá ver los resultados en paso banda desde Simulink. La amplitud de la moduladora está normalizada.

Tabla 17. Parámetros de la práctica 3

Bloque	Parámetros	Valores en el ejemplo
Sistema	Frecuencia de muestreo, f_s en Hz.	500 kHz
	Valor de la impedancia Z del sistema.	50 Ω
Tonos	Frecuencia de la senoide, f_0 en Hz.	10 kHz
	Valor de pico de la senoide, A en V.	1 V
	Fase inicial de la senoide, θ en rad.	0 rad
Mod. AM	Índice de modulación m .	0.5
Mod. IQ	Frecuencia del oscilador local, f_c en Hz.	50 kHz
	Amplitud de la portadora, A_c	1 V

1) De acuerdo con el diagrama de bloques del modulador, y usando las ecuaciones de los fundamentos teóricos, calcule los valores de los siguientes parámetros. Se ha elegido una amplitud de portadora igual a 1 voltio:

- Valor máximo y mínimo de la envolvente de la señal modulada.

$$A_{max} = A_c(1 + m) = 1.5 \text{ V} \quad A_{min} = A_c(1 - m) = 0.5 \text{ V}$$

- Potencia de la portadora sola P_c .

$$p_c = \frac{A^2}{2Z} = 0.01 W \rightarrow P_c = 10 \text{ dBm}$$

- Potencia de una banda lateral P_{BL} .

$$p_{BL} = \frac{A^2 \cdot m^2}{8Z} = 6.25 \cdot 10^{-4} W \rightarrow P_{BL} = -2.04 \text{ dBm}$$

- Potencia total de la señal modulada P_t .

$$p_t = p_c + 2 \cdot p_{BL} = 0.0113 W \rightarrow P_t = 10.5 \text{ dBm}$$

2) Indicar los valores máximos y mínimos de la envolvente de la señal modulada $y(t)$. Compruebe que coinciden con los valores teóricos. En la Figura 84 se muestran la señal modulada en AM en el dominio del tiempo junto con la envolvente. Utilizando los cursores se puede obtener el valor de la amplitud de la envolvente.

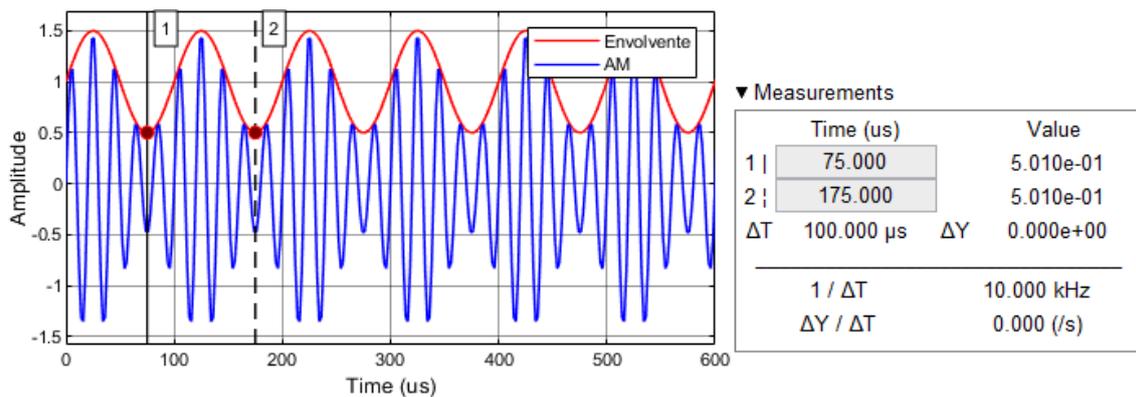


Figura 84. Señal modulada en AM y su envolvente

3) Indique los valores de potencia (portadora sola, banda lateral, potencia total) obtenidos de la gráfica. Compruebe que coinciden con los valores del apartado 1. Explique posición en frecuencia de las bandas laterales y de la portadora.

De la Figura 85 se pueden obtener las potencias e información en frecuencia. En esta figura aparecen la señal portadora centrada en 50 kHz y las dos bandas laterales en 40 y 60 kHz, que son las que contienen la información de la señal moduladora. La posición de las bandas laterales se corresponde con la frecuencia de la señal moduladora.

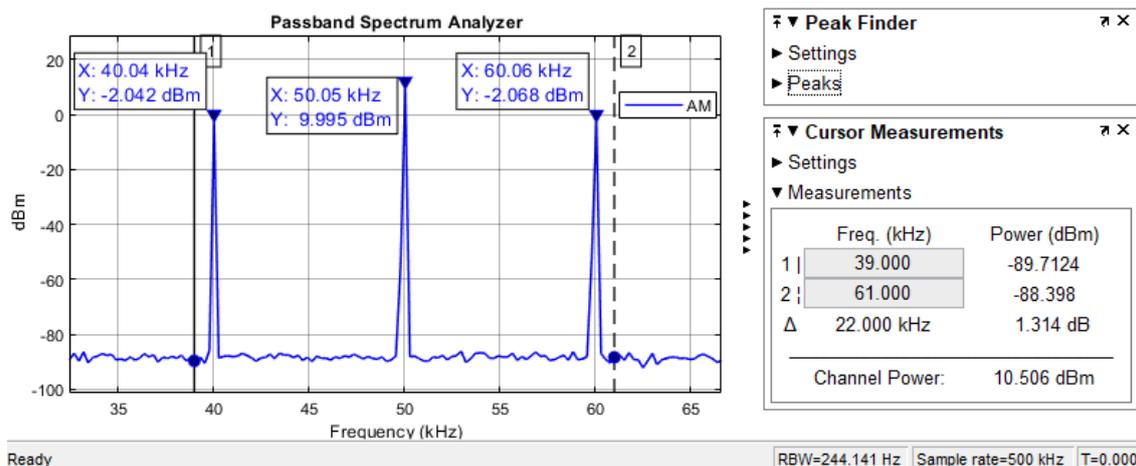


Figura 85. Espectro de un tono modulado en AM

Para medir la potencia se ha utilizado un ancho de banda de 22 kHz, tal y como se indica en la sección *Cursor Measurements* del analizador de espectros. El ancho de banda de resolución utilizado es de 244.14 Hz. Este valor se pone de forma automática, aunque el usuario puede cambiarlo de forma sencilla desde la pestaña de configuración del bloque *Spectrum Settings > Main Options*.

4) Una vez entendida la modulación AM se propone que los alumnos conozcan las modulaciones DSB y SSB. Es de gran interés la forma del espectro y los valores de potencia de las bandas laterales. Para ello se utilizan los mismos datos y valores que para la modulación AM.

En la Figura 86 se muestra el espectro de un tono modulado en DSB. Puede apreciarse que en este caso no hay una señal portadora en 50 kHz. Solo aparecen las bandas laterales en 40 y 60 kHz. Es en estas bandas donde se lleva la información de la señal moduladora. Los valores de ancho de banda de resolución y de ancho de banda en el que se mide la potencia son los mismos que para el caso de AM.

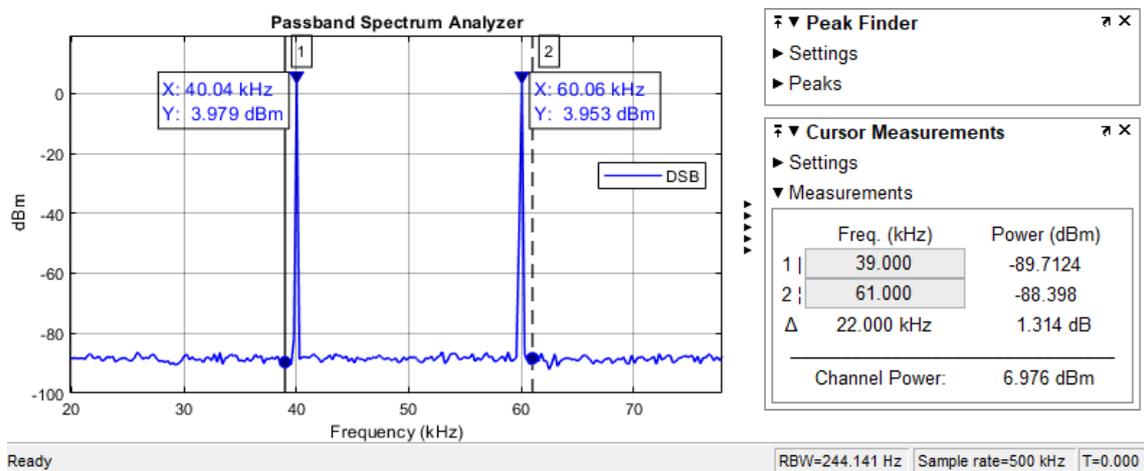


Figura 86. Espectro de un tono modulado en DSB

En la Figura 87 se muestra el tono modulado en SSB-LSB. En este caso, desaparece la portadora y también una de las bandas laterales. La banda lateral suprimida aparece en algunas ocasiones en el espectro con un valor muy bajo en comparación con la banda utilizada, esto se debe a que la implementación de SSB no es ideal.

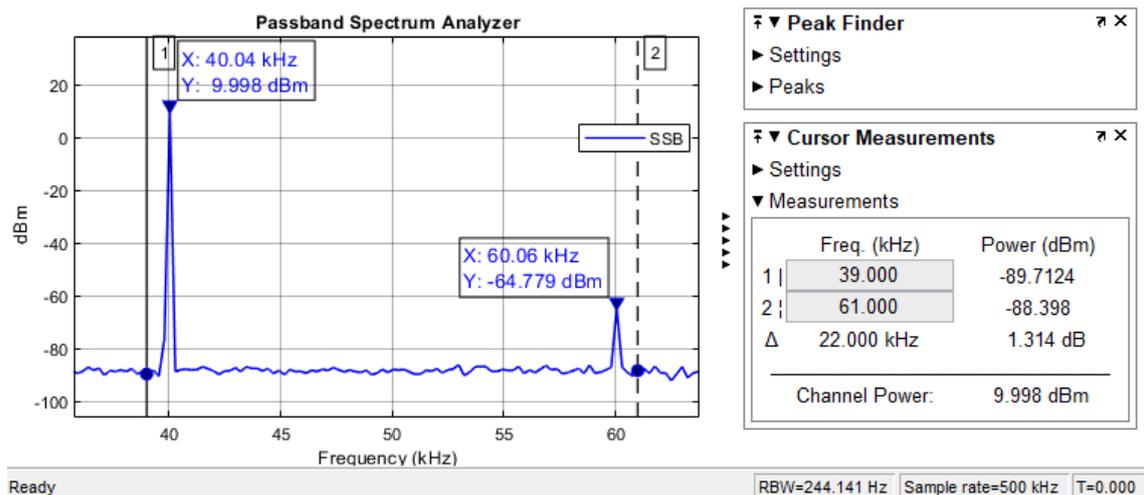


Figura 87. Espectro de un tono modulado en SSB-LSB (Lower Side Band)

Una vez obtenidos los valores de potencia se puede calcular la eficiencia de cada una de estas modulaciones. La eficiencia espectral se define como:

$$\eta = \frac{P_{util}}{P_{tot}} \quad (46)$$

La potencia útil se define como la potencia contenida en una única banda lateral. Esto se hace así porque una banda contiene toda la información necesaria para recuperar la información de la señal moduladora en el receptor.

Calculamos los valores para cada modulación:

$$\eta_{AM} = \frac{1.242}{11.23} = 11.1 \% ; \eta_{DSB} = \frac{2.48}{4.98} = 49.85 \% ; \eta_{SSB} = \frac{9.99}{9.99} \approx 100 \%$$

5) El último apartado de la práctica consiste en transmitir señal con el SDR y recibir en un analizador de espectros. El montaje de la práctica es el de la Figura 88.



Figura 88. Montaje para ver la señal de salida en el analizador de espectros

En la Figura 89 se muestra una captura de pantalla del analizador de espectros con los datos de la práctica para la modulación AM. Los datos son los mismos excepto la frecuencia de portadora que será siempre 433.5 MHz, ya mencionada en el apartado 3.1.1.

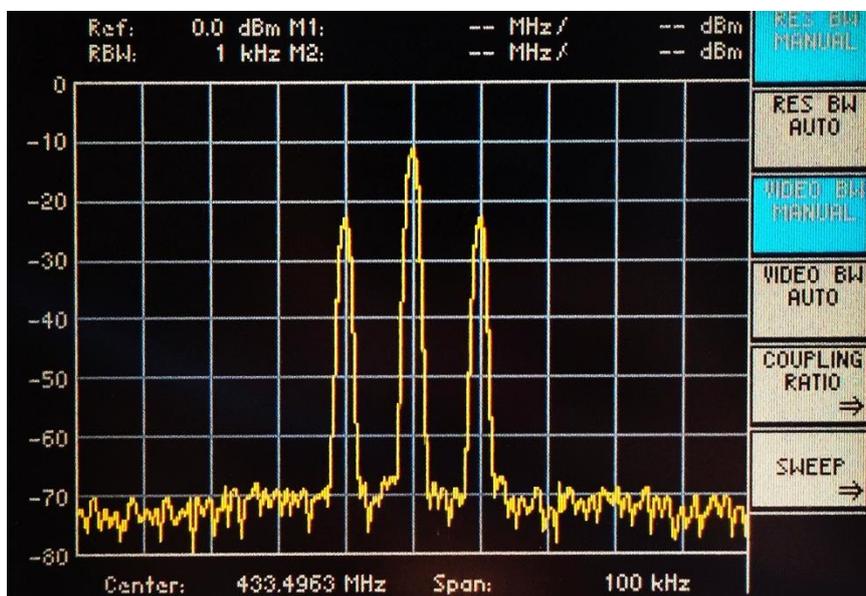


Figura 89. Tono de 10kHz modulado en AM

En la Figura 90 se muestra una captura de pantalla del analizador de espectros con los datos de la práctica para la modulación DSB. Desaparece prácticamente la portadora de AM y se mantienen las dos bandas laterales a 10kHz.

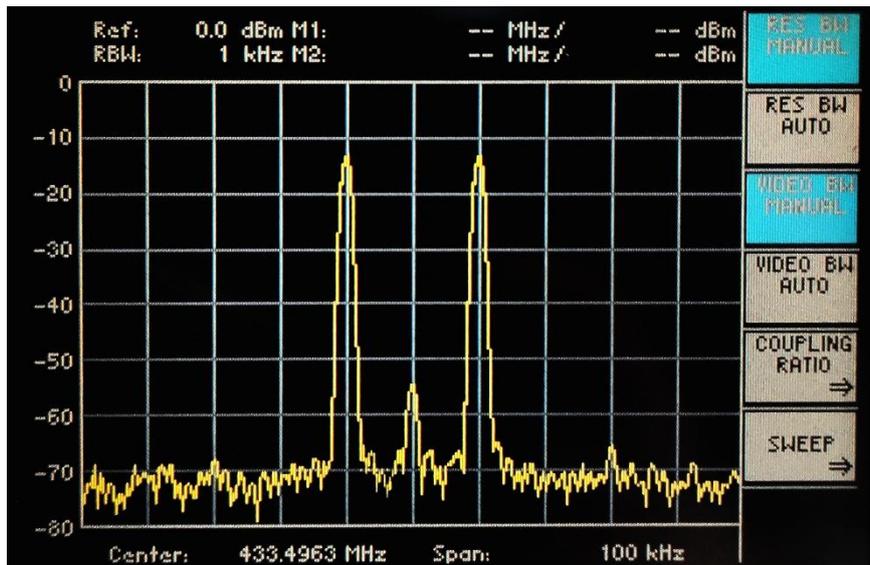


Figura 90. Tono de 10kHz modulado en DSB

En la Figura 91 se muestra una captura de pantalla del analizador de espectros con los datos de la práctica para la modulación SSB. La atenuación de la banda no deseada es mala porque se está utilizando un tono de baja frecuencia en comparación con la frecuencia de muestreo y el filtro que genera la SSB no es capaz de filtrarlo bien.

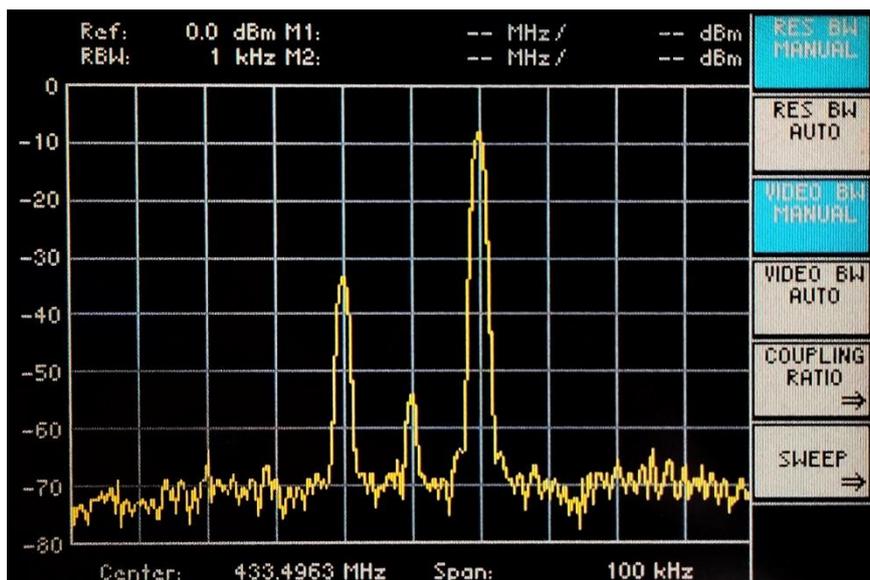


Figura 91. Tono de 10kHz modulado en SSB-USB

Para solucionar este problema se propone reducir la frecuencia de muestreo o aumentar la frecuencia del tono para que, de esta manera, el filtro sea más estrecho y pueda eliminar la banda no deseada lo máximo posible, o bien, que el tono caiga en la zona de filtrado. Para demostrar este efecto se han obtenido nuevas capturas modificando los parámetros utilizados.

En la Figura 92 y Figura 93 se han utilizado los siguientes datos: frecuencia de muestreo $f_s = 2 \text{ MHz}$, frecuencia del tono 500 kHz . Se puede observar que en estos casos la banda no deseada se atenúa por completo. En estas capturas solo está la banda lateral deseada a 500 kHz de la frecuencia de la portadora.

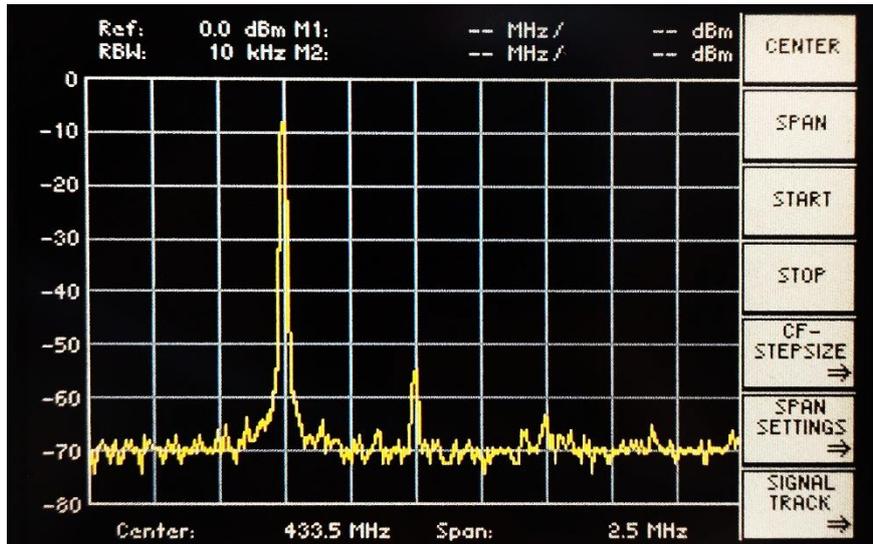


Figura 92. Tono de 500 kHz modulado en SSB-LSB

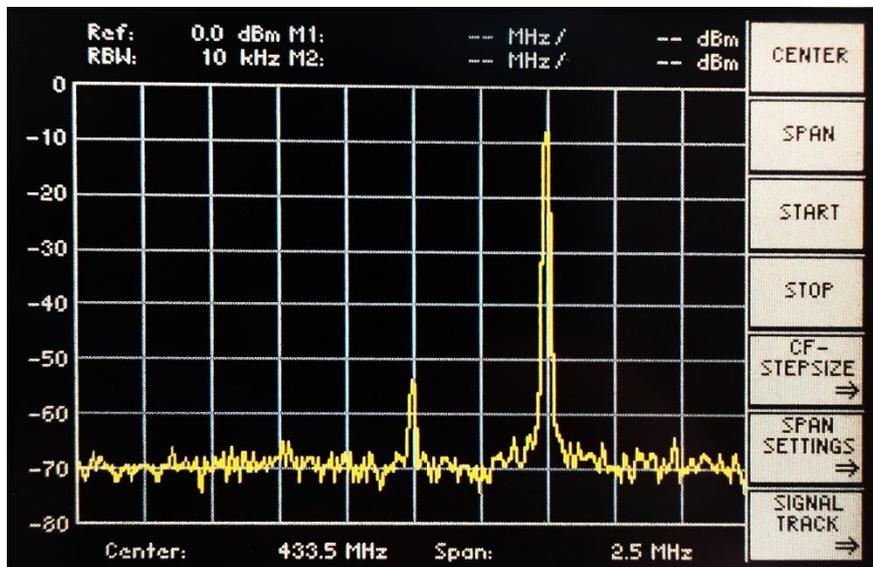


Figura 93. Tono de 500 kHz modulado en SSB-USB

5.4 Práctica 4. Modulación FM

En esta práctica se estudia la modulación FM de banda ancha y de banda estrecha y los conceptos relacionados con esta modulación como, por ejemplo, el ancho de banda de Carson, la desviación en frecuencia de la modulación y el índice de modulación para el caso de una señal moduladora que sea un tono sinusoidal. Para ello se utiliza el mismo diagrama de bloques que en la práctica anterior. Los parámetros son los descritos en la Tabla 18.

Tabla 18. Parámetros de la práctica 4

Bloque	Parámetros	Valores en el ejemplo
Sistema	Frecuencia de muestreo, f_s en Hz.	500 kHz
	Valor de la impedancia Z del sistema.	50 Ω
Tonos	Frecuencia de la senoide, f_m en Hz.	3 kHz
	Valor de pico de la senoide, A en V.	1 V
	Fase inicial de la senoide, θ en rad.	0 rad
Mod. FM	Desviación de frecuencia, Δf en Hz	8 kHz
Mod. IQ	Frecuencia del oscilador local, f_c en Hz.	70 kHz
	Amplitud de la portadora, A_c	1 V

Al igual que en la práctica 3, la amplitud de la señal modulada está normalizada, por tanto, la amplitud siempre es 1. Para aumentar la potencia de salida del sistema es necesario modificar la amplitud de la portadora A_c desde el bloque del modulador IQ.

Parte 1. FM de banda ancha.

1) Calcule los valores de:

- Potencia media de la señal FM: $P_y(\text{dBm}) = 30 + 10 \log\left(\frac{A^2}{2Z}\right) = 10 \text{ dBm}$
- Índice de modulación: $\beta = \frac{\Delta f}{f_m} = 2.667$
- Ancho de banda de Carson: $B_C = 2 \cdot (\Delta f + f_m) = 22 \text{ kHz}$

2) Visualice la densidad espectral de potencia de la señal FM. Indique el número de deltas significativas (por encima del suelo de ruido) así como su separación en frecuencia.

La Figura 94 muestra la señal en FM. La separación en frecuencia entre deltas es de 3 kHz, correspondiente al tono modulador. La frecuencia de la delta central es 70 kHz.

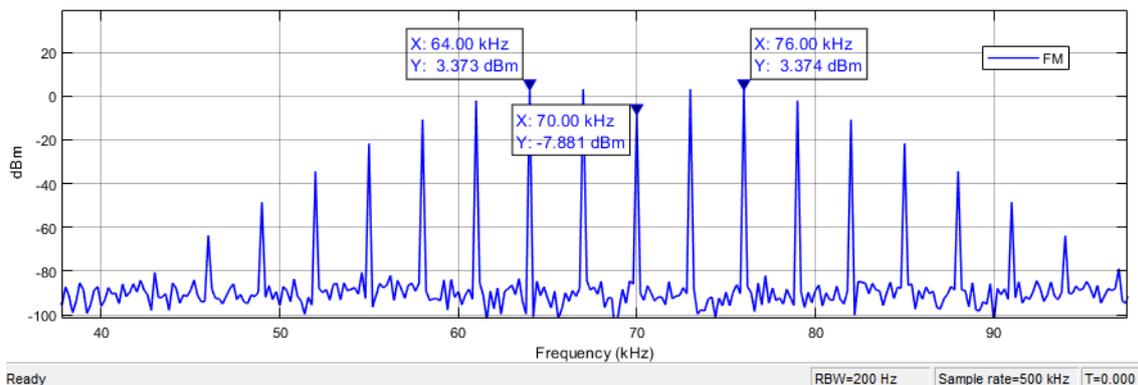


Figura 94. Espectro de un tono de 3 kHz modulado en FM con $\Delta f = 8 \text{ kHz}$

3) Con la gráfica de la Figura 94 mida la delta situada en f_c . Compruebe que su potencia p_0 coincide con el resultado de las fórmulas teóricas. Se puede obtener el valor de la función de Bessel utilizando la función `besselj(0, beta)` de MATLAB.

El valor obtenido en la gráfica para la delta en f_c es de -7.881 dBm. Realizando el desarrollo teórico llegamos al mismo resultado.

$$p_0(W) = \frac{A^2}{2Z} \cdot J_0^2(\beta) = 0.1627 \text{ mW} \rightarrow P_0 = -7.885 \text{ dBm}$$

4) Con la función *Channel Measurements*  del analizador de espectros mida la potencia total contenida dentro del ancho de banda de Carson y la potencia total en todo el ancho de banda. ¿Qué porcentaje representa respecto a la potencia total de la señal FM? ¿Se cumple la regla de Carson?

En la Figura 95 se muestra la medida de la potencia p_0 contenida en el ancho de banda de Carson. La potencia del ancho de banda completo se puede medir utilizando la misma herramienta. El porcentaje que representa es:

$$P_0 = 9.921 \text{ dBm} \rightarrow p_0 = 9.8197 \text{ mW}; P_T = 10 \text{ dBm} \rightarrow p_T = 10 \text{ mW}$$

$$\frac{p_0}{p_T} \cdot 100 = 98.197\%$$

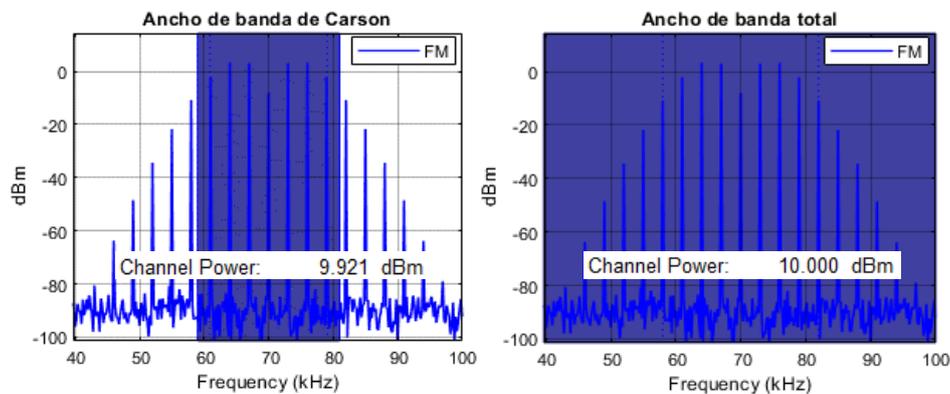


Figura 95. Potencia en el ancho de banda de Carson y total para FM banda ancha.

5) Obtener la señal en el tiempo utilizando la función `viewFMwaveform` desde la consola de comandos de MATLAB. Medir el periodo máximo y mínimo y obtener a partir de ello la máxima desviación en frecuencia. Introducir el siguiente comando `viewFMwaveform(y, sdrVar.Fs, fc)` para obtener la representación temporal de la señal mostrada en la Figura 96. Es necesario guardar la señal modulada en el espacio de trabajo.

$$\Delta f = \frac{1}{2} \left(\frac{1}{T_{\min}} - \frac{1}{T_{\max}} \right) = \frac{78118 - 62192}{2} \approx 8000 \text{ Hz}$$

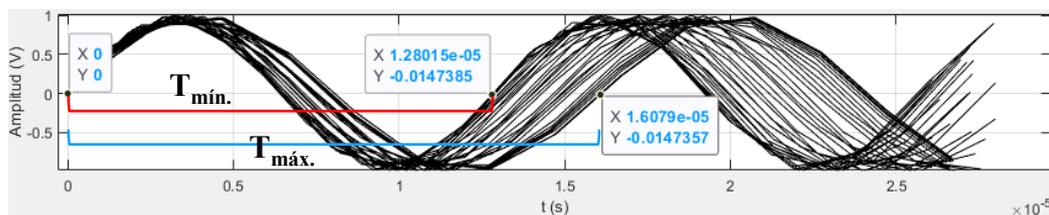


Figura 96. Representación temporal de un tono de 3 kHz en FM banda ancha

Parte 2. FM de banda estrecha ($\beta=0.1$)

Antes de comenzar con esta parte, ajuste el valor de la desviación de frecuencia para que el índice de modulación β sea 0.1 y anótelo. En este ejemplo, el valor de la desviación en frecuencia puede obtenerse con la fórmula del índice de modulación:

$$\beta = \frac{\Delta f}{f_m} \rightarrow \Delta f = \beta \cdot f_m = 300 \text{ Hz}$$

6) Visualice el espectro de la señal modulada FM e indique el número de deltas significativas, así como su separación.

En este ejemplo, la separación entre las deltas es de 3 kHz y hay 3 deltas.

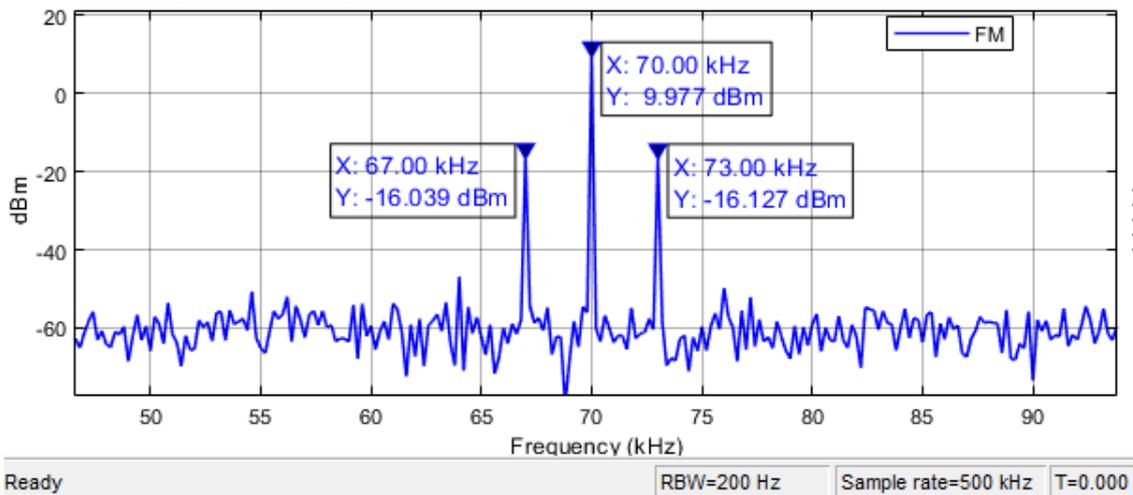


Figura 97. Espectro de FM banda estrecha para un tono de 3 kHz

7) Con la gráfica anterior compruebe que la potencia de la delta situada en $f_c + f_m$ es la misma que la calculada mediante la fórmula teórica. Utilizando la aproximación de las funciones de Bessel de primera especie para valores de β muy pequeños. También se puede obtener el valor de la función de Bessel utilizando la función `besselj(1, β)` de MATLAB.

Realizando los cálculos mediante la fórmula teórica se obtiene el siguiente valor para la delta situada en 73 kHz.

$$p_{f_c+f_m}(W) = \frac{A^2}{2Z} \cdot J_1^2(\beta) = 0.0249 \text{ mW}$$

$$P_{f_c+f_m} = -16.031 \text{ dBm}$$

Utilizando la aproximación para valores de β muy pequeños se puede conseguir un resultado muy próximo para la delta en 73 kHz.

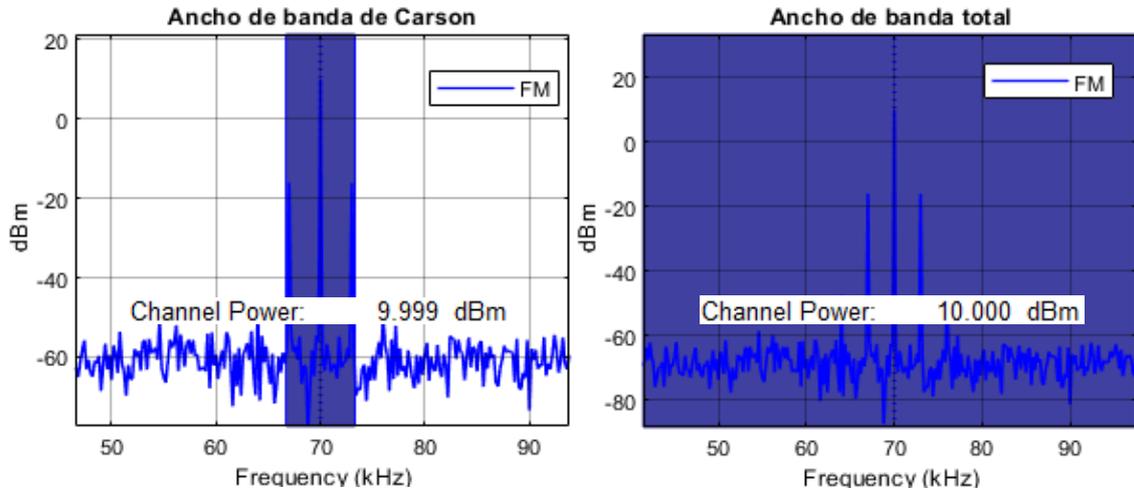
$$p_{aprox}(W) = \frac{A^2}{2Z} \cdot \left(\frac{\beta}{2}\right)^2 = 0.0250 \text{ mW}$$

$$P_{aprox} = -16.021 \text{ dBm}$$

Por tanto, se comprueba que, para este caso, la aproximación de las funciones de Bessel sirve para valores muy pequeños de β .

8) Determine el nuevo valor del ancho de banda de Carson y mida la potencia contenida dentro de él. ¿Qué porcentaje representa respecto a la potencia total de la señal?

El nuevo ancho de banda de Carson es $B_C = 2 \cdot (\Delta f + f_m) = 6.6 \text{ kHz}$. Siguiendo los mismos pasos que en el apartado 5, obtenemos la potencia del ancho de banda de Carson $P_{BC} = 9.999 \text{ dBm} \rightarrow 9.9977 \text{ mW}$ y también la potencia del ancho de banda total $p_T = 10.000 \text{ dBm} \rightarrow 10 \text{ mW}$. Tenem



os un 99.97%.

Figura 98. Potencia en el ancho de banda de Carson y total para FM banda estrecha.

9) Obtener la señal en el tiempo utilizando la función `viewFMwaveform` desde la consola de comandos de MATLAB. Medir el periodo máximo y mínimo y obtener a partir de ello la máxima desviación en frecuencia. Introducir el siguiente comando `viewFMwaveform(y, sdrVar.Fs, fc)` para obtener la representación temporal de la señal mostrada en la Figura 99. Es necesario guardar la señal modulada en el espacio de trabajo.

Para obtener una posición más exacta de los marcadores se recomienda hacer click derecho sobre la figura y darle a *Selection Style > Mouse Position*.

$$\Delta f = \frac{1}{2} \left(\frac{1}{T_{mín}} - \frac{1}{T_{máx}} \right) = \frac{70450 - 69550}{2} \approx 450 \text{ Hz}$$

En este caso, los resultados no son tan precisos como en el apartado anterior ya que tenemos una desviación en frecuencia de tan solo 300 Hz.

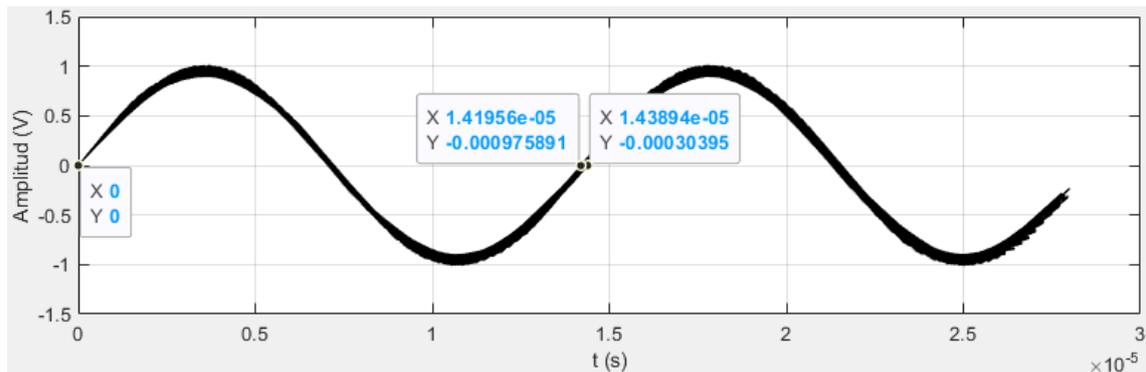


Figura 99. Representación temporal de un tono de 3kHz en FM banda estrecha

10) El último apartado de la práctica consiste en transmitir señal con el SDR y recibir en un analizador de espectros. El montaje de la práctica es el de la Figura 88.

Para FM de banda ancha se han utilizado los siguientes datos: desviación de frecuencia $\Delta f = 75 \text{ kHz}$, frecuencia de muestreo $f_s = 2 \text{ MHz}$ y frecuencia del tono 50 kHz . Se ha obtenido el espectro de la Figura 100.

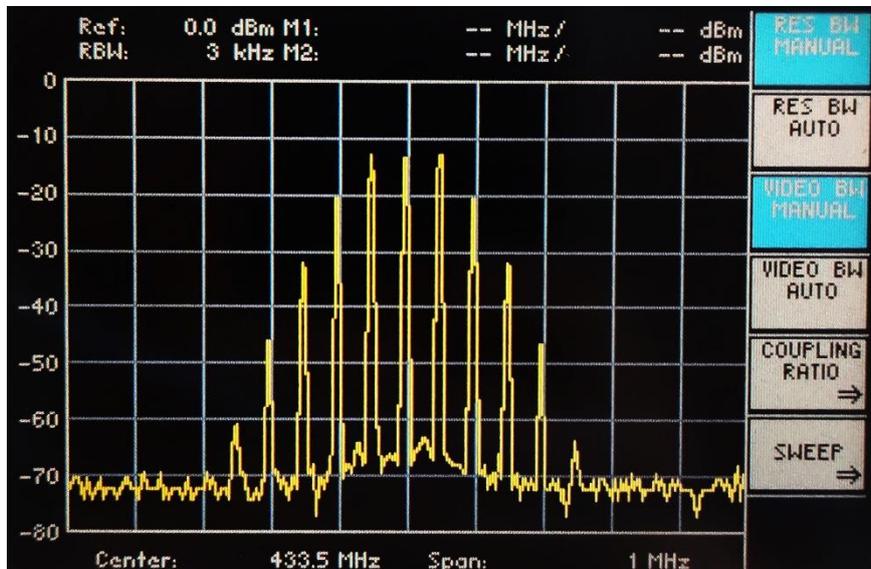


Figura 100. Tono de 50kHz modulado en FM de banda ancha

Para FM de banda estrecha se han utilizado los siguientes datos: desviación de frecuencia $\Delta f = 5 \text{ kHz}$, frecuencia de muestreo $f_s = 2 \text{ MHz}$ y frecuencia del tono 50 kHz . Se ha obtenido el espectro de la Figura 101.

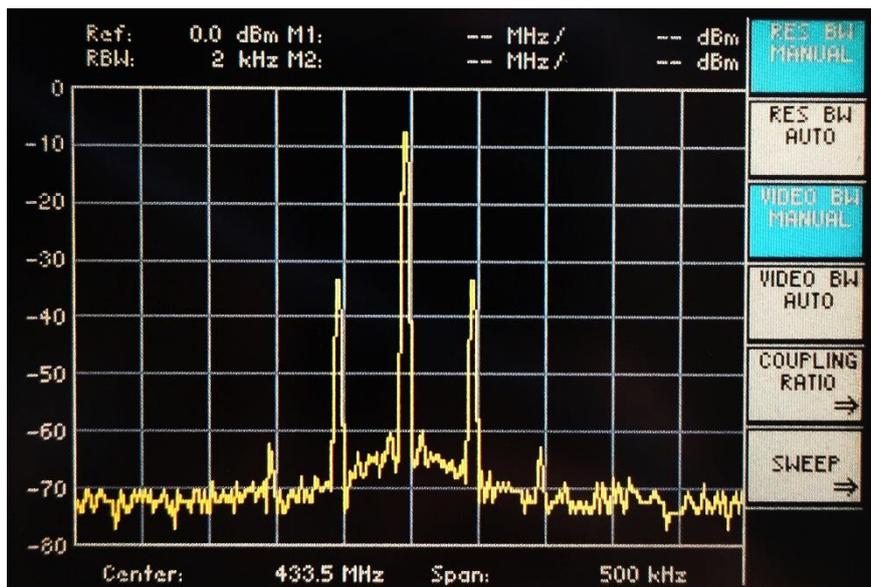


Figura 101. Tono modulado en FM de banda estrecha

5.5 Práctica 5. Modulaciones digitales

El objetivo de esta práctica es aplicar los conocimientos de modulaciones digitales y observar algunos casos de aplicación, los efectos que tiene el medio sobre la señal, el efecto de los códigos de corrección de errores y realizar cálculos de BER.

Se comienza observando las diferentes constelaciones, después se analizan los canales IQ y el espectro. Tras esto se aplicarán efectos de canal para ver los efectos en la constelación y en el espectro. Por último, se analiza cómo afecta el canal en el cálculo de la BER. Todo esto se hará en el espectro del equivalente paso bajo, al contrario que en analógico que se realizaba todo en paso banda.

Para los ejercicios de la práctica se van a utilizar los parámetros de la Tabla 19.

Tabla 19. Parámetros de la práctica 5

Bloque	Parámetros	Valores en el ejemplo
Sistema	Régimen de símbolo, R_s en Hz	200 kHz
	Nº de símbolos a la salida del modulador	5000 símbolos
	Factor de interpolación u <i>oversampling</i>	8
	Fuente de información digital	PRBS 16

1) A partir de las siguientes constelaciones, diga a qué modulación corresponde cada una de las imágenes. En caso de que haya alguna que pueda ser de varias modulaciones diferentes diga a cuáles se corresponde.

La constelación 1 se corresponde con una modulación 4-ASK. La constelación 2 con una modulación 8-PSK con desfase $\pi/8$. La constelación 3 puede ser de varios tipos de modulación, para distinguirlo sería necesario ver cuál es la trayectoria de la señal. En este caso se puede responder a la pregunta diciendo que al ver solo la constelación puede ser una MSK o una QPSK. La constelación 4 es una 64-QAM

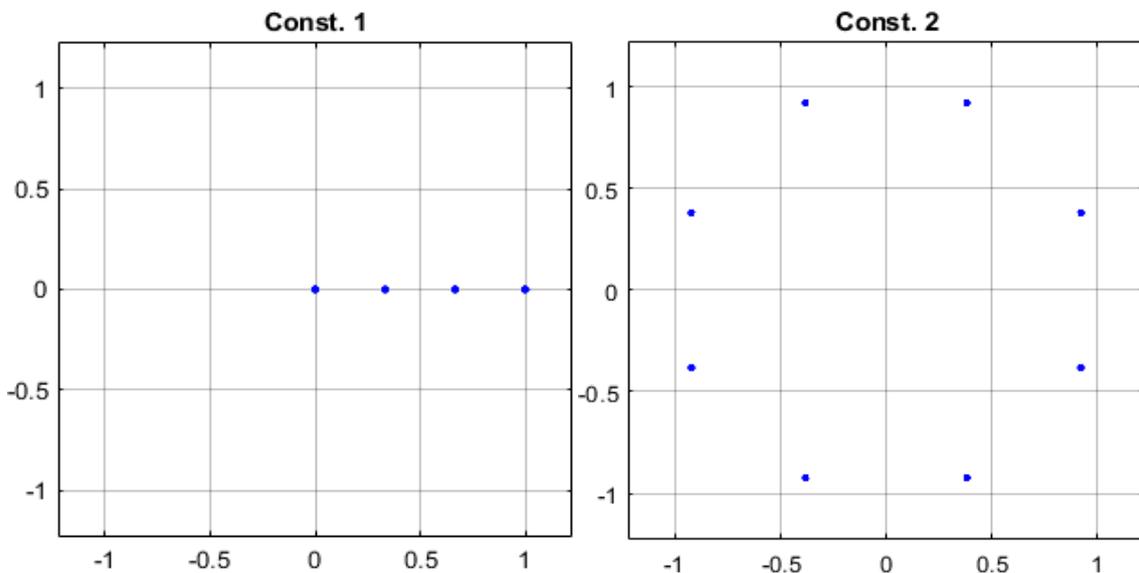


Figura 102. Constelaciones 1 y 2 de la práctica 5

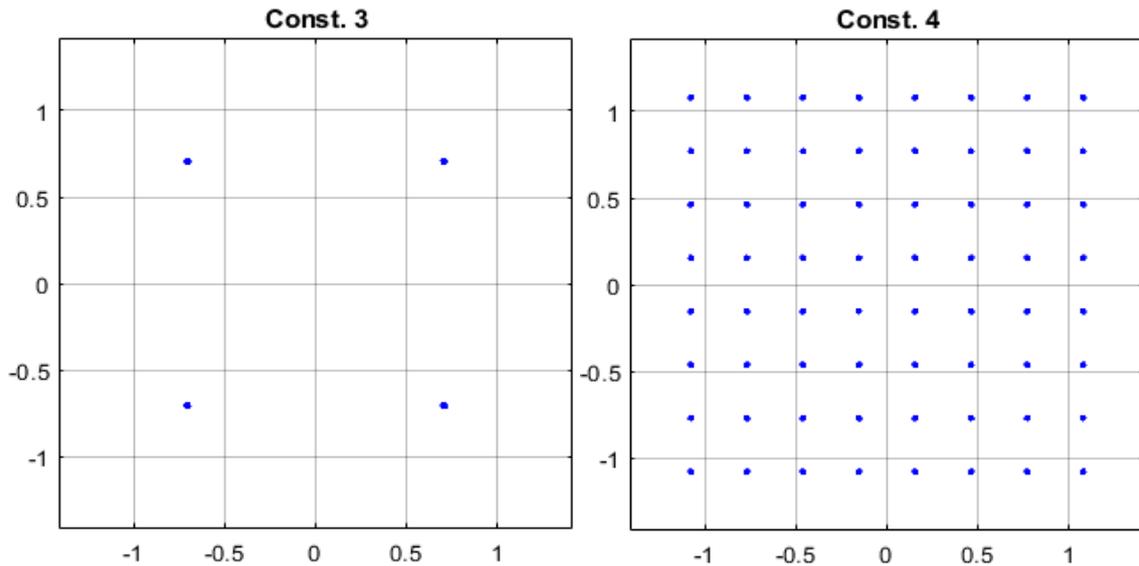


Figura 103. Constelaciones 3 y 4 de la práctica 5

2) A partir del espectro de una modulación 8-PSK con desfase $\pi/8$ obtenga el régimen simbólico de la señal.

A partir de la figura se puede obtener que el régimen simbólico es de 200 kHz.

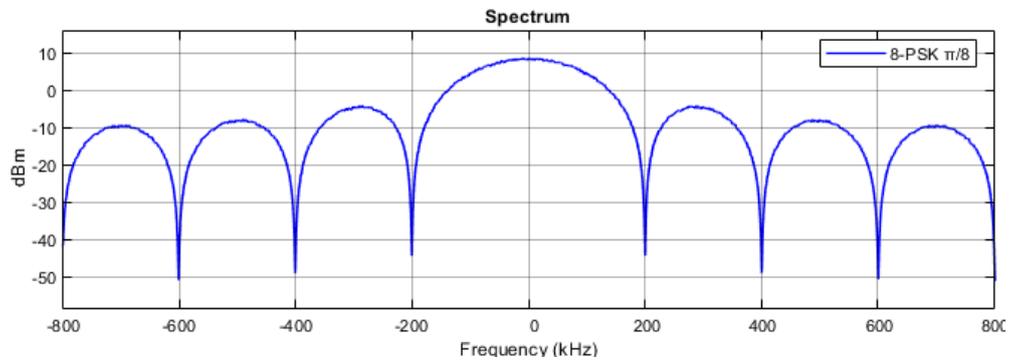


Figura 104. Espectro paso bajo de 8-PSK para la práctica 5

3) Obtenga el diagrama de ojo para una modulación 8-PSK. Utilice un filtro coseno alzado poder apreciar mejor los ojos.

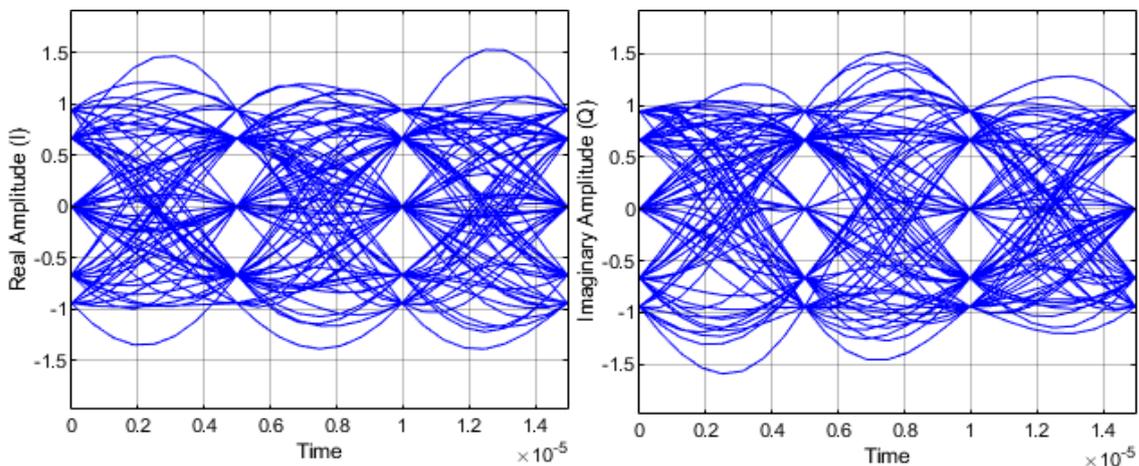


Figura 105. Diagramas de ojo de 8-PSK con coseno alzado $\alpha=0.2$

4) A partir de las siguientes figuras obtenga el factor de *roll-off* del filtro coseno alzado sabiendo que el régimen simbólico es de 200 kHz.

$$B = \frac{R_S}{2}(1 + \alpha) \rightarrow \alpha = \frac{2B}{R_S} - 1$$

Para el espectro 1 tenemos un ancho de banda de 240 kHz, por tanto, aplicando la fórmula del filtro coseno alzado obtenemos un $\alpha = 0.2$. Para el espectro 2 tenemos un ancho de banda de 320 kHz, luego $\alpha = 0.6$.

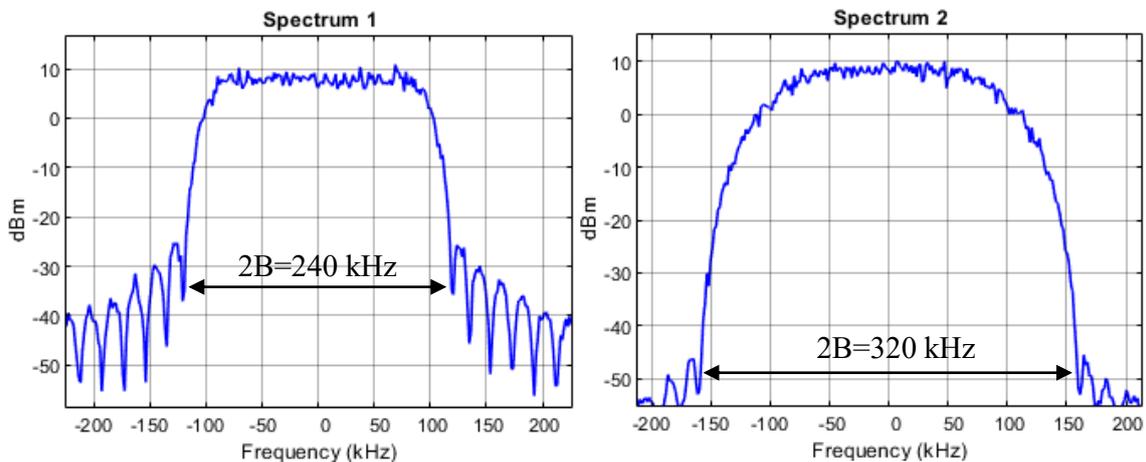


Figura 106. Espectros paso filtrados en coseno alzado para la práctica 5

5) Generar una constelación QPSK y una MSK en la que se muestre la trayectoria que sigue la señal. Explicar las similitudes y diferencias entre ambas constelaciones. Para ver las trayectorias utilizar el botón *Show Signal Trajectory* del visor de constelación.

Las modulaciones son para la constelación 1 una QPSK y para la constelación 2 una MSK. La principal diferencia entre las constelaciones es que la trayectoria de QPSK se pasa por cero mientras que MSK no. Ambos casos tienen 4 símbolos diferentes, es decir, 2 bits por símbolo.

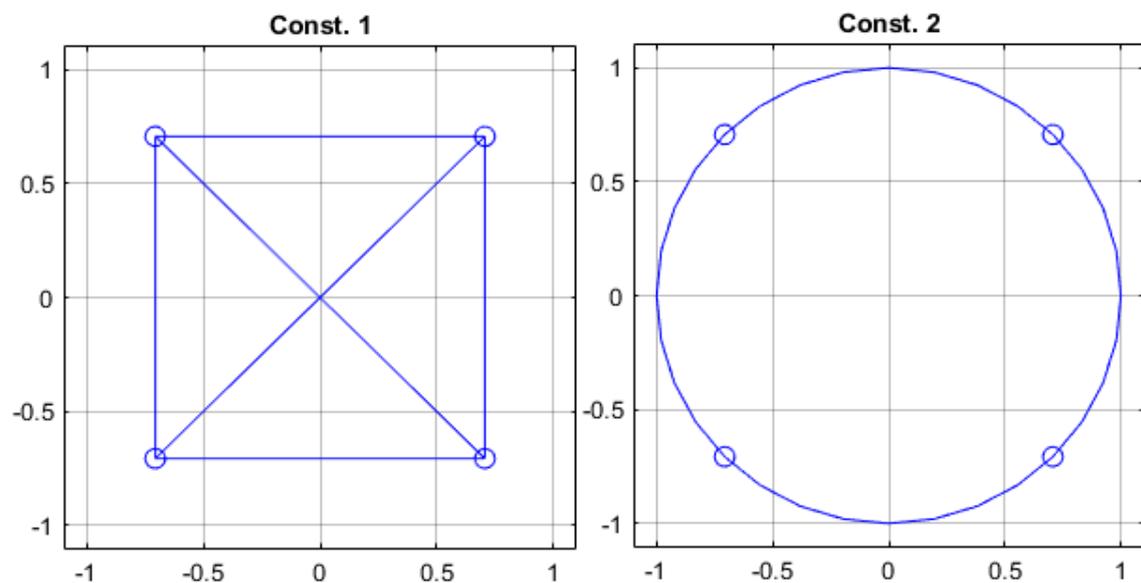


Figura 107. Comparación entre QPSK y MSK

6) Generar una 8-PSK o 64-QAM y aplicar diferentes efectos de canal, observando cómo afecta a la tasa de error de bit (BER) y a la constelación.

En primer lugar, añade ruido blanco gaussiano (AWGN) y anote los valores de la BER para distintos valores de E_B/N_0 . Tras esto, genere las curvas de BER utilizando comandos de MATLAB. Compare la BER obtenida con la BER teórica.

Tabla 20. Tasa de Error de Bit para 8-PSK y 64-QAM

E_B/N_0	8-PSK	64-QAM
10	$1.06 \cdot 10^{-3}$	$2.71 \cdot 10^{-2}$
11	$3.00 \cdot 10^{-4}$	$1.74 \cdot 10^{-2}$
12	$6.67 \cdot 10^{-5}$	$1.00 \cdot 10^{-2}$
13	$9.62 \cdot 10^{-6}$	$5.09 \cdot 10^{-3}$
14	$6.63 \cdot 10^{-7}$	$2.25 \cdot 10^{-3}$
15	$8.32 \cdot 10^{-8}$	$8.15 \cdot 10^{-4}$
16	0	$2.25 \cdot 10^{-4}$
17	0	$4.79 \cdot 10^{-5}$
18	0	$6.14 \cdot 10^{-6}$
19	0	$6.63 \cdot 10^{-7}$

Una vez obtenidos los diferentes valores de BER, se ha generado mediante la función de MATLAB `berawgn` las curvas teóricas para cada modulación. Se muestran todas sobre la misma gráfica de la Figura 108 en la que se puede ver que prácticamente coinciden con el valor esperado. En este caso se ha realizado solo para AWGN se pueden hacer diferentes variantes de este ejercicio utilizando todos los efectos de canal deseados para sacar las gráficas y compararlas con las teóricas.

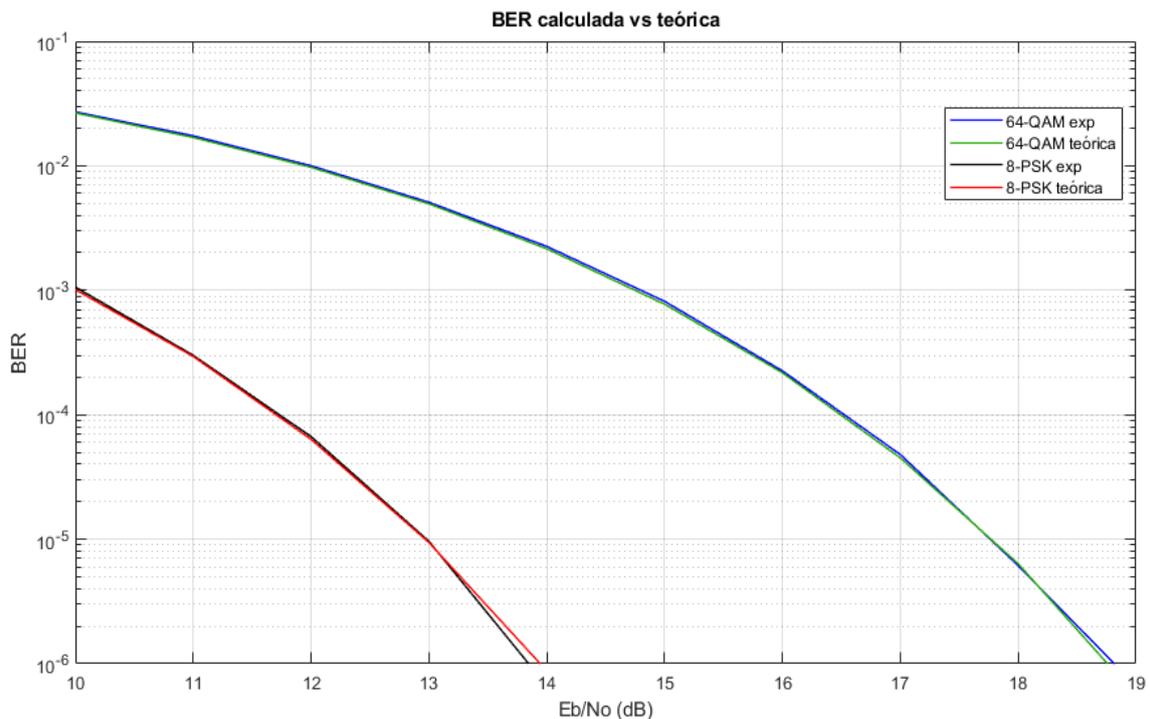


Figura 108. Gráficas de BER teóricas y experimentales

7) Utilizando efectos de canal se puede ver también cómo afectan a la constelación como, por ejemplo, la distorsión sobre una 16-QAM.

Para este ejemplo se ha utilizado AWGN con 25 dB de E_B/N_0 , atenuación de 2 dB, cambio de fase de 7° y distorsión no lineal con $k_2=0.05$ y $k_3=-0.05$.

La BER inicial, solo con AWGN, es de 0. La BER obtenida con los parámetros de distorsión ya mencionados es de $5.87 \cdot 10^{-4}$. Como se puede observar en la Figura 109 la constelación se ha deformado bastante provocando errores.

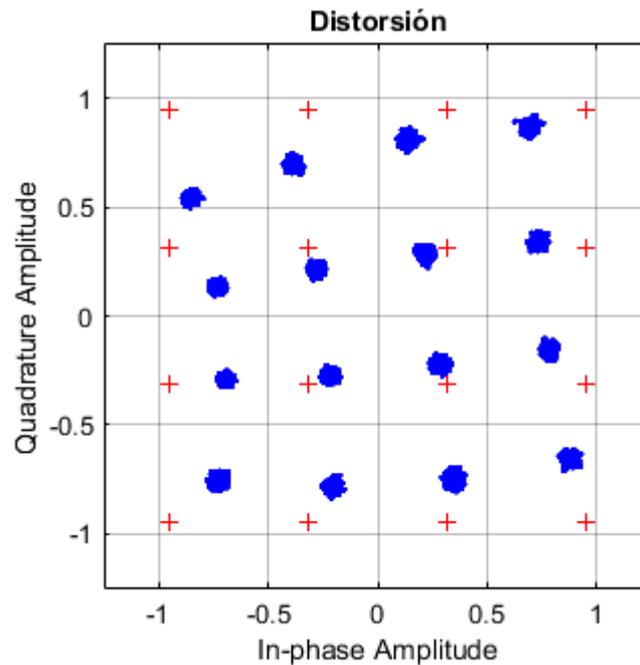


Figura 109. Efectos de la distorsión sobre una 16-QAM

8) El último ejercicio consiste en comparar diferentes anchos de banda de distintas modulaciones y el efecto de los códigos de canal en el ancho de banda. Para ello será necesario guardar las señales en el espacio de trabajo de MATLAB.

En este caso, se compara MSK con QPSK y con 4-FSK. Se observa que, para el mismo régimen simbólico y mismo número de bits por símbolo, MSK y 4-FSK ocupan algo más de ancho de banda que QPSK.

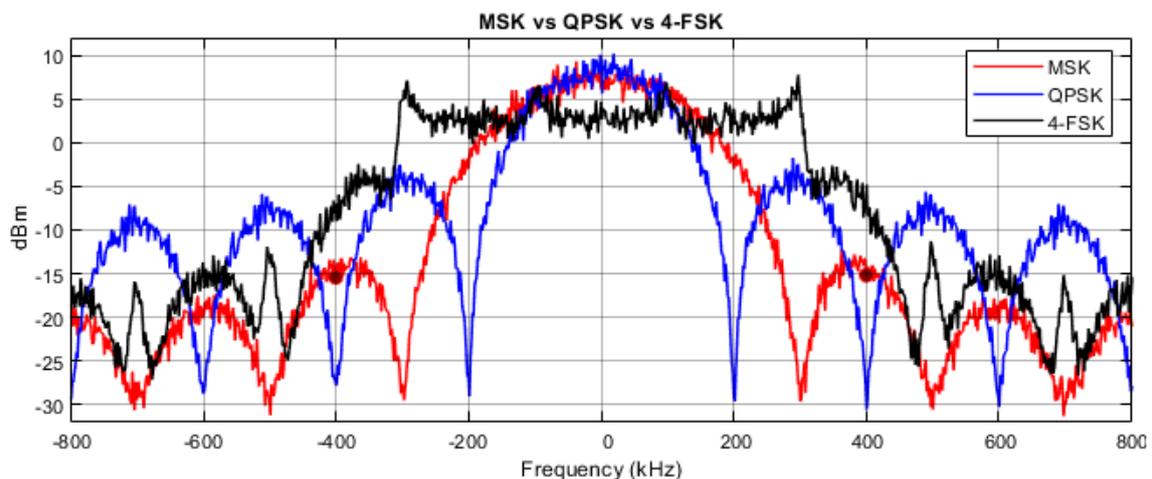


Figura 110. Comparación del ancho de banda de MSK, QPSK y 4-FSK en paso bajo

5.6 Practica 6. Introducción a DVB-T y Bluetooth

Esta práctica sirve como introducción para conocer, a grandes rasgos, dos sistemas comerciales muy populares como son DVB-T y Bluetooth, así como las modulaciones y técnicas de corrección de errores que se utilizan en cada uno.

5.6.1 OFDM

En primer lugar, se realiza una pequeña introducción para que los alumnos conozcan los principales parámetros y características de la modulación OFDM. Los parámetros y características de OFDM están explicados en el apartado 4.6.10 de este documento. Para este ejemplo se han utilizado los valores de la Tabla 21 y el diagrama de bloques que se muestra en la Figura 111. Todo se realizará utilizando la envolvente compleja banda base.

Tabla 21. Parámetros de OFDM

Bloque	Parámetros	Valores en el ejemplo
Sistema	Frecuencia de muestreo, R_{Sym} , en kbauds.	20 kbauds
	Fuente de información	Bit Aleatorio Uniforme
Modulación	64-QAM	10000 símbolos
OFDM	Longitud de la FFT, número de portadoras	64
	Bandas de guarda	6 por izq., 5 por dcha.
	Número de símbolos OFDM	5000
	Nulo en DC	Sí
	Posiciones de los pilotos	15, 27, 39, 51

El espectro obtenido con el diagrama de bloques es más sencillo que el de DVB-T ya que solo se utiliza una FFT de longitud 64. La ventaja de utilizar este montaje es que se obtiene un espectro de OFDM fácil de analizar por los alumnos porque tiene menos portadoras, lo que permite identificar mejor los parámetros utilizados.

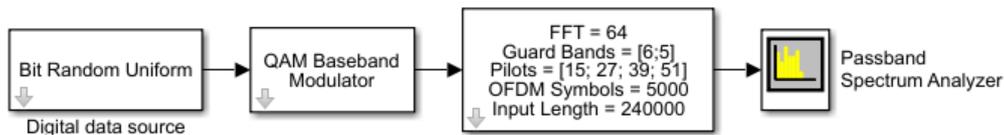


Figura 111. Diagrama de bloques para obtener el espectro de OFDM

1) Obtenga el espectro de OFDM. Identifique las posiciones de las bandas de guarda, los pilotos y el nulo en DC. Calcule el número de portadoras que llevan información.

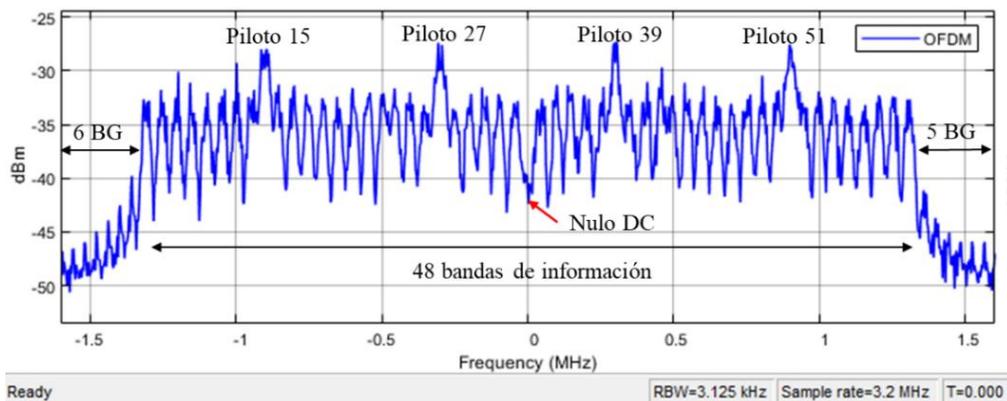


Figura 112. Espectro en banda base de OFDM con 4 pilotos y nulo en DC.

5.6.2 DVB-T

El estándar DVB-T [42] se corresponde con *Digital Video Broadcasting – Terrestrial*, más conocido en España como Televisión Digital Terrestre (TDT). Este estándar utiliza una modulación 64-QAM, multiplexación OFDM, entrelazado y códigos Reed-Solomon RS (204,188) entre otras características que se muestran en el diagrama de bloques del sistema, Figura 113.

Para la implementación de la práctica se ha modificado el ejemplo [43] que ofrece el paquete de herramientas *Communication Toolbox*. Este ejemplo implementa la interfaz radio de DVB-T en modo 2k. Se puede acceder al ejemplo original mediante este comando `commdvbt`. A este ejemplo se le han incluido los efectos de canal ya explicados en el apartado 4.8.

El proceso de codificación, modulación y transmisión es el siguiente: primero se aplica una codificación exterior a nivel de byte de tipo Reed Solomon (204, 188). A cada paquete se le añaden 16 bytes de redundancia cíclica. La capacidad de corrección es de hasta 8 bytes. En segundo lugar, se aplica un entrelazado exterior, a nivel de bytes, que sirve para evitar ráfagas de errores. Tras esto, se aplica un código convolucional a nivel de bit con *puncturing* con una tasa r_{cc} de 3/4. En cuarto lugar, se aplica un entrelazado interior a nivel de bit y a nivel de símbolo y después se modula en 64-QAM. Por último, se generan 1512 portadoras OFDM, tal y como indica el estándar, y se transmite la señal.

Para el proceso de recepción, demodulación y decodificación se siguen los mismos pasos que en transmisión, pero en orden inverso. Se obtiene la tasa de error de bit antes y después del codificador Reed Solomon.

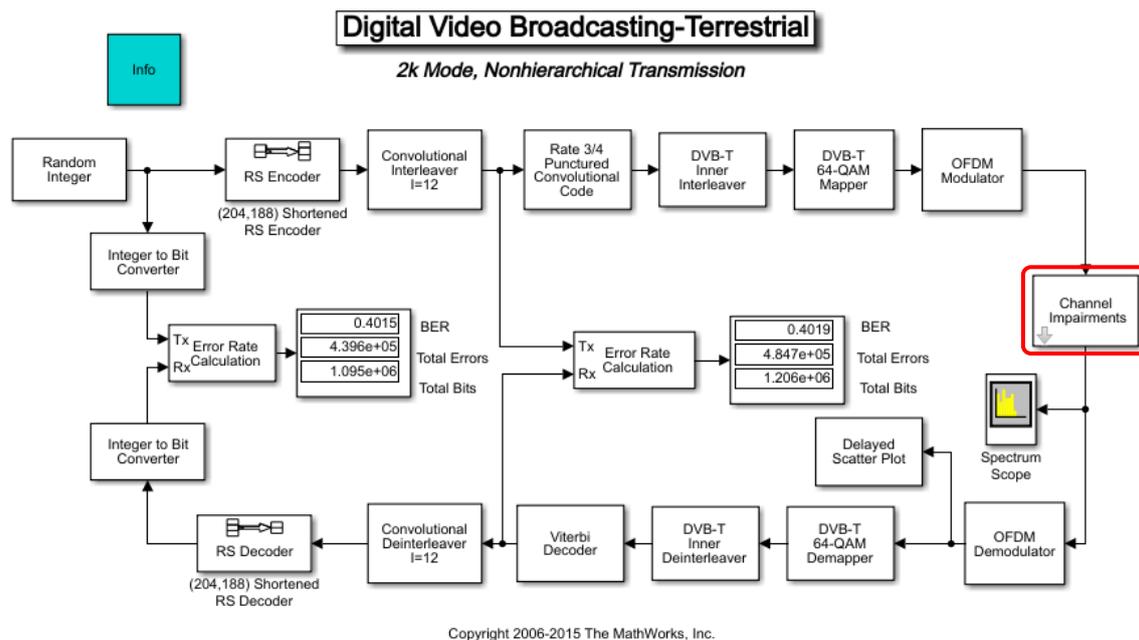


Figura 113. Diagrama de bloques de DVB-T modo 2K

El objetivo de esta práctica es obtener las curvas de BER para diferentes escenarios, es decir, para diferentes efectos de canal y valores de SNR. Estos valores también sirven para entender la necesidad de utilizar técnicas de corrección de errores.

1) Obtenga las curvas de tasa de error de bit (BER) para AWGN y para multitrayecto con dos rayos, con la configuración de la Figura 114. También obtenga las curvas que se obtienen a la salida de los decodificadores.

Ray 1		
<input checked="" type="checkbox"/>	Level (dB) 20	Delay (s) 10e-6
		Phase (deg) 8
Ray 2		
<input checked="" type="checkbox"/>	Level (dB) 20	Delay (s) 50e-6
		Phase (deg) -10

Figura 114. Parámetros de los rayos interferentes

Tabla 22. BER calculada para DVB-T con AWGN y multitrayecto

E_B/N_0	AWGN	AWGN+RS	C/N	Multitrayecto	Multitrayecto+RS
15	0.057834	$5.834 \cdot 10^{-2}$	15	0.3218	0.3204
16	0.001667	$1.103 \cdot 10^{-2}$	16	0.1878	0.1871
17	0	$1.439 \cdot 10^{-3}$	17	$6.545 \cdot 10^{-2}$	$6.543 \cdot 10^{-2}$
18	0	$1.302 \cdot 10^{-4}$	18	$3.700 \cdot 10^{-3}$	$1.279 \cdot 10^{-2}$
19	0	$1.078 \cdot 10^{-5}$	19	$5.092 \cdot 10^{-5}$	$1.545 \cdot 10^{-3}$
20	0	0	20	0	$1.166 \cdot 10^{-4}$
21	0	0	21	0	$5.092 \cdot 10^{-5}$
22	0	0	22	0	0

Los resultados obtenidos se muestran en la Figura 115. En esta gráfica se observa que la utilización de Reed Solomon mejora enormemente la BER en ambos casos.

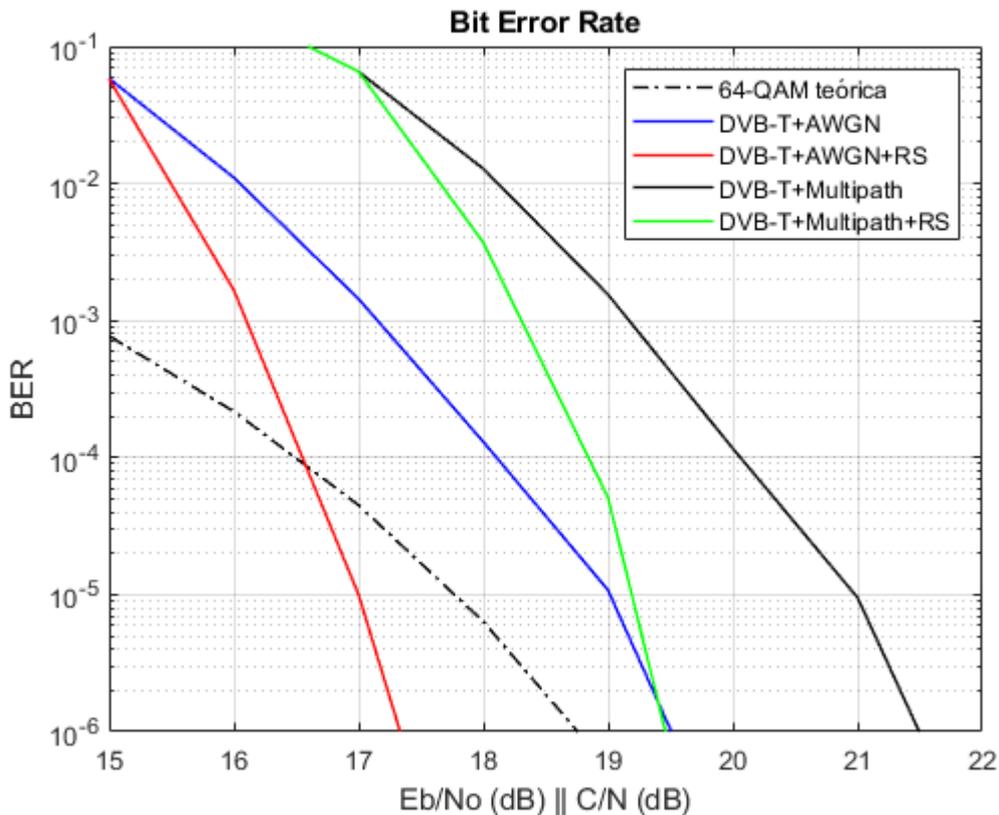


Figura 115. Curvas de BER para DVB-T con AWGN y multitrayecto

2) Obtenga el espectro de OFDM, así como la constelación de 64-QAM. Explique las imágenes. En este ejemplo se utiliza una SNR de 25 dB.

En la Figura 116 se puede apreciar el espectro de OFDM. Tiene bandas de guarda a los laterales y muchas portadoras que no se aprecian a simple vista. Según el estándar, se tienen 1512 portadoras de información y las bandas de guarda son de 268 portadoras a cada lado del espectro.

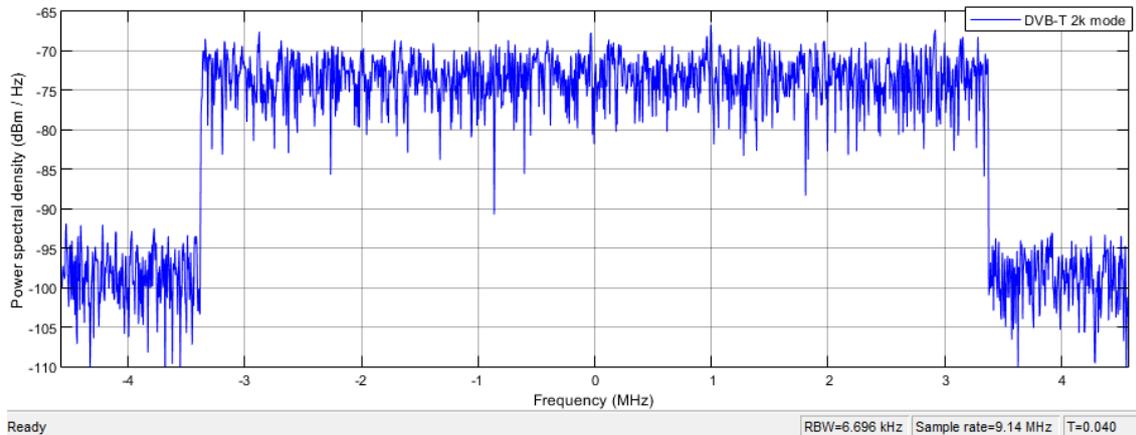


Figura 116. Espectro banda base de DVB-T en modo 2k

En la Figura 117 se muestra la constelación de la señal recibida. Se observa que es una modulación 64-QAM.

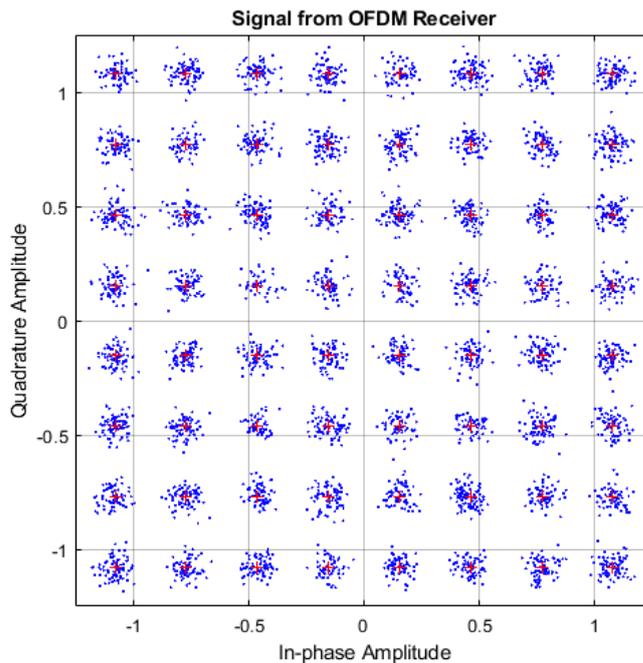


Figura 117. Constelación de 64-QAM utilizada en DVB-T

5.6.3 GFSK

En este apartado se realiza una introducción a la modulación GFSK. Esta modulación es muy parecida a la modulación FSK, de ahí su nombre. La principal diferencia es que GFSK utiliza un filtro los pulsos de datos utilizando un filtro paso bajo Gaussiano para obtener unas transiciones más suaves consiguiendo reducir el ancho de banda y la potencia de las bandas laterales. Se consigue reducir la interferencia con los canales vecinos, aunque se incrementa la interferencia intersimbólica (ISI). GFSK es una de las modulaciones utilizadas en el estándar de Bluetooth que se verá en el siguiente apartado.

Los parámetros de GFSK son la desviación de frecuencia respecto a la portadora y los parámetros típicos del filtro Gaussiano como son el producto BT, el ancho de banda a -3dB y el número de muestras por símbolo, ya explicados en el apartado 4.7.3.

En esta parte de la práctica se obtiene el espectro de GFSK con el objetivo de observar lo estrecha que es la señal en comparación con FSK.

En la Figura 118 se muestra el ancho de banda necesario para transmitir una señal con un régimen binario de 1Mbps. Se puede comprobar que es mucho mayor el ancho de banda que necesita FSK que el de GFSK.

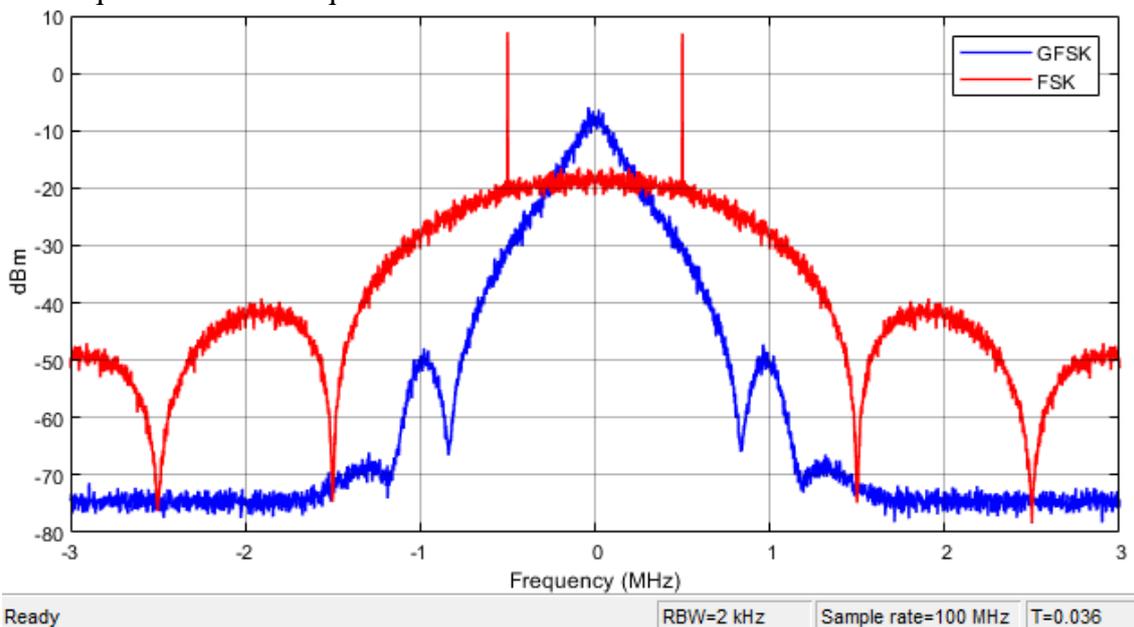


Figura 118. Anchos de banda para GFSK y FSK con una señal de 1Mbps

La principal ventaja de GFSK es la gran caída que produce el filtro Gaussiano, lo que facilita utilizar esta modulación con varios canales. En la Figura 119 se puede observar que las interferencias entre canales adyacentes son muy pequeñas en GFSK.

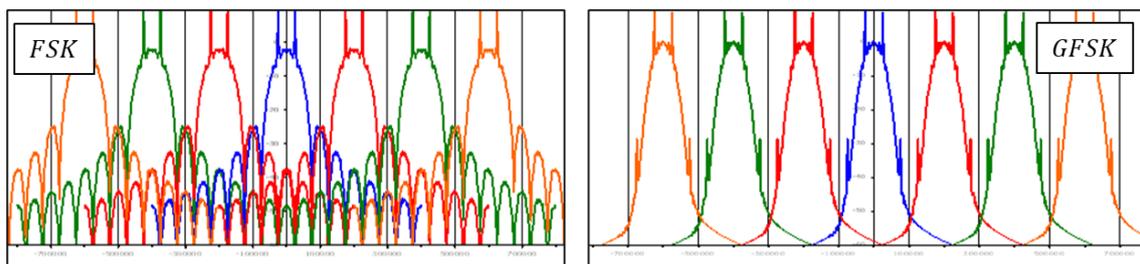


Figura 119. Canalización de FSK y GFSK

5.6.4 Bluetooth

En esta parte de la práctica se realiza un estudio de la interfaz radio del estándar Bluetooth. Esta tecnología nació en 1998 y fue estandarizada por el IEEE como 802.15.1. En la actualidad, la organización que mantiene el estándar es Bluetooth Special Interest Group (SIG) [44], formado por más de 36.000 compañías tecnológicas con el objetivo de unificar, armonizar e impulsar la innovación alrededor de este estándar.

Bluetooth trabaja en la banda de frecuencias no licenciada entre 2.400 y 2.480 GHz. La interfaz radio utiliza una técnica conocida como saltos de frecuencia con espectro ensanchado o *Frequency Hopping Spread Spectrum* (FHSS) que consiste en cambiar rápidamente la frecuencia de la portadora para evitar interferencias.

Existen dos tipos de Bluetooth según la aplicación deseada: Bluetooth Classic y Bluetooth Low Energy (BLE) que se utiliza en aplicaciones en las que se busca consumir muy poca energía. En la Tabla 23, obtenida de [45], se muestran algunas características de ambos estándares. En esta práctica se utiliza Bluetooth Classic.

Tabla 23. Comparativa entre Bluetooth LE y Bluetooth Classic

	Bluetooth Low Energy	Bluetooth Classic
Banda de Frecuencias	2.402 – 2.480 GHz	2.402 – 2.480 GHz
Canales	40 canales (3 control, 37 datos)	79 canales
	Espaciado: 2 MHz entre canales	Espaciado: 1 MHz entre canales
Uso del canal	FHSS	FHSS
Modulación	GFSK	GFSK, $\pi/4$ DQPSK, 8DPSK
Velocidad	2M PHY: 2 Mb/s 1M PHY: 1 Mb/s Coded PHY (S=2): 500 Kb/s Coded PHY (S=8): 125 Kb/s	8DPSK: 3 Mb/s $\pi/4$ DQPSK: 2 Mb/s GFSK: 1 Mb/s
Topologías	Point-to-Point (including piconet) Broadcast Mesh	Point-to-Point (including piconet)

En la Figura 120, se muestra cómo están distribuidos los canales de Bluetooth LE y Bluetooth Classic. En BLE se muestran los 37 canales de datos en azul claro y los 3 de control en azul oscuro, separados 2 MHz. Bluetooth Classic, en naranja, tiene 79 canales separados 1 MHz, pero no dispone de canales específicos para control.

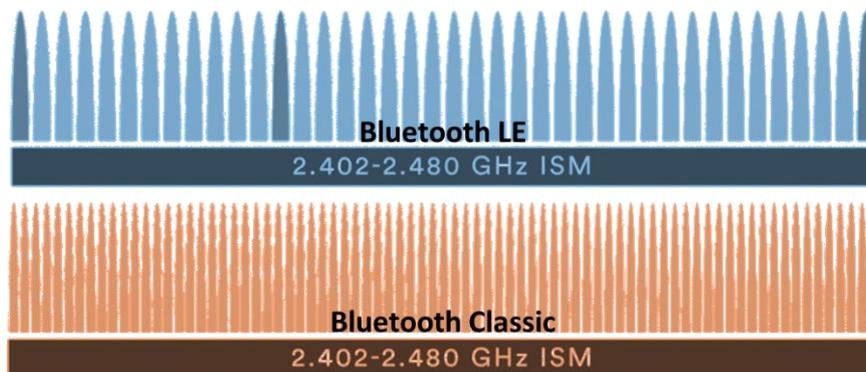


Figura 120. Canales de Bluetooth LE y Bluetooth Classic

Para la realización de esta práctica se parte del modelo de interfaz radio de Bluetooth desarrollado por V. Krishna [46], que pertenece al equipo de desarrollo de MathWorks. Este modelo se ha modificado para medir la BER antes y después del código de corrección de errores. La velocidad binaria es de 1 Mbps y se producen 1600 saltos de frecuencia por segundo. El diagrama de bloques utilizado en esta práctica es el de la Figura 121. Se utiliza un modulador GFSK con salto de frecuencia.

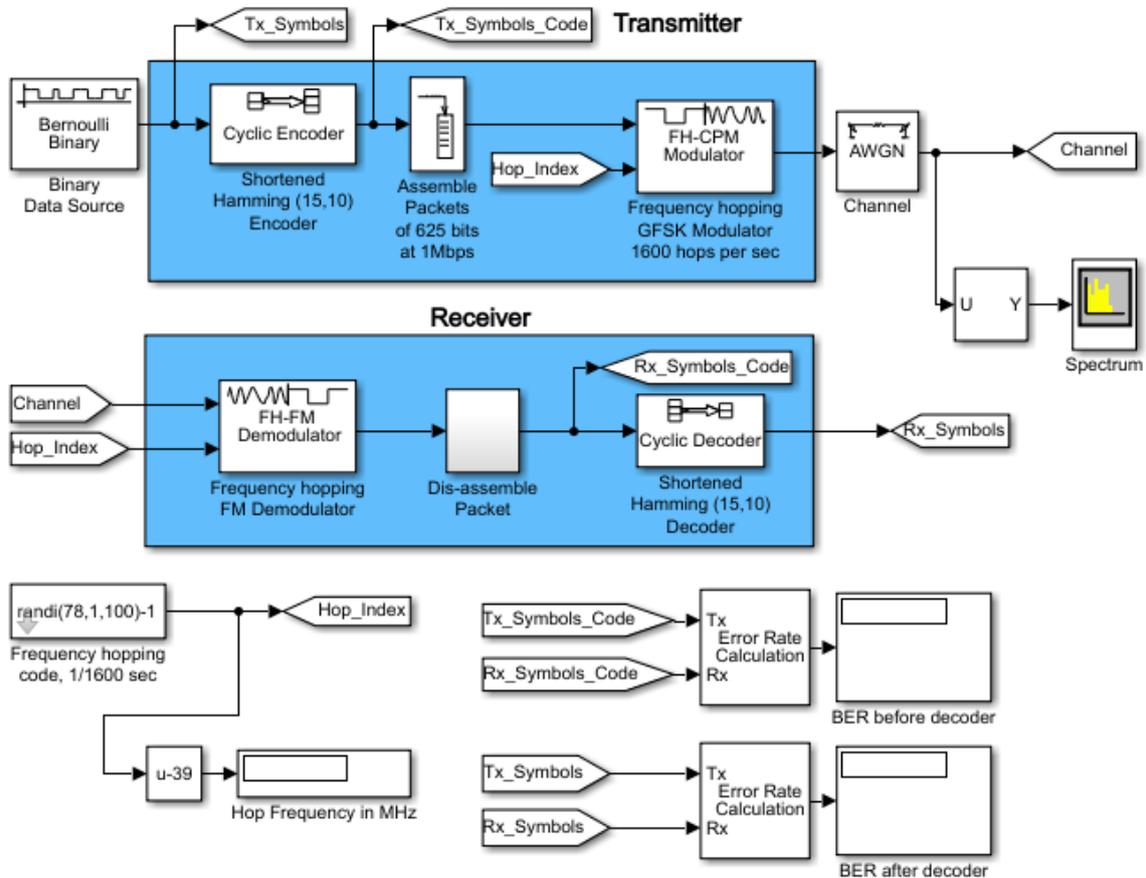


Figura 121. Diagrama de bloques de la interfaz radio de Bluetooth Classic

1) Capture el espectro en banda base de la señal transmitida en diferentes instantes de tiempo. Compruebe que la señal cambia de frecuencia rápidamente.

En las tres capturas se muestra que la señal está a frecuencias diferentes.

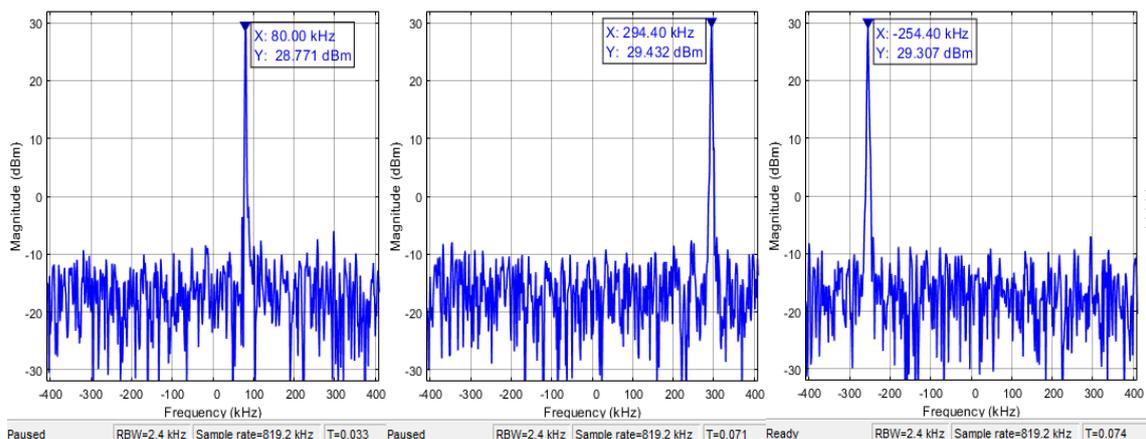


Figura 122. Espectro banda base de Bluetooth Classic en diferentes instantes

2) Obtenga las curvas de tasa de error de bit (BER) antes y después de los codificadores.

Tabla 24. Cálculos de BER para Bluetooth

E_B/N_0	AWGN antes de código Hamming	AWGN después de código Hamming
9	$5.679 \cdot 10^{-2}$	$4.345 \cdot 10^{-2}$
10	$3.754 \cdot 10^{-2}$	$2.118 \cdot 10^{-2}$
11	$2.251 \cdot 10^{-2}$	$8.698 \cdot 10^{-3}$
12	$1.206 \cdot 10^{-2}$	$2.598 \cdot 10^{-3}$
13	$5.677 \cdot 10^{-3}$	$5.794 \cdot 10^{-4}$
14	$2.229 \cdot 10^{-3}$	$7.055 \cdot 10^{-5}$
15	$6.924 \cdot 10^{-4}$	$7.505 \cdot 10^{-6}$
16	$1.771 \cdot 10^{-4}$	0
17	$2.502 \cdot 10^{-5}$	0
18	$1.001 \cdot 10^{-6}$	0

Las curvas obtenidas para GFSK con y sin código de corrección de errores se muestran en la Figura 123. Se observa el efecto del código de Hamming para corregir los errores producidos.

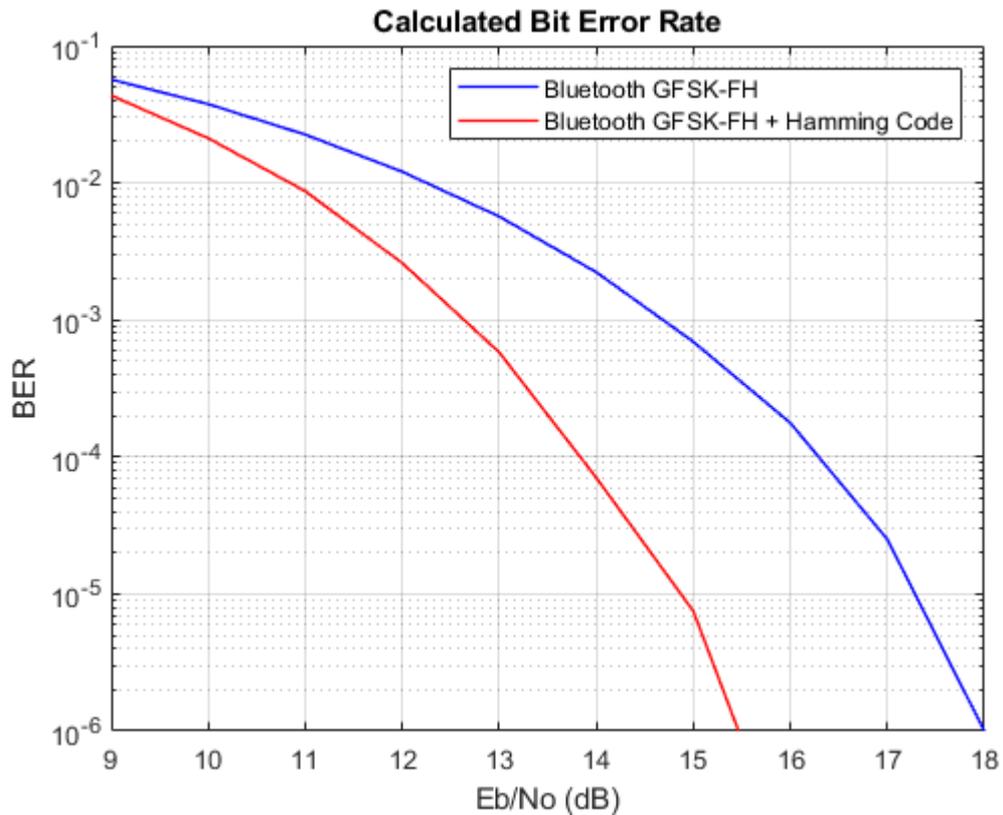


Figura 123. Curvas de BER para Bluetooth

6. Planos

Como este proyecto no consiste en un diseño hardware, se muestra el diagrama de bloques del sistema que se ha implementado en Simulink. La disposición de los bloques en el programa es en horizontal, pero para una mejor visualización se ha sacado la captura en vertical. Las funcionalidades de estos bloques ya han sido explicadas en los apartados anteriores. Los bloques que no aparecen en este diagrama se incluyen dentro del bloque More Blocks.

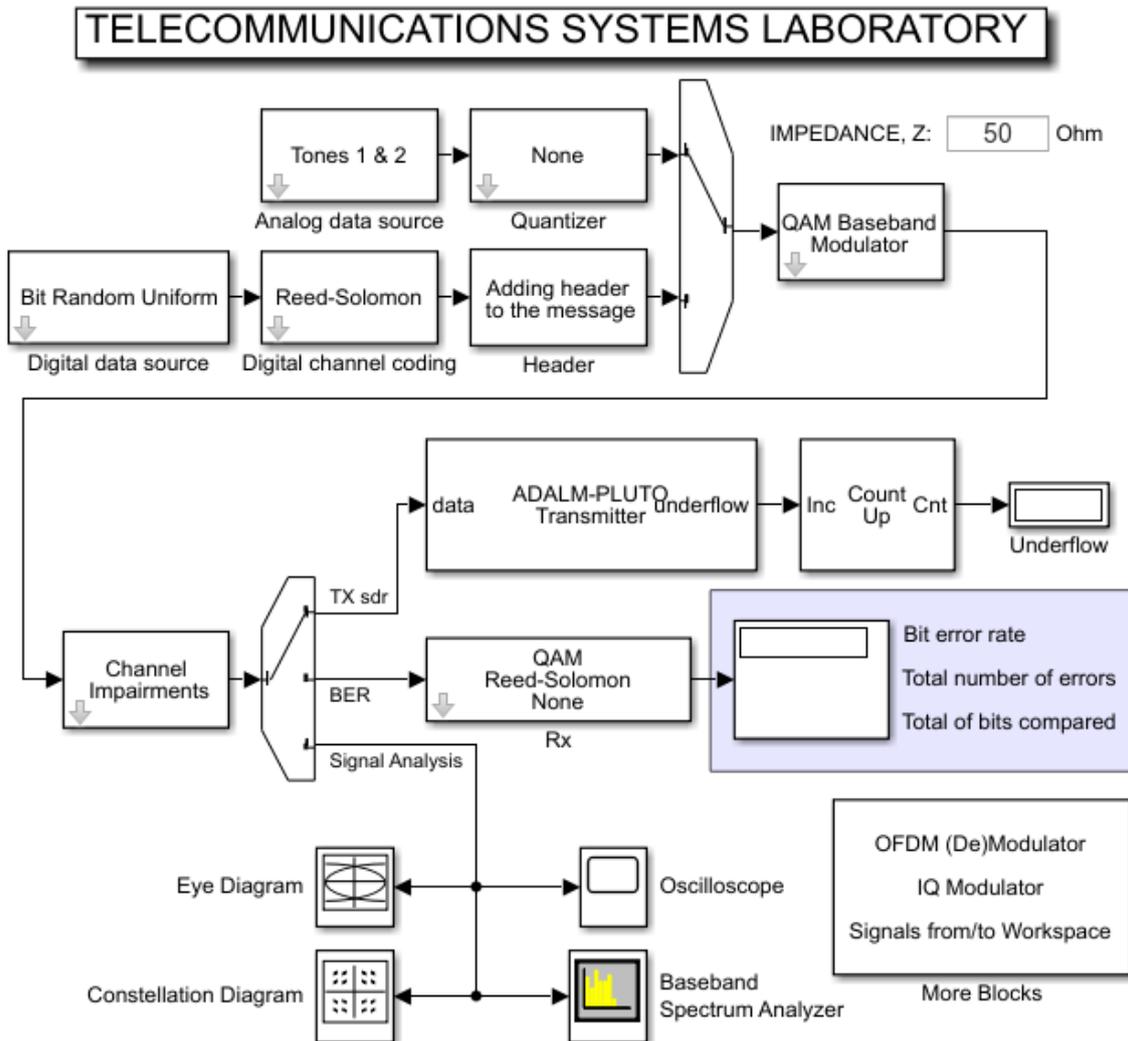


Figura 124. Diagrama de bloques del sistema

En las siguientes capturas se muestra el contenido de cada uno de los bloques que aparecen en la Figura 124.

En la Figura 125 se muestra el diagrama de bloques del bloque que genera las señales analógicas. Tenemos los cuatro generadores: tonos, aleatoria uniforme, gaussiana y desde un archivo analógico. Todos estos generadores tienen la salida normalizada.

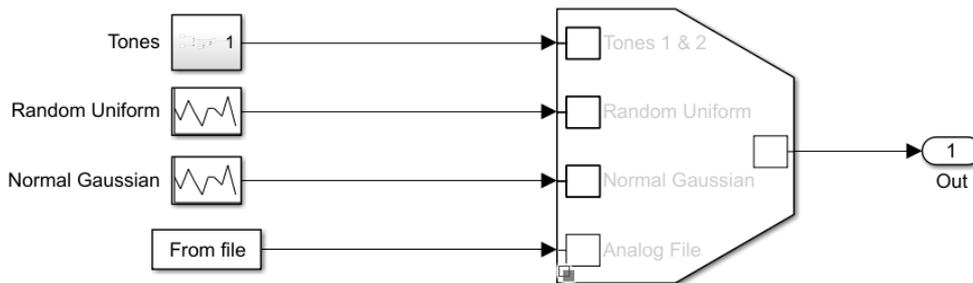


Figura 125. Diagrama de bloques del bloque generador de señales analógicas

En la Figura 126 se muestra el diagrama de bloques del generador de señales digitales. Tenemos seis bloques generadores: todo ceros, todo unos, patrón definido por el usuario, generador de PRBS, bit aleatorio uniforme y un archivo binario.

Todas estas señales generan unos y ceros a excepción del bloque del archivo binario. Este bloque envía los unos y ceros que lee del fichero y, cuando termina de leer el fichero, empieza a enviar -1. Por este motivo se han incluido los elementos del recuadro rojo que sirven para terminar la transmisión cuando se termina de leer el fichero.

El elemento del recuadro azul es un bloque que envía la señal binaria generada al bloque que calcula la BER. De esta manera se tiene un diagrama de bloques del sistema más limpio y ordenado.

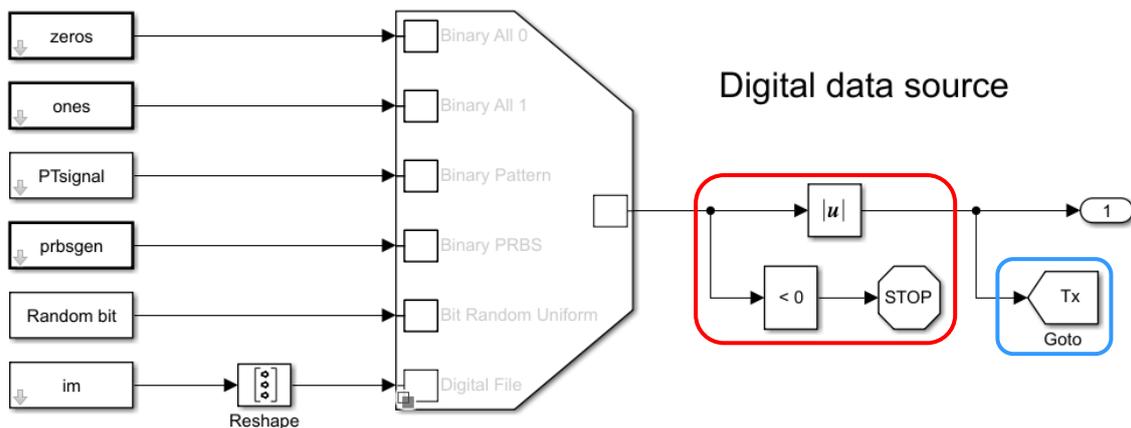


Figura 126. Diagrama de bloques del bloque generador de señales digitales

En la Figura 127 se muestra el diagrama de bloques del codificador de canal en el que se incluyen los codificadores BCH y Reed Solomon y el bloque de entrelazado.

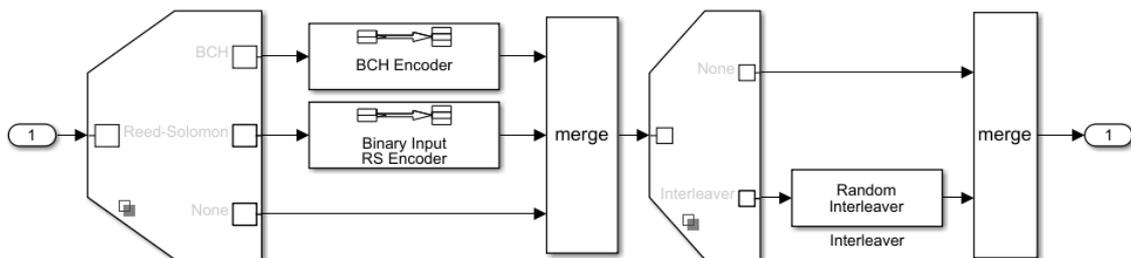


Figura 127. Diagrama de bloques del bloque codificador de canal

En la Figura 128 se muestra el diagrama de bloques del cuantificador en el que se incluyen los cuantificadores Uniformes Midrise y Midtreat, Ley A y Ley Mu.

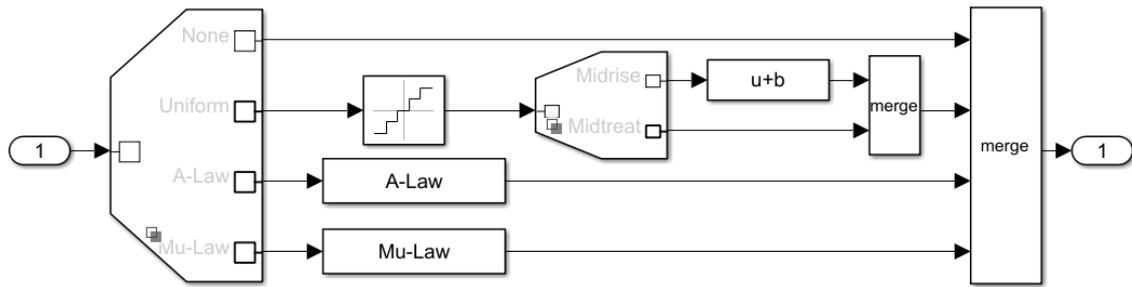


Figura 128. Diagrama de bloques del bloque cuantificador

En la Figura 129 se muestra el diagrama de bloques del bloque que añade la cabecera al mensaje.

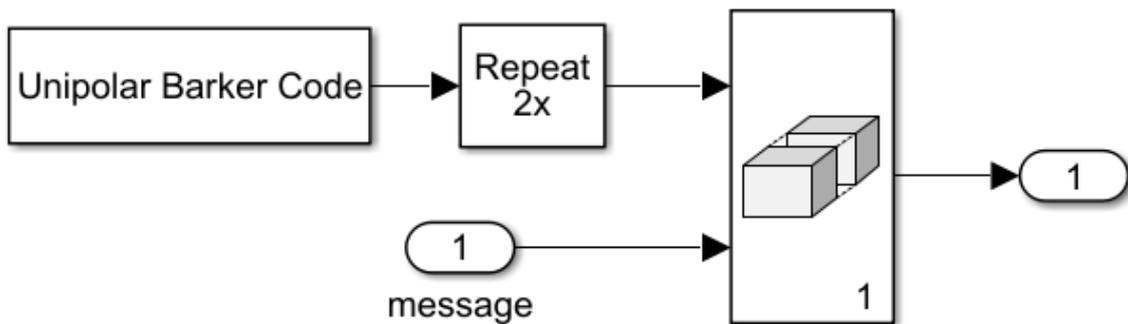


Figura 129. Diagrama de bloques del bloque que añade la cabecera

En la Figura 130 se muestra el diagrama de bloques del bloque modulador. Este bloque se divide en tres partes: la primera, en rojo, corresponde con el filtro de pre-énfasis. La segunda parte, en azul, es la parte que incluye todos los moduladores, tanto digitales como analógicos. Por último, en verde, está la parte de filtrado y sobremuestreo (*oversampling*) de la señal ya modulada.

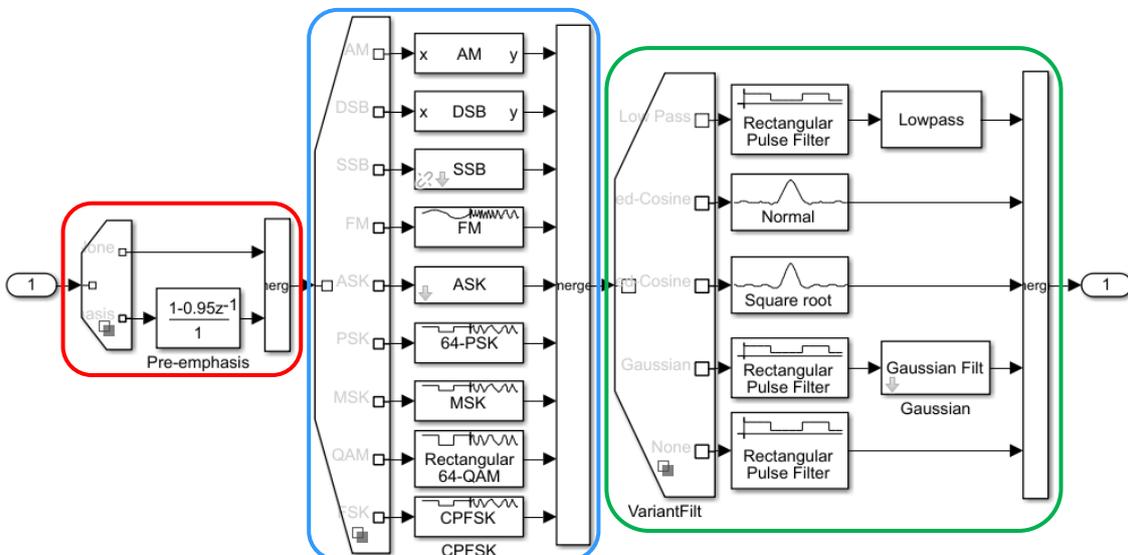


Figura 130. Diagrama de bloques del bloque modulador

En la Figura 131 se muestra el diagrama de bloques del bloque de efectos de canal. Los efectos se aplican en el orden que aparece en la imagen. Si el usuario quiere cambiarlos puede hacerlo cambiando el orden de los bloques.

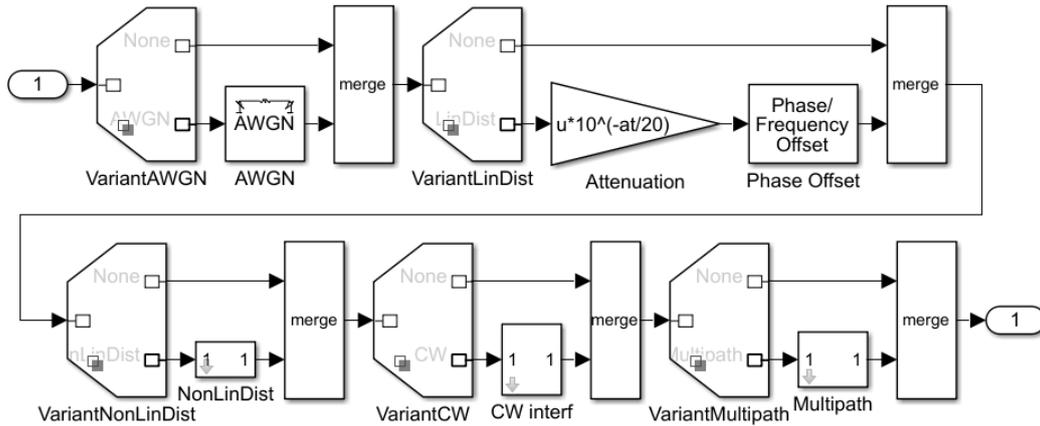


Figura 131. Diagrama de bloques del bloque de efectos de canal

En la Figura 132 se muestra el diagrama de bloques del modulador IQ. En el proyecto se incluyen dos moduladores IQ: uno para señales analógicas y otro para señales digitales. Ambos tienen el mismo diagrama de bloques. Es muy importante no equivocarse y utilizar el indicado para cada tipo de señal, si no se hace así no va a funcionar.

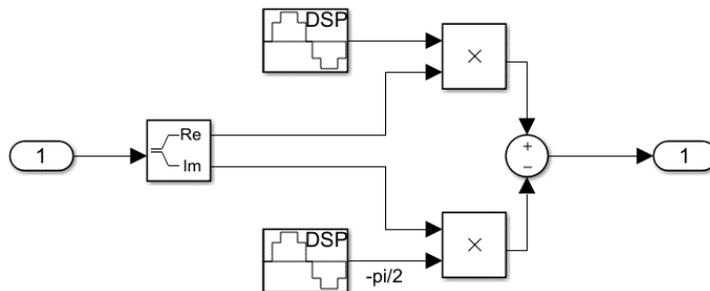


Figura 132. Diagrama de bloques del modulador IQ

En la Figura 133 se muestra el diagrama de bloques del receptor. En primer lugar, en rojo, se encuentra la etapa de filtrado. Después, en azul, está la etapa de detección de cabecera y de sincronización de trama. Tras la sincronización, en el bloque DemodDecod, en verde, se demodula y decodifica la señal y también se realiza el cálculo de BER. Por último, en amarillo, está el bloque para guardar la señal recibida ya demodulada y decodificada en el espacio de trabajo de MATLAB.

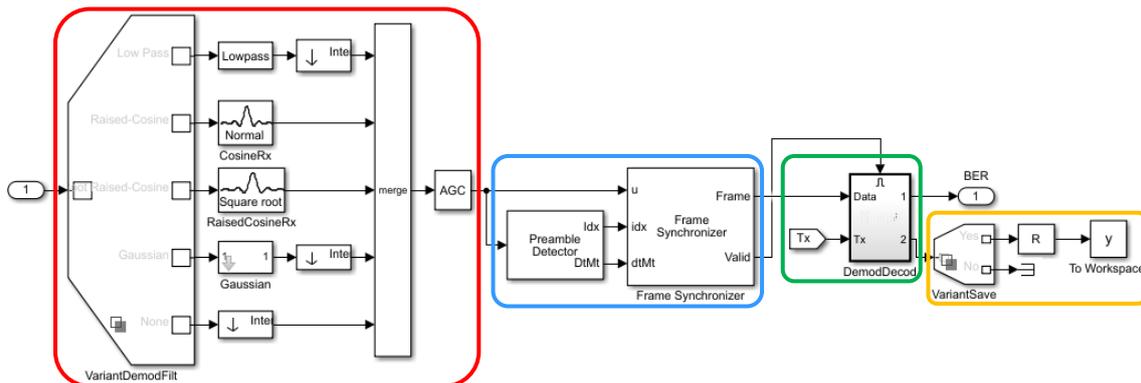


Figura 133. Diagrama de bloques del receptor

El bloque de demodulación de la Figura 133 está compuesto por los bloques de la Figura 134. En primer lugar, en rojo, se demodula la señal. Tras esto, en amarillo, se eliminan las cabeceras para poder decodificar correctamente. Después, en azul, se quita el entrelazado y de decodifica. Por último, en verde, se calcula la BER comparando la señal recibida con la señal transmitida. Las salidas de este bloque son la señal recibida y el cálculo de la BER.

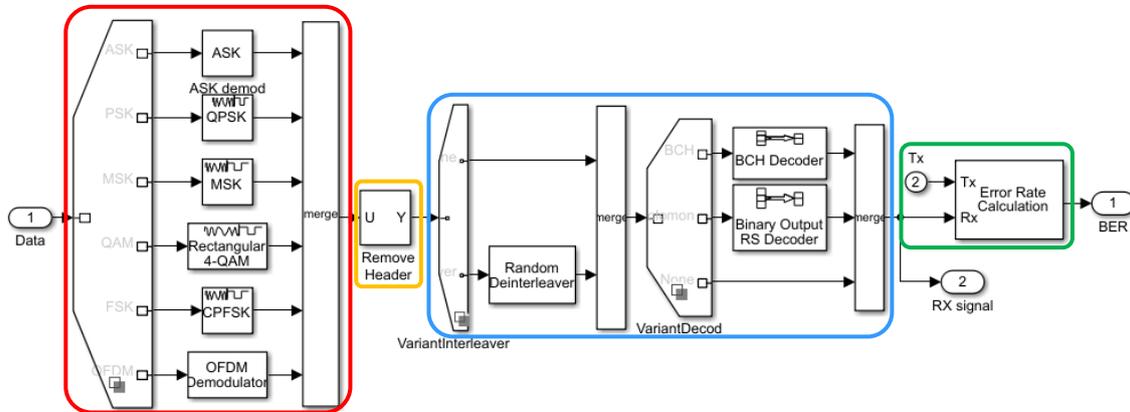


Figura 134. Diagrama de bloques del bloque DemodDecod

El bloque More Blocks sirve para guardar los bloques que no se están utilizando y tener un espacio de trabajo ordenado. El contenido de este bloque es el de la Figura 135: moduladores IQ para señales analógicas y digitales, un bloque que permite utilizar las señales del espacio de trabajo de MATLAB en Simulink y otro bloque que guarda las señales de Simulink en el espacio de trabajo de MATLAB. Por último, hay dos bloques de OFDM: uno corresponde al modulador y otro al demodulador.

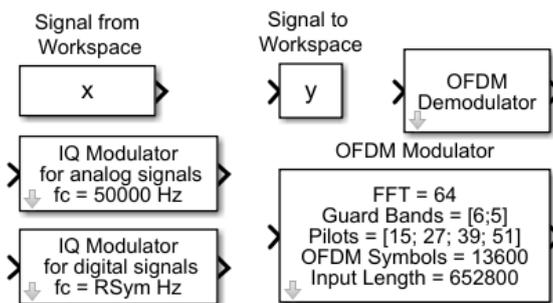


Figura 135. Bloques contenidos en el bloque More Blocks

----- Hoja en blanco -----

7. Presupuesto

En la Tabla 25 se muestra el presupuesto de los recursos humanos necesarios para desarrollar este proyecto.

Tabla 25. Presupuesto destinado a recursos humanos

Recursos humanos	Tiempo	Precio (€)
Mano de obra (15 €/ hora)	300	4.500
Total		4.500

En la Tabla 26 se muestra el coste material de todos los equipos utilizados durante el desarrollo del proyecto.

Tabla 26. Presupuesto destinado a equipamiento hardware

Equipamiento	Unidades	Precio (€)
Acer Aspire E1-572	1	500
Analog Devices Adalm-Pluto SDR	1	160
Monitor externo	1	80
Teclado	1	10
Ratón	1	10
Total		760

En la Tabla 27 se incluye el presupuesto de los recursos *software* utilizados en este proyecto fin de grado. En este caso, todos los recursos utilizados son gratuitos con la licencia universitaria.

Tabla 27. Presupuesto destinado a paquetes software

Paquetes Software	Unidades	Precio (€)
MATLAB 2020b	1	0
Simulink 10.2	1	0
Communications Toolbox 7.4	1	0
Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio 20.2.1	1	0
DSP System Toolbox 9.11	1	0
SerDes Toolbox 2.0	1	0
Total		0

En la Tabla 28 se muestra el presupuesto del material fungible utilizado para el diseño del proyecto.

Tabla 28. Presupuesto destinado a material fungible

Material fungible	Unidades	Precio/Ud.	Precio (€)
Cuaderno A5	2	3	6
Bolígrafo	5	0.20	1
Papel A4	50	0.02	1
Total			8

El presupuesto total del proyecto se muestra en la Tabla 29. En esta tabla se incluyen las contribuciones monetarias de las tablas anteriores.

Tabla 29. Presupuesto total del proyecto

Presupuesto total	Precio (€)
Recursos humanos	4.500
Equipamiento <i>hardware</i>	760
Paquetes <i>software</i>	0
Material fungible	8
Total	5.268

Además, en la Tabla 30 se incluye un presupuesto orientativo para diseñar desde cero un laboratorio en la Escuela con veinte ordenadores y veinte dispositivos SDR para los puestos de los alumnos y otro para el puesto del profesor.

Tabla 30. Presupuesto para la creación de un laboratorio

Equipamiento	Ud./Tiempo	Precio/Ud.	Precio (€)
Mesa oficina	21	150	3.150
Sillas alumnos	20	50	1.000
Silla profesor	1	100	100
Adalm-Pluto SDR	1	160	160
RTL-SDR	20	30	600
Ordenadores	21	550	11.550
Monitor	21	100	2.100
Pack Teclado y Ratón	21	15	315
Cables Ethernet	21	10	210
Montaje e instalación	80 horas	25 €/hora	2.000
Total			21.185

8. Manual de usuario

8.1 Instalación y configuración

Para un correcto funcionamiento del programa desarrollado es necesario instalar correctamente MATLAB, Simulink, las tres *toolbox* explicadas en el apartado 3.3.2 y el paquete de soporte del SDR siguiendo las instrucciones de los manuales propios que pueden obtenerse desde la página de descarga.

Una vez instalado todo, iniciar MATLAB, ir a la pestaña Home > Add-Ons > Manage Add-Ons. Conectar el ADALM-Pluto al ordenador mediante un cable USB incluido en la caja del dispositivo. Tras esto, se realizará una prueba de conexión entre MATLAB-Simulink y el SDR. Para iniciar la prueba hay que pulsar en la rueda de configuración del paquete de soporte de ADALM-Pluto como indica la Figura 136.



Figura 136. Paquete de soporte de ADALM-Pluto

Al pulsar sobre este botón se abre el asistente de conexión del dispositivo. Una vez seguidos los pasos indicados se habrá completado el proceso de instalación del SDR. Este proceso solo es necesario realizarlo durante la instalación, aunque también puede ser útil para comprobar que la conexión entre el ordenador y el SDR está bien configurada o para ver si el transmisor y receptor del SDR están funcionando correctamente. Si la instalación ha sido correcta saldrá una pantalla como la de la Figura 137. En caso de error se recomienda ver las posibles soluciones en la guía de usuario [33].

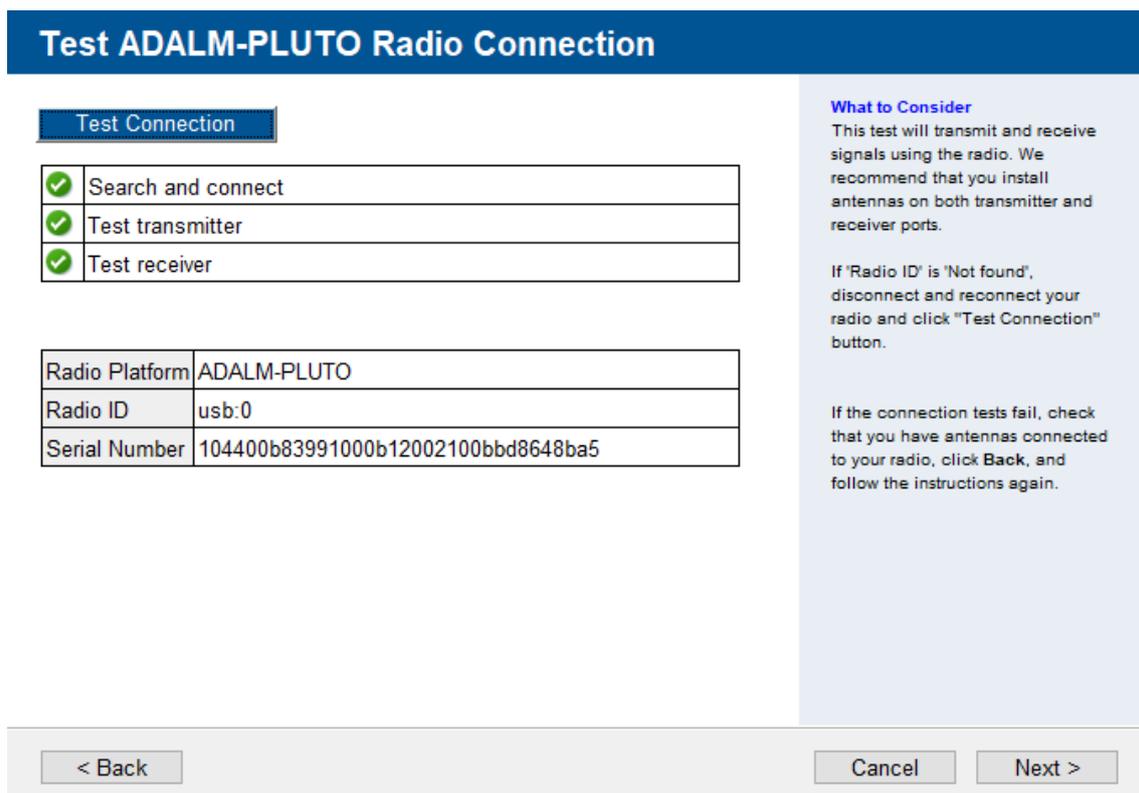


Figura 137. Instalación y test de conexión de ADALM-Pluto

8.2 Selección del tipo de señal y de funcionalidad

Para seleccionar una fuente analógica o digital simplemente hay que hacer doble click en el interruptor (Selector). El camino de arriba corresponde con la fuente analógica y el de abajo con la digital.

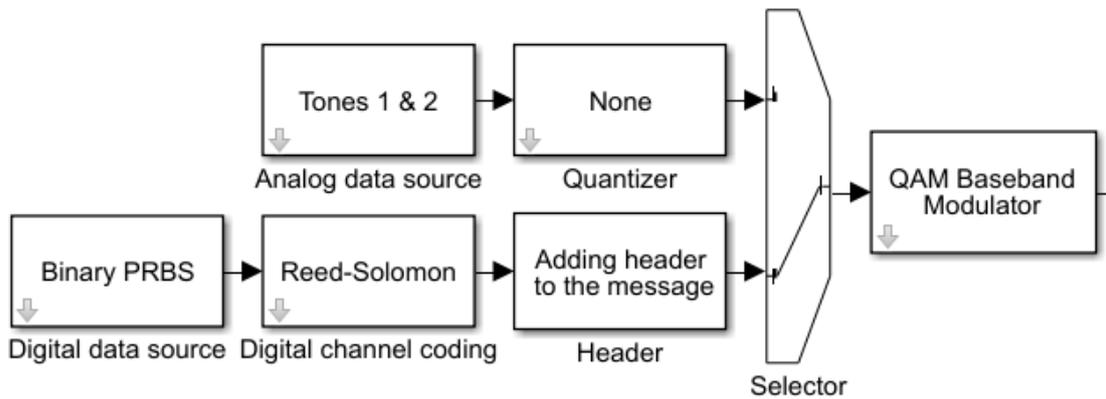


Figura 138. Selector de fuente analógica o digital

Para seleccionar la funcionalidad de salida del sistema hay que seguir el mismo funcionamiento que en el caso anterior. Para transmitir con el SDR hay que seleccionar el camino superior, para medir la BER seleccionar el camino central y para observar la señal hay que seleccionar el camino inferior.

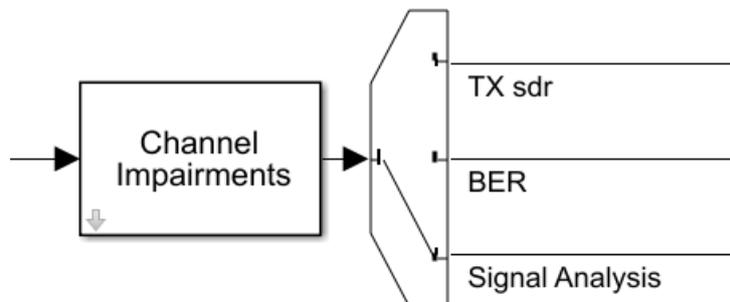


Figura 139. Selector de funcionalidad

Es necesario inhabilitar los bloques de las funcionalidades que no se están utilizando para aumentar la rapidez y que no haya errores en la comprobación de parámetros. Para quitar los bloques se deben comentar utilizando Ctrl + Shift + X o dando botón derecho y seleccionar la opción *Comment Out*.

	Cut	Ctrl+X
	Copy	Ctrl+C
	Paste	Ctrl+V
	Comment Through	Ctrl+Shift+Y
	Comment Out	Ctrl+Shift+X
	Uncomment	
	Delete	Del

Figura 140. Botón para inhabilitar los bloques no utilizados

8.3 Añadir nuevos bloques y funcionalidades

En este apartado se explica cómo añadir, a futuro, nuevas funcionalidades en el proyecto. A continuación, se pone un ejemplo sobre cómo añadir un nuevo codificador en el proyecto. Para ello, hay que seguir los siguientes pasos:

Primero hay que desarrollar el bloque de forma individual. Este bloque se puede crear de forma programática mediante System Objects o mediante los bloques incluidos en Simulink. En este caso un codificador de Hamming que ya está incluido en las librerías de Simulink.

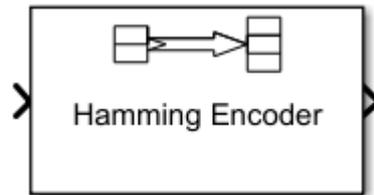


Figura 141. Bloque que incluye la nueva funcionalidad

Tras desarrollar el bloque, hay que incluirlo dentro del bloque que contiene a todos los bloques con esta funcionalidad. En este ejemplo, hay que incluirlo en el bloque de codificación de canal. Para ello hay que pulsar en la flecha inferior izquierda del bloque contenedor, de esta manera se entra dentro del contenedor y se puede incluir ahí el nuevo bloque.

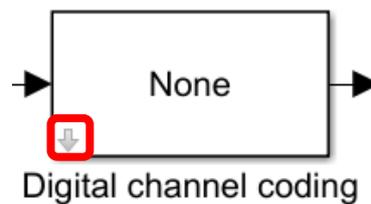


Figura 142. Bloque contenedor de los codificadores digitales

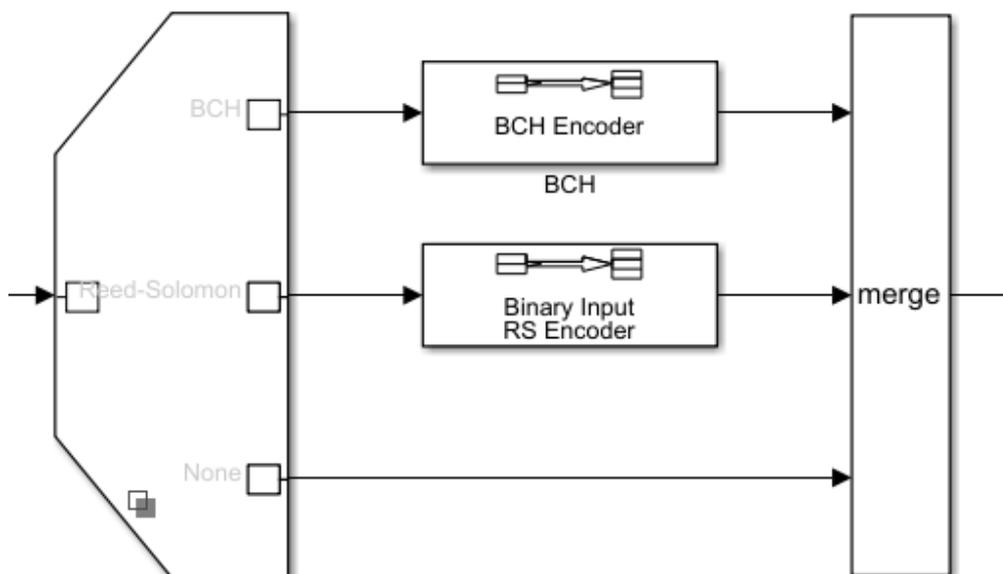


Figura 143. Interior del bloque antes de la modificación

Una vez dentro del bloque será necesario añadir una nueva salida al Variant Sink con el nombre de la funcionalidad que se está añadiendo, para ello, hacer doble click y dar al símbolo . El resultado debe ser el de la Figura 144.

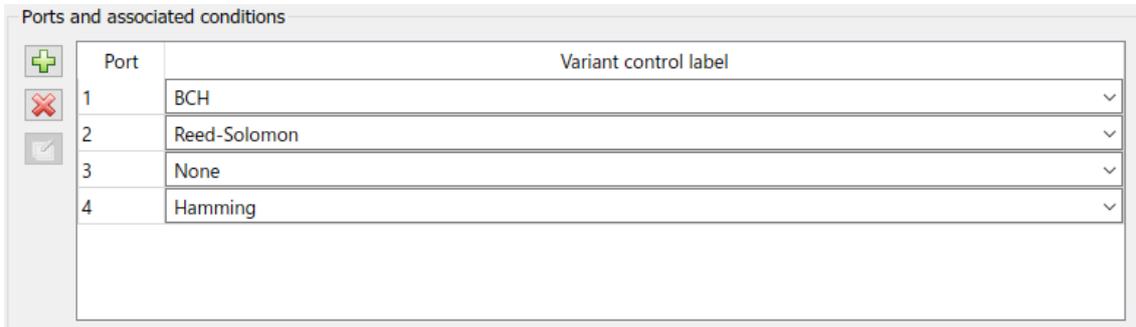


Figura 144. Añadir una nueva salida al Variant Sink

Una vez se tienen las 4 salidas, se añade el bloque del codificador Hamming consiguiendo el circuito de la Figura 145.

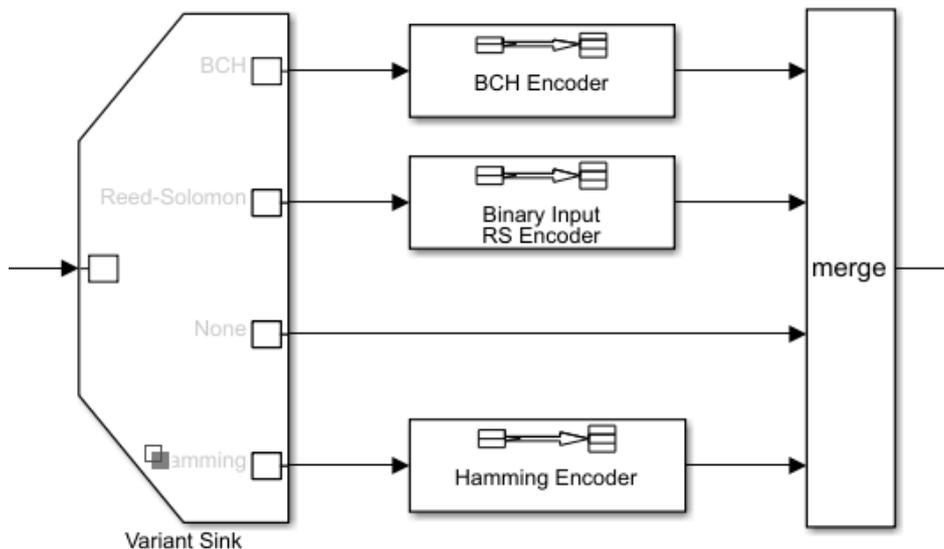


Figura 145. Circuito final después de la modificación

Una vez incluido el bloque, es necesario crear su máscara o interfaz de usuario, para ello hay que promocionar sus parámetros desde el editor de máscaras de Simulink. Para abrir el editor de máscaras: hacer click derecho sobre la flecha inferior izquierda del bloque contenedor y pulsar en Edit Mask o bien, utilizar el comando Ctrl+M.

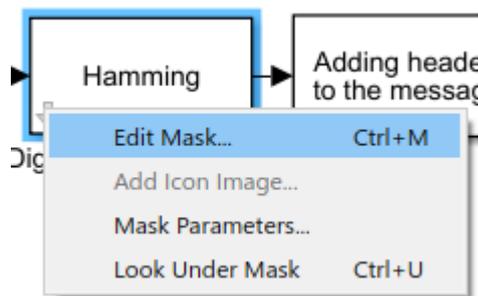


Figura 146. Abrir el editor de máscaras de Simulink

Se recomienda crear un nuevo Group Box para organizar mejor los parámetros y poder quitar la visibilidad de todo el panel cuando no se esté utilizando mediante un Callback. Para promocionar los parámetros hay que pulsar en Promote y seleccionar los parámetros deseados.

En la Figura 147 se muestra la máscara del bloque de codificación digital. En rojo se muestran los botones de Group Box y Promote y, en azul, el nuevo panel creado para el codificador de Hamming. En verde se muestra el parámetro desde el cual hay que cambiar el Callback para que esté todo conectado.

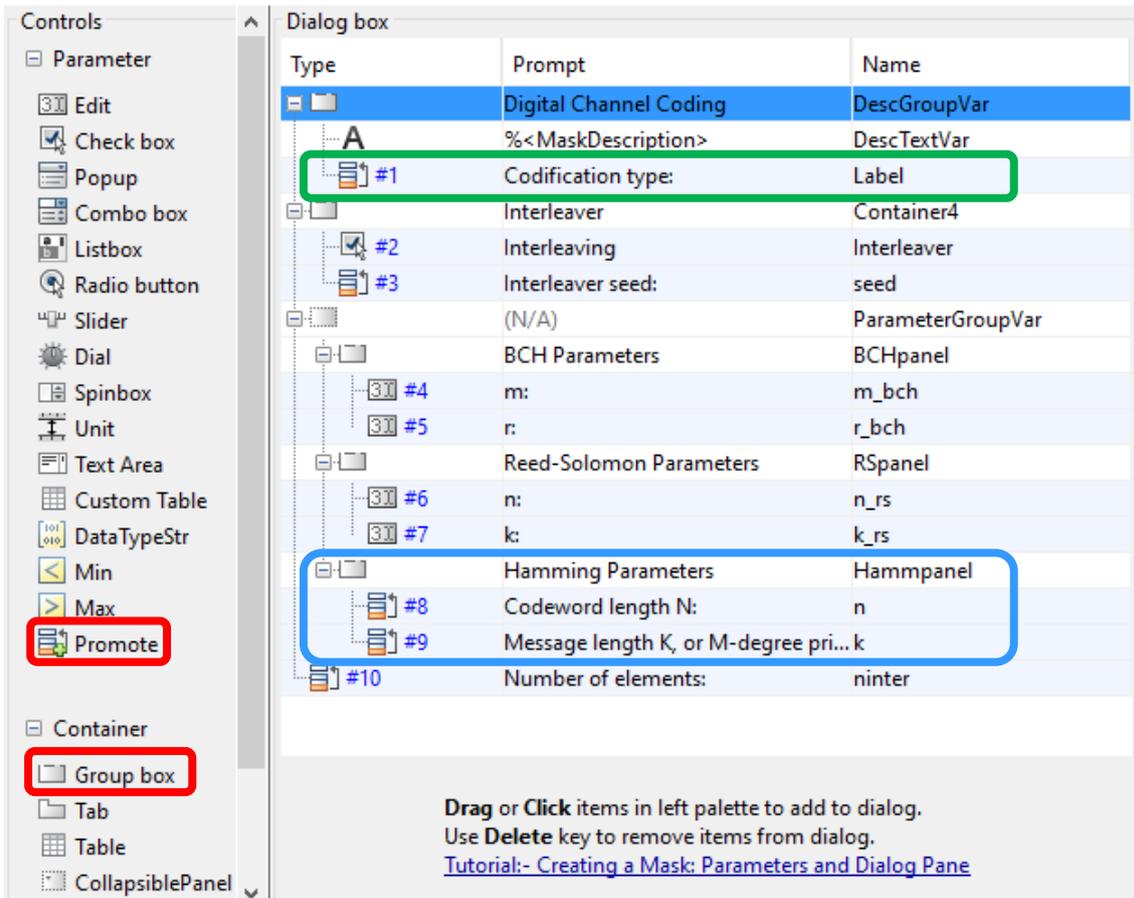


Figura 147. Editor de máscaras de Simulink

El Callback es el mismo que para los dos casos anteriores así que simplemente es copiar y pegar cambiando los nombres de las variables y añadiendo la conexión con el receptor, el cual también debe tener un decodificador Hamming. Para más información sobre las máscaras de Simulink y el funcionamiento de los Callback se recomienda leer la documentación oficial de MATLAB y Simulink que se encuentra en [31] y en [48].

8.4 Contenido del proyecto

El proyecto contiene diferentes ficheros entre los que se incluyen funciones de MATLAB, código de *System Objects* y los proyectos de las prácticas.

A continuación, se enumeran los ficheros necesarios para cada práctica:

- Práctica 1: se necesita el fichero LabTC_p1.slx
- Práctica 2: se necesita el fichero LabTC_p2.slx
- Práctica 3,4 y 5: se necesitan los siguientes ficheros: sdr_pfg.slx, amBlock.m, dsbBlock.m, filtdetector.m, FrameSynchronizer.m, offsetCalculator.m, osciloscopio.m, viewFMwaveform.m.
- Práctica 6 - DVB-T: se necesita el fichero commdvbt_modified.slx.
- Práctica 6 – Bluetooth: se necesita el fichero bluetooth_modified.slx

Se incluyen ficheros adicionales como bercalc.m, bit2jpg.m y jpg2bit.m que se proceden a explicar en el siguiente apartado.

El *script* bercalc.m sirve para calcular la BER teórica de una señal y también para introducir los valores de una BER obtenida a mano y representarla.

La función jpg2bit.m se utiliza para transformar una imagen en formato .jpg a un fichero binario, es decir, ceros y unos. Esta función se ha creado con el objetivo de poder utilizar imágenes como fuente de información “archivo binario” del bloque de señales digitales, ver apartado 4.3. Esta función devuelve la imagen de entrada convertida a bits y también devuelve las dimensiones de la imagen para poder reconstruirla.

La función bit2jpg.m se encarga de transformar los bits recibidos en un fichero .jpg para poder ver cómo se ha degradado la imagen debido a los efectos del canal. Para reconstruir la señal, necesita las dimensiones de la imagen.

8.5 Utilizar imágenes como fuente de datos

En este apartado se explica con más detalle los pasos que hay que seguir para utilizar una imagen como fuente de información digital.

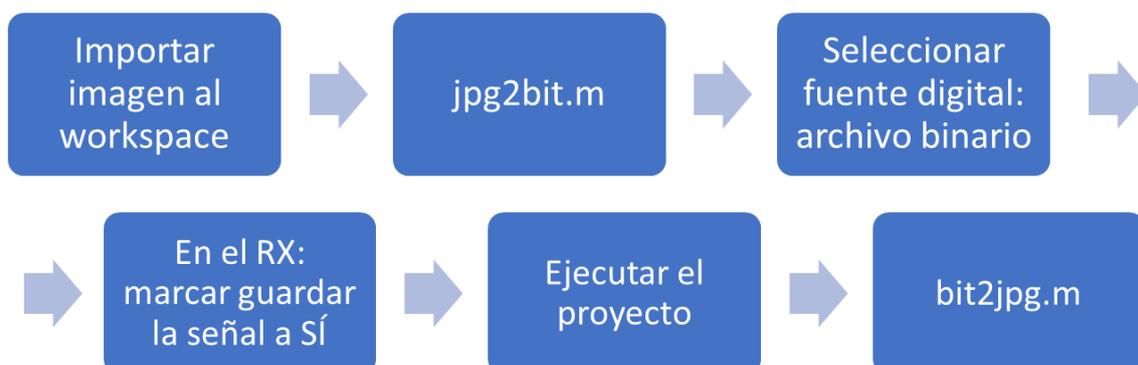


Figura 148. Pasos para utilizar una imagen como fuente digital

En primer lugar, el usuario debe importar la imagen al espacio de trabajo de MATLAB. Esto se puede hacer de varias maneras, aunque lo más recomendado es arrastrar la imagen directamente. Tras arrastrar la imagen saldrá un asistente para importar la imagen, como el de la Figura 149. Pulsar en “Finish”.

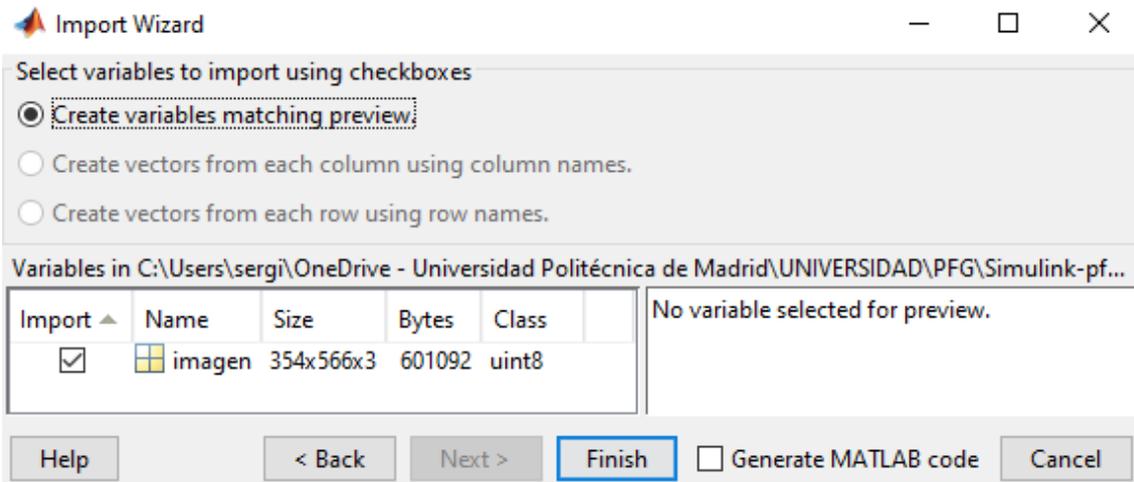


Figura 149. Asistente de importación de imágenes

El segundo paso es ejecutar la función `jpg2bit.m`, que transforma la imagen en un array de bits. El comando es el siguiente: `[img_bits,h,w,d]=jpg2bit(imagen);`

En tercer lugar, seleccionar la fuente de datos digital y poner el nombre de la variable donde se almacena el array de bits. Por defecto es `img_bits`.

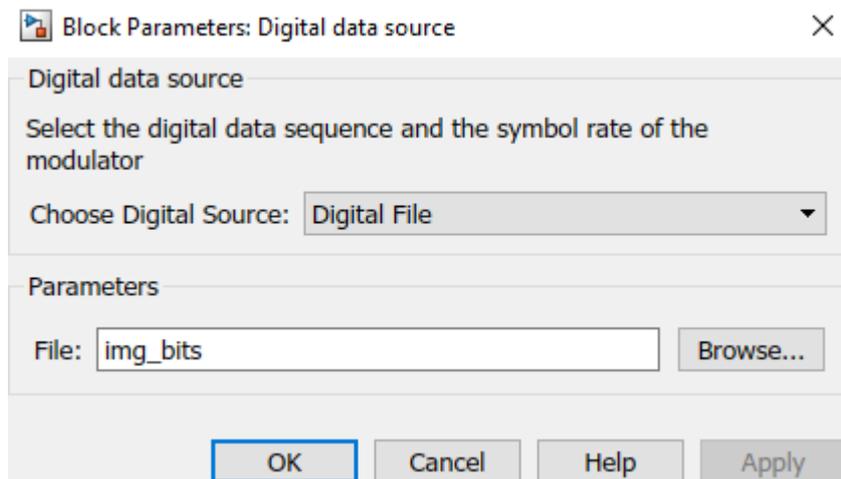


Figura 150. Señales digitales seleccionando imagen del workspace de MATLAB

El cuarto paso es seleccionar en el receptor que se desea guardar la señal recibida en el espacio de trabajo.

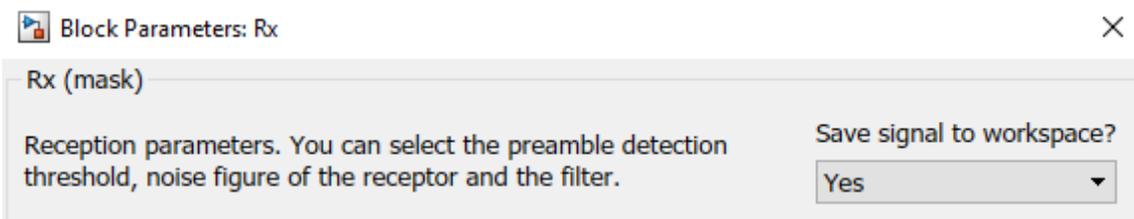


Figura 151. Guardar la señal recibida en el workspace

El quinto paso es ejecutar el proyecto añadiendo los efectos de canal deseados.

Por último, ejecutar desde el workspace la función `bit2jpg.m` para recuperar la señal. El comando es el siguiente: `bit2jpg(y, h, w, d)`. Tras ejecutar la función se genera un fichero en el directorio de trabajo con la imagen recibida con el nombre de `imagen_rx.jpg`.

El resultado de la ejecución es el de la Figura 152.

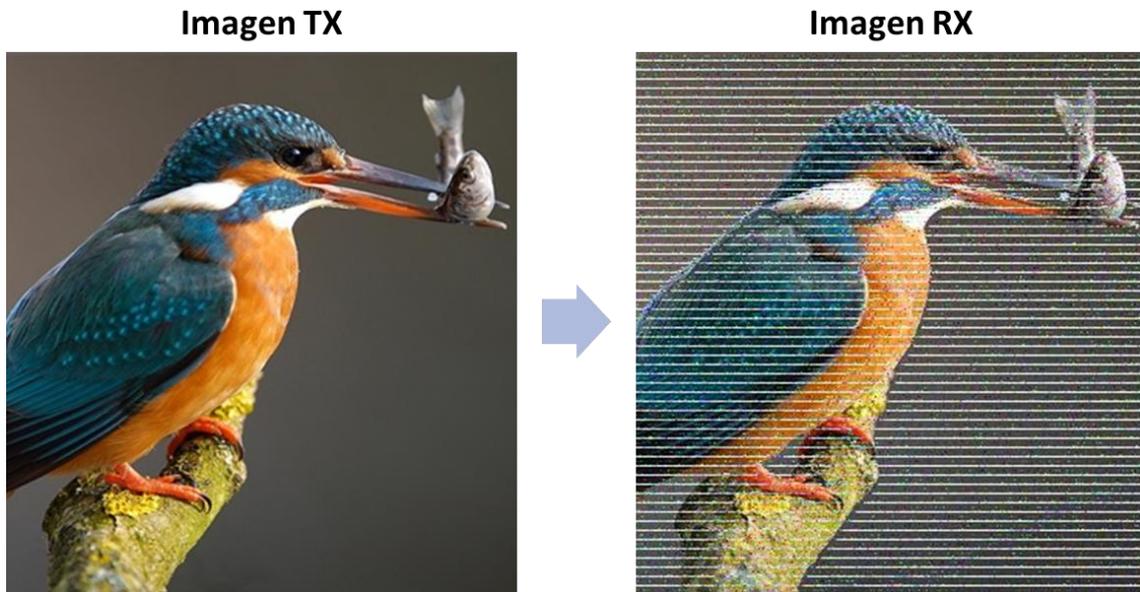


Figura 152. Resultado de la ejecución

9. Conclusiones

9.1 Resumen

El objetivo principal de este proyecto fin de grado es rejuvenecer y actualizar el laboratorio de la asignatura de cuarto semestre Teoría de la Comunicación de la Escuela. El laboratorio, tradicionalmente, ha consistido en utilizar un *script* de MATLAB para realizar medidas de las modulaciones analógicas y utilizar WinIQSIM para estudiar las modulaciones digitales.

Con este proyecto fin de grado se ha conseguido actualizar el laboratorio potenciando las funcionalidades ya existentes y añadiendo dos nuevas funcionalidades: transmitir y recibir señales utilizando SDR y realizar cálculos de BER.

Las principales mejoras se encuentran en que los alumnos disponen de un entorno mucho más visual gracias a la utilización de Simulink. De esta forma, los alumnos pueden identificar los diferentes bloques que participan en un sistema de telecomunicaciones de forma sencilla y mucho más clara que leyendo el código de un *script*.

Aunque se utilice Simulink, se puede seguir utilizando MATLAB para el procesado de las señales de salida, si así lo desea el usuario. De esta manera, el docente tiene la posibilidad de que los alumnos trabajen también con MATLAB.

Los resultados de este proyecto fin de grado son seis prácticas, con su correspondiente solución, para las diferentes áreas de los laboratorios de comunicaciones. En las prácticas propuestas se incluyen muchos apartados para el profesor pueda elegir los que mejor se adapten a su organización.

Para la parte de señal y ruido se proponen dos prácticas: Visualización de Señales en Simulink y Análisis de Señal y Ruido. En estas prácticas se utiliza un proyecto más pequeño para que los alumnos se vayan acostumbrando a la herramienta de forma progresiva.

Para la parte de modulaciones analógicas se proponen otras dos prácticas: Modulaciones AM, DSB y SSB y Modulación FM. En estas prácticas se utiliza el proyecto completo y se anima al profesorado a utilizar el SDR para transmitir las señales generadas y recibirlas con otro SDR. En caso de no disponer de otro SDR se pueden visualizar las señales en los analizadores de espectros del laboratorio.

En tercer lugar, se propone una práctica para Modulaciones Digitales. Esta práctica es muy amplia y, posiblemente, se necesite más de una sesión de laboratorio para completarla. Se incluyen cálculos de tasa de error de bit, visores de constelación, de forma de onda, analizadores de espectro en banda base, diagramas de ojo, diferente tipos de filtros y efectos de canal.

Por último, se propone una práctica de Introducción a los estándares DVB-T y Bluetooth. Esta práctica está orientada a cursos superiores como, por ejemplo, la asignatura de sexto semestre Sistemas de Telecomunicación. Además de explicar los estándares, se realiza una parte de introducción a las técnicas de acceso radio y modulaciones que éstos utilizan, 64 QAM + OFDM para DVB-T y GFSK para Bluetooth.

El objetivo de la práctica es que el alumno conozca estos estándares y el funcionamiento de las interfaces radio de estos sistemas. También es importante que el alumno comprenda la importancia que tiene la utilización de códigos de corrección de errores.

En esta práctica se modifican modelos de los estándares que ya han implementados por los desarrolladores de MATLAB y Simulink. Las modificaciones que se realizan son a nivel de efectos de canal y de cálculo de BER. En esta práctica se visualizará el espectro de los sistemas y se obtienen las curvas de BER antes y después de los códigos de corrección de errores.

9.2 Líneas de trabajo futuro

El diseño de este proyecto fin de grado abre un abanico de oportunidades de innovación y diseño de nuevos proyectos relacionados con la tecnología SDR, Simulink y otras herramientas *software*. Algunas líneas de trabajo futuro que se proponen son:

- Añadir nuevas funcionalidades y característica al programa creado en este proyecto fin de grado. Algunas mejoras que se proponen son: incluir nuevas modulaciones, añadir otras fuentes de información, efectos de canal, codificadores más complejos, añadir más tipos de cabeceras...
- Crear un receptor de SDR utilizando Simulink para recibir diferentes bandas comerciales como radio, televisión, ADS-B...
- Utilizar diferentes dispositivos SDR según las necesidades del departamento.
- Realizar un programa similar utilizando GNU Radio y comparar el rendimiento con el de Simulink.
- Investigar la posibilidad de utilizar un SDR como transmisor y receptor de Bluetooth o de WLAN.

10. Referencias

- [1] P. Johnson, “New Research Lab Leads to Unique Radio Receiver”. *E-Systems Team*, vol. 5, pp 6-7, May 1985.
- [2] J. Mitola, “Software radios-survey, critical evaluation and future directions”, *[Proceedings] NTC-92: National Telesystems Conference*, 1992, pp. 13-23
- [3] R. I. Lackey and D.W. Upmal, “Speakeasy: the military software radio”, *IEEE Communications Magazine*, vol, 33, no.5, pp.55-61, May 1995.
- [4] A. M. Wyglinski, D. P. Orofino, M. N. Ettus and T. W. Rondeau, “Revolutionizing software defined radio: case studies in hardware, software, and education,” in *IEEE Communications Magazine*, vol. 54, no. 1, pp. 68-75, January 2016.
- [5] O. Holland, H. Bogucka and A. Medeisis, “The Universal Software Radio Peripheral (USRP) Family of Low-Cost SDRs,” in *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*, Wiley, 2015, pp.3-23
- [6] “IEEE Standard for Definitions and Concepts for Dynamic Spectrum Access: Terminology Relating to Emerging Wireless Networks, System Functionality, and Spectrum Management,” in *IEEE Std 1900.1-2019 (Revision of IEEE Std 1900.1-2008)*, vol., no., pp.1-78, 23 April 2019
- [7] T.J. Roupael, “The Software-Defined Radio Concept”, in *RF and Digital Signal Processing for Software-Defined Radio*, Burlington, MA, USA: Newnes, 2009.
- [8] B. Riyanto, A. ZR. Langi, A. Kurniawan, E.Marpanaji, A. Mahendra and T. Liung, “Software Architecture of Software-Defined Radio (SDR)” in *ITB Research Center on ICT*, Bandung, Indonesia, 2015, pp. 1-3
- [9] V. Rivero, “La SDR y su rol en la ciberseguridad”, Incibe-cert.es, <https://www.incibe-cert.es/blog/sdr-y-su-rol-ciberseguridad>, (accessed Jun. 23, 2021).
- [10] Afedri Development, “LAN-IQ SDR (Stand-alone SDR)”, Afedri-sdr.com, <http://www.afedri-sdr.com/index.php/lan-iq-sdr>, (accessed Jun. 23, 2021).
- [11] GNU Radio Wiki, “Commercially Available SDR Platforms”, GNURadio.org, https://wiki.gnuradio.org/index.php/Hardware#Commercially_Available_SDR_Platforms (accessed Jun. 20, 2021).
- [12] Native Instruments, “USRP Software Defined Radios”, <https://www.ni.com/es-es/shop/hardware/products/usrp-software-defined-radio-device.html> (accessed Jun. 20, 2021).
- [13] Lime Microsystems, “LimeSDR features & specifications”, Limemicro.com <https://limemicro.com/products/boards/limesdr/> (accessed Jun. 20, 2021).
- [14] Nuand Blade RF, “BladeRF features and technical specifications”, Nuand.com, <https://www.nuand.com/bladerf-1/> (accessed Jun. 20, 2021).

- [15] Analog Devices, “ADALM-Pluto overview: features and benefits”, analog.com, <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-documentation> (accessed Jun. 20, 2021).
- [16] Rafael Micro, “R820T2 High Performance Low Power Advanced Digital TV Silicon Tuner Datasheet”, datasheet, Sept. 2011
- [17] Realtek Semiconductor, “RTL2832U DVB-T COFDM DEMODULATOR + USB 2.0”, JATR-2265-11 datasheet, Nov. 2010
- [18] Lime Microsystems, “LMS7002M FPRF MIMO Transceiver IC”, datasheet, Nov. 2013, [Revised May 2017]
- [19] Analog Devices, “AD9363 RF Agile Transceiver”, datasheet, Nov. 2016
- [20] A.Csete, “ADALM-Pluto SDR Hack: Tune 70 MHz to 6 GHz”, rtl-sdr.com, <https://www.rtl-sdr.com/adalm-pluto-sdr-hack-tune-70-mhz-to-6-ghz-and-gqrx-install/> (accessed Jun. 20, 2021).
- [21] S. Katz and J. Flynn, “Using software defined radio (SDR) to demonstrate concepts in communications and signal processing courses”, *2009 39th IEEE Frontiers in Education Conference*, 2009, pp. 1-6
- [22] Z. Xu, W. Chen, X. Hei and X. Wang, “Design and Evaluation of SDR-based Labs for Learning Principles of Communications”, *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2018, pp. 1049-1052
- [23] R. W. Stewart *et al.*, "A low-cost desktop software defined radio design environment using MATLAB, simulink, and the RTL-SDR", in *IEEE Communications Magazine*, vol. 53, no. 9, pp. 64-71, September 2015
- [24] R. W. Stewart *et al.*, *Software Defined Radio using the MATLAB & Simulink and the RTL-SDR*, Strathclyde Academic Media, 2015.
- [25] M. Petrova, A. Achtzehn and P. Mähönen, "System-oriented communications engineering curriculum: teaching design concepts with SDR platforms," in *IEEE Communications Magazine*, vol. 52, no. 5, pp. 202-209, May 2014
- [26] M. Mateo, *Diseño de un sistema basado en radio definida por software para realizar medidas de canal radio*. Proyecto Fin de Carrera / Trabajo Fin de Grado, E.T.S.I. y Sistemas de Telecomunicación (UPM), Madrid, 2020
- [27] G. Bartolomé, *Diseño de una estación remota de comunicaciones basada en radio software*. Proyecto Fin de Carrera / Trabajo Fin de Grado, E.T.S.I. y Sistemas de Telecomunicación (UPM), Madrid, 2020

- [28] Boletín Oficial del Estado, *Orden IET/1311/2013*, <https://www.boe.es/buscar/doc.php?id=BOE-A-2013-7624> (accessed Jun. 20, 2021).
- [29] *Universal Serial Bus Specification Revision 2.0*, USB 2.0 Specification, 2000
- [30] MathWorks Help Center, “What Are System Objects?”, MathWorks.com, https://es.mathworks.com/help/matlab/matlab_prog/what-are-system-objects.html (accessed Jun. 20, 2021).
- [31] MathWorks Help Center, “Masking fundamentals”, MathWorks.com, <https://es.mathworks.com/help/simulink/ug/block-masks.html> (accessed Jun. 20, 2021).
- [32] *Communications Toolbox User’s Guide*, The MathWorks Inc., Natick, MA, USA, 2020.
- [33] *Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio User’s Guide*, The MathWorks Inc., Natick, MA, USA, 2020
- [34] *DSP System Toolbox User’s Guide*, The MathWorks Inc., Natick, MA, USA, 2020
- [35] *SerDes Toolbox User’s Guide*, The MathWorks Inc., Natick, MA, USA, 2020
- [36] *Pulse Code Modulation (PCM) of Voice Frequencies*, Rec. ITU-T G.711, International Telecommunications Union, Geneva, Switzerland, 1972
- [37] M. Singh, M. Sakare and S. Gupta, “Testing of high-speed DACs using PRBS generation with “Alternate-Bit-Tapping””, *2011 Design, Automation & Test in Europe*, 2011, pp. 1-6.
- [38] S. T. Rappaport, “Fundamentals of Channel Coding”, in *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall PTR, 2002
- [39] D. K. Weaver, “A Third Method of Generation and Detection of Single-Sideband Signals”, in *Proceedings of the IRE*, vol. 44, no. 12, pp. 1703-1705, Dec. 1956
- [40] MathWorks Help Center, “FM Modulator Baseband”, MathWorks.com, <https://es.mathworks.com/help/comm/ref/fmmodulatorbaseband.html> (accessed Jun. 20, 2021).
- [41] S. O. Ajose, A. L. Imoize, O. M. Obiukwu, “Bit error rate analysis of different digital modulation schemes in orthogonal frequency division multiplexing systems” in *Nigerian Journal of Technology*, vol. 37, No.5, pp. 727-734, July, 2018
- [42] *Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, ETSI EN 300 744, European Telecommunications Standards Institute, Valbonne, France, 2009

- [43] MathWorks Help Center, “Digital Video Broadcasting-Terrestrial”, MathWorks.com, <https://es.mathworks.com/help/comm/ug/digital-video-broadcasting-terrestrial.html> (accessed Aug. 20, 2021).
- [44] “About us”, Bluetooth.com, <https://www.bluetooth.com/about-us/> (accessed Sep. 10, 2021).
- [45] “Learn About Bluetooth: Bluetooth Technology Overview”, Bluetooth.com, <https://www.bluetooth.com/learn-about-bluetooth/tech-overview/> (accessed Sep. 10, 2021).
- [46] V. Krishna, “Simulink Model of Bluetooth Modulation and Frequency Hopping”, MATLAB Exchange, <https://www.mathworks.com/matlabcentral/fileexchange/722-simulink-model-of-bluetooth-modulation-and-frequency-hopping> (accessed Sep. 10, 2021).
- [47] MathWorks Help Center, “Spectrum Analyzer: Display frequency spectrum”, MathWorks.com, <https://es.mathworks.com/help/dsp/ref/spectrumanalyzer.html> (accessed Aug. 20, 2021).
- [48] MathWorks Help Center, “Time Scope: Display and analyze signals generated during simulation and log signal data to MATLAB”, MathWorks.com, <https://es.mathworks.com/help/dsp/ref/timescopeblock.html> (accessed Aug. 20, 2021).
- [49] MathWorks Help Center, “Block Callbacks”, MathWorks.com, <https://es.mathworks.com/help/simulink/ug/block-callbacks.html> (accessed Aug. 20, 2021).