



Universidad de  
Oviedo



# **ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

## **MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN**

**ÁREA DE TEORÍA DE LA SEÑAL Y COMUNICACIONES**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE  
COMUNICACIONES ADAPTATIVAS SOBRE DISPOSITIVOS USRP**

**D. Álvaro Pendás Recondo**  
**TUTOR: D. Jesús Alberto López Fernández**

**FECHA: Julio 2021**



# ACRÓNIMOS

- ADC: Analog To Digital Converter
- AM: Adaptive Modulation
- BER: Bit Error Rate
- BNC: Bayonet Neill-Concelman
- BPSK: Binary Phase Shift Keying
- CCE: Canal Continuo Equivalente
- CDE: Canal Discreto Equivalente
- CMA: Constant Modulus Algorithm
- DEP: Densidad Espectral de Potencia
- DAC: Digital To Analog Converter
- DDC: Digital Down Converter
- DUC: Digital Up Converter
- FIR: Finite Impulse Response
- FPGA: Field Programmable Gate Array
- FSE: Fractional Spaced Equalizer
- GRC: GNU Radio Companion
- ISI: Inter-Symbol Interference
- IoT: Internet of Things
- LMS: Least Mean Square
- LTE: Linear Transversal Equalizer
- LTE: Long Term Support
- MIMO: Multiple-Input Multiple-Output
- ML: Maximum Likelihood
- MMSE: Minimun Mean Squared Error



- PDU: Protocol Data Unit
- PER: Packet Error Rate
- PLL: Phase Locked Loop
- PMT: Polymorphic Type
- PSR: Packet Success Rate
- PSK: Phase Shift Keying
- QAM: Quadrature Amplitude Modulation
- QPSK: Quadrature Phase Shift Keying
- RF: Radiofrecuencia
- SDR: Software Defined Radio
- SMA: SubMiniature version A
- SER: Symbol Error Rate
- TFM: Trabajo Fin de Máster
- UHD: USRP Hardware Driver
- USRP: Universal Software Radio Peripheral
- VCO: Voltage Control Oscillator



# NOTACIÓN

- $j$ : Unidad imaginaria.
- $\bar{s}(t)$ : Envolvente compleja de la señal  $s(t)$ . Considerando  $s(t) = s_I(t)\cos(\omega_c t) - s_Q(t)\sin(\omega_c t)$ , la envolvente compleja se define como  $\bar{s}(t) = s_I(t) + js_Q(t)$ .
- $\langle g(t), h(t) \rangle$ : Producto escalar de las señales  $g(t)$  y  $h(t)$ .
- $g(t) * h(t)$ : Convolución de las señales  $g(t)$  y  $h(t)$ .
- $TF \{g(t)\} = G(f)$ : Transformada de Fourier de la señal  $g(t)$ .



# ÍNDICE GENERAL

<b>1. INTRODUCCIÓN</b>	<b>10</b>
1.1.- MOTIVACIÓN . . . . .	10
1.2.- OBJETIVOS . . . . .	11
1.3.- ANTECEDENTES . . . . .	11
1.4.- ESTRUCTURA DE LA MEMORIA . . . . .	13
<b>2. CONCEPTOS BÁSICOS DE COMUNICACIONES DIGITALES</b>	<b>14</b>
2.1.- ESQUEMA GENERAL DE UN SISTEMA DE COMUNICACIONES DIGITALES . . . . .	14
2.2.- MODULACIONES LINEALES . . . . .	19
2.2.1.- PARÁMETROS DE LOS SISTEMAS DE MODULACIÓN DIGITAL . . . . .	19
2.2.2.- MODULACIONES M-PSK . . . . .	22
2.2.3.- MODULACIONES M-QAM . . . . .	24
2.2.4.- COMPARATIVA M-PSK Y M-QAM . . . . .	25
2.3.- TIPOS DE CANALES Y PERTURBACIONES EN UN SISTEMA DE COMUNICACIONES DIGITALES . . . . .	26
2.3.1.- TIPOS DE CANALES . . . . .	26
2.3.2.- PERTURBACIONES . . . . .	28
2.4.- IGUALACIÓN LINEAL DE CANAL . . . . .	31
<b>3. HERRAMIENTAS EMPLEADAS</b>	<b>35</b>
3.1.- GNU RADIO . . . . .	35
3.1.1.- INTRODUCCIÓN . . . . .	35
3.1.2.- BLOQUES EN GNU RADIO Y GRC . . . . .	36
3.1.3.- COMUNICACIÓN ENTRE BLOQUES . . . . .	38
3.2.- DISPOSITIVOS USRP . . . . .	41
3.2.1.- INTRODUCCIÓN . . . . .	41
3.2.2.- ARQUITECTURA GENERAL . . . . .	42
3.2.3.- USRP B200 . . . . .	44
3.2.4.- COMPATIBILIDAD CON GNU RADIO . . . . .	46
<b>4. DISEÑO DE SUBSISTEMAS</b>	<b>48</b>
4.1.- TRANSMISOR BASE . . . . .	48
4.2.- RECEPTOR BASE . . . . .	53



<b>5. SISTEMA FINAL</b>	<b>60</b>
5.1.- EXPLICACIÓN GENERAL DEL SISTEMA . . . . .	60
5.2.- TRANSMISOR . . . . .	62
5.2.1.- Tx_Tx . . . . .	62
5.2.2.- Tx_Rx . . . . .	69
5.3.- RECEPTOR . . . . .	71
5.3.1.- Rx_Rx . . . . .	71
5.3.2.- Rx_Tx . . . . .	77
<b>6. MEDIDAS Y RESULTADOS</b>	<b>82</b>
6.1.- ASPECTOS GENERALES . . . . .	82
6.1.1.- CONFIGURACIÓN DE LOS USRP . . . . .	82
6.1.2.- PROBABILIDAD DE ERROR Y BER . . . . .	83
6.2.- CANAL CONTROLADO POR SOFTWARE . . . . .	86
6.2.1.- INTRODUCCIÓN . . . . .	86
6.2.2.- CANAL GAUSSIANO . . . . .	90
6.2.3.- CANAL LINEAL . . . . .	92
6.2.4.- MODULACIÓN ADAPTATIVA . . . . .	98
6.3.- CANAL INALÁMBRICO . . . . .	102
<b>7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS</b>	<b>105</b>
7.1.- CONCLUSIONES . . . . .	105
7.2.- LÍNEAS DE TRABAJO FUTURAS . . . . .	106
<b>Referencias</b>	<b>108</b>



# ÍNDICE DE FIGURAS

2.1.	Esquema general de un sistema de comunicaciones digitales. . . . .	14
2.2.	Representación de una posible constelación para la modulación QPSK. . . . .	17
2.3.	Representación temporal de la función raíz de coseno alzado para distintos factores de roll-off, $\alpha$ . . . . .	18
2.4.	Representación de una posible constelación para la modulación 2-PSK o BPSK.	22
2.5.	Representación de una posible constelación para la modulación 8-PSK con Codificación Gray. . . . .	23
2.6.	Representación de una posible constelación para la modulación 16-QAM con Codificación Gray. . . . .	25
3.1.	Ejemplo de un esquema en GNU Radio de comparativa de una señal coseno diezmada e interpolada con la original. . . . .	37
3.2.	Representación gráfica de la salida del esquema presentado en la Figura 3.1. . .	38
3.3.	Leyenda de tipos de valores en GRC. . . . .	38
3.4.	Ejemplo de un esquema en GNU Radio en el que se hace uso de las etiquetas de flujo (stream tags). . . . .	39
3.5.	Representación gráfica de la salida del esquema mostrado en la Figura 3.4 . . .	40
3.6.	Ejemplo de un esquema en GNU Radio en el que se hace uso del envío de mensajes (message passing). . . . .	41
3.7.	USRP modelo B200 recubierto con la carcasa diseñada para dicho dispositivo. .	42
3.8.	Arquitectura general de un dispositivo USRP. . . . .	43
3.9.	Placa del USRP modelo B200. . . . .	44
3.10.	Diagrama de bloques del USRP modelo B200. . . . .	45
3.11.	Ejemplo de un esquema en GNU Radio que incluye el bloque UHD: USRP Source.	47
4.1.	Esquema del Transmisor Base en GNU Radio (GRC). . . . .	49
4.2.	Muestras complejas del flujo de símbolos transmitido en el Transmisor Base. .	51
4.3.	Señal digital a transmitir enviada al USRP en el Transmisor Base. . . . .	52
4.4.	Esquema del Receptor Base en GNU Radio (GRC). . . . .	54
4.5.	Señal digital a la salida del bloque Polypahse Clock Syn en el Receptor Base. .	56
4.6.	Constelación BPSK recibida para la cabecera en el Receptor Base. . . . .	57
4.7.	Constelación QPSK recibida para la carga útil en el Receptor Base. . . . .	58
5.1.	Esquema general del sistema de comunicaciones adaptativo. . . . .	60
5.2.	(a) Esquema del Tx_Tx en GNU Radio (GRC). . . . .	64
5.3.	(b) Esquema del Tx_Tx en GNU Radio (GRC). . . . .	65
5.4.	(c) Esquema del Tx_Tx en GNU Radio (GRC). . . . .	66
5.5.	Señal digital a transmitir enviada al USRP utilizando BPSK en la carga útil. . .	68
5.6.	Señal digital a transmitir enviada al USRP utilizando 16-QAM en la carga útil. .	68



5.7. (a) Esquema del Tx_Rx en GNU Radio (GRC). . . . .	70
5.8. (a) Esquema del Rx_Rx en GNU Radio (GRC). . . . .	72
5.9. (b) Esquema del Rx_Rx en GNU Radio (GRC). . . . .	73
5.10. (c) Esquema del Rx_Rx en GNU Radio (GRC). . . . .	74
5.11. Ejemplo de constelaciones recibidas para la cabecera y carga útil. . . . .	75
5.12. Representación de las condiciones de cambio de modulación en la carga útil. . .	77
5.13. Señal digital a transmitir enviada al USRP para una petición al Transmisor de 64-QAM. . . . .	78
5.14. (a) Esquema del Rx_Tx en GNU Radio (GRC). . . . .	79
5.15. (b) Esquema del Rx_Tx en GNU Radio (GRC). . . . .	80
5.16. (c) Esquema del Rx_Tx en GNU Radio (GRC). . . . .	81
6.1. Probabilidad de error de bit frente a $\Gamma_b$ para las modulaciones empleadas. . . .	84
6.2. Probabilidad de error de bit frente a $\sigma^2$ normalizada para las modulaciones em- pleadas. . . . .	85
6.3. Montaje para las medidas del canal controlado por software. . . . .	88
6.4. Medida de la densidad espectral de potencia de la señal transmitida por el Tx_Tx utilizando BPSK en la carga útil. . . . .	89
6.5. Medida de la densidad espectral de potencia de la señal transmitida por el Tx_Tx utilizando 64-QAM en la carga útil. . . . .	90
6.6. $P_b$ teórica y BER medida en función de $\sigma^2$ en canal gaussiano para las distintas modulaciones empleadas en la carga útil. Resultados teóricos y experimentales representados mediante línea continua y asteriscos, respectivamente. . . . .	91
6.7. Constelación de la carga útil para BPSK antes y después de la convergencia de los bloques digitales de la cadena receptora para un canal multicamino. . . . .	93
6.8. Constelación de la carga útil para 64-QAM antes y después de la convergencia de los bloques digitales de la cadena receptora para un canal multicamino con efecto Doppler. . . . .	94
6.9. Constelación de la carga útil para 16-QAM antes y después de la convergencia de los bloques digitales de la cadena receptora para un desvanecimiento en potencia de 3 dB. . . . .	94
6.10. $P_b$ teórica y BER medida en función de $\sigma^2$ para BPSK en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (aste- riscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD). . . . .	95
6.11. $P_b$ teórica y BER medida en función de $\sigma^2$ para QPSK en la carga útil. Result- ados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multica- mino y efecto Doppler (MD). . . . .	96





6.12. $P_b$ teórica y BER medida en función de $\sigma^2$ para 16-QAM en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD). . . . .	97
6.13. $P_b$ teórica y BER medida en función de $\sigma^2$ para 64-QAM en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD). . . . .	98
6.14. $\tilde{P}_b$ estimada en función de $\tilde{\sigma}^2$ para las distintas modulaciones empleadas en la carga útil. Las medidas se corresponden con las del Apartado 6.2.4. Resultados teóricos y experimentales representados mediante línea continua y asteriscos, respectivamente. . . . .	99
6.15. $\tilde{P}_b$ estimada en función de $\tilde{\sigma}^2$ . Representación de los cambios entre las distintas modulaciones de la carga útil para $BER_{max} = 10^{-4}$ . . . . .	101
6.16. Montaje del sistema para el canal inalámbrico. . . . .	102
6.17. Constelación de la cabecera y la carga útil una vez alcanzada 16-QAM en la segunda. . . . .	103
6.18. Montaje del sistema para el canal inalámbrico introduciendo un efecto multicamino. . . . .	104
6.19. Constelación de la carga útil (16-QAM) antes y después de la convergencia al modificar el escenario para introducir un canal con efecto multicamino. . . . .	104



# ÍNDICE DE TABLAS

4.1. Estructura de los paquetes generados por el Transmisor Base. CA indica Código de Acceso y LCU Longitud de la Carga Útil. . . . .	53
5.1. Estructura de los paquetes generados por el Tx_Tx. La leyenda de campos es la siguiente: C - Cabecera; CU - Carga Útil; CA - Código de Acceso; LCU - Longitud de la Carga Útil; BPS - Bits Por Símbolo en la carga útil; CP - Contador de Paquete; RS - Reseteo de Estado. Los valores entre paréntesis indican longitud en bytes. En los casos en los que se indican 4 valores estos se corresponden con la utilización en la carga útil de BPSK, QPSK, 16-QAM y 64-QAM respectivamente. 63	
5.2. Estructura de los paquetes generados por el Rx_Tx. La leyenda de campos es la siguiente: C - Cabecera; CU - Carga Útil; CA - Código de Acceso; LCU - Longitud de la Carga Útil; CM - Código de Modulación. . . . .	78
6.1. Resumen de la configuración de los USRP. . . . .	82
6.2. Configuración de los umbrales en las instancias de Dispersion Probe para $BER_{max} = 10^{-4}$ . . . . .	100
6.3. Configuración de los valores mínimos de PSR en el bloque CRC32 Check para $BER_{max} = 10^{-4}$ . . . . .	100



# 1. INTRODUCCIÓN

## 1.1.- MOTIVACIÓN

Las comunicaciones digitales ofrecen, en la mayoría de aplicaciones de uso masivo y frente a los sistemas analógicos, ciertas ventajas destacables, entre ellas: un abaratamiento de los costes; capacidad de regeneración de las señales, así como detección y corrección de errores; y, en general, una mayor versatilidad y capacidad adaptativa. Todas estas características son de especial interés en el ámbito de las comunicaciones móviles, tanto en lo que respecta a su comercialización como a la calidad de las aplicaciones. Por otro lado, en el caso de los sistemas inalámbricos, esta última depende cada vez más de la capacidad para adaptarse según el entorno en el que se desarrolla la comunicación.

La tecnología Software Defined Radio (SDR), se basa el diseño de sistemas de comunicaciones en los que elementos de la capa física, tradicionalmente implementados mediante hardware electrónico, pasan a ser programados por software. Así, por ejemplo, usando técnicas de procesamiento de señal en lugar de circuitería es posible implementar filtros o demoduladores sobre el sistema SDR. Las ventajas que ofrece SDR frente a otras comunicaciones digitales más dependientes de elementos físicos, son similares a las que ofrecen, en general, las comunicaciones digitales frente a las analógicas. Por un lado, se abaratan los costes de fabricación enormemente. Por otro, se aumenta considerablemente la versatilidad y se abren nuevas posibilidades en el campo de las comunicaciones adaptativas al poder modificar los elementos del sistema mediante software. Por último, se aumenta la robustez frente a posibles fallos de dispositivos físicos.

No obstante, como no podría ser de otra manera, los sistemas de SDR también presentan una serie de limitaciones. Entre ellas, el drástico aumento del coste computacional y la dificultad de reducir el tiempo de conmutación de los dispositivos de muestreo, siendo estas las principales limitaciones actualmente y la razón por la que muchos de los bloques de hardware no serán sustituidos a corto plazo. Esto es especialmente cierto en la banda de radiofrecuencia y microondas, donde las altas frecuencias hacen especialmente difícil el muestreo y tratamiento digital de las señales.

El propósito de este Trabajo Fin de Máster (TFM) es implementar un sistema de comunicaciones digitales que sea capaz de adaptarse a las condiciones de un canal variante explotando las ventajas que ofrecen los sistemas SDR. Las herramientas empleadas para esta tarea serán principalmente dos. En primer lugar, y para la parte de procesamiento digital, el programa de software libre y código abierto GNU Radio [1]. En segundo lugar, y referido a la parte hardware del sistema, los dispositivos Universal Software Radio Peripheral (USRP) del fabricante Ettus Research [2].



## 1.2.- OBJETIVOS

El objetivo de este TFM es implementar un sistema de comunicaciones digitales capaz de adaptarse a las condiciones de un canal variante en el tiempo. Tal es el caso, por ejemplo, de un escenario de comunicaciones móviles. Este propósito requiere del análisis y la utilización de las prestaciones que ofrece el paradigma SDR

Con este fin se estudiarán las prestaciones ofrecidas por la combinación de la herramienta de software GNU Radio con los dispositivos físicos USRP. Si bien los subsistemas se desarrollarán primero mediante simulaciones de software, el objetivo último será siempre su funcionamiento en la implementación física. Se actuará tanto en el lado del transmisor como del receptor, y se contrastarán los resultados finales con los de las simulaciones, así como con los esperados teóricamente, analizando las causas de las posibles diferencias.

El sistema final será diseñado en base a los resultados y conclusiones que se vayan extrayendo en el análisis de sistemas más sencillos. Esta contará con mecanismos de protección frente a multicamino, desplazamientos en frecuencia, desvanecimientos en potencia y fallos de sincronismo en los relojes de transmisión y recepción. Además, se implementará un canal de retorno entre el receptor y el transmisor, de manera que el primero pueda informar de cómo está transcurriendo la comunicación para que ambos puedan adaptar la modulación utilizada en el sistema según las condiciones de un canal variante en el tiempo. Todo ello se realizará de manera continua y en tiempo real, sin necesidad de interrumpir el flujo de información ni reiniciar la comunicación.

El análisis y estudio de los sistemas implementados se centrará en la probabilidad de error y la robustez frente a las perturbaciones descritas. Para la prueba de los sistemas se emplearán distintos escenarios que den lugar a canales diferentes, así como un canal digital añadido en el transmisor y controlado por software que permite añadir perturbaciones adicionales a las introducidas por dichos canales físicos.

## 1.3.- ANTECEDENTES

Desde la implementación del primer sistema de unidireccional de telefonía móvil, a principios del siglo XX, comenzó a cobrar especial importancia el uso de algoritmos y sistemas capaces de adaptarse a canales dispersivos en frecuencia y variantes en tiempo. En este ámbito, grandes avances fueron realizados durante la Segunda Guerra Mundial en el campo de las aplicaciones militares. Más adelante, y tras la implementación de la primera red celular pública en Arabia Saudita en 1981, la llegada de los primeros sistemas celulares digitales en la década de los 90 permitió la comercialización masiva de los dispositivos móviles, hoy imprescindibles en nuestra sociedad [3].

Paralelamente al crecimiento exponencial de los dispositivos móviles personales, experimentaron un crecimiento similar infinidad de aplicaciones y tecnologías que también



necesitaban, y necesitan, de una capacidad adaptativa en sus comunicaciones. Por citar algunos ejemplos se pueden destacar las comunicaciones aeronáuticas y satelitales. El auge de las comunicaciones digitales presentó una serie de beneficios innegables en lo que se refiere a la detección y corrección de errores, pero también facilitó enormemente la adaptación de los sistemas tanto en transmisión como en recepción frente a canales variantes en el tiempo [4].

Ciertamente, en un radioenlace fijo, o un enlace por cable, también aparecen perturbaciones y/o interferencias importantes. Claros ejemplos serían el multicamino en el primer caso, o la dispersión intermodal en el segundo<sup>1</sup>, además de la atenuación y el ruido gaussiano en ambos. No obstante, si bien son canales dispersivos en frecuencia, estos no presentan una variación significativa en el tiempo. De esta manera, un estudio previo a la implantación del sistema pueda bastar para que este pueda funcionar correctamente sin necesidad de adaptación. Sin embargo, la variación de las características del canal que se producen en un escenario de comunicaciones móviles hace imprescindible la adaptabilidad del sistema para poder mantener una tasa de transmisión y una probabilidad de error adecuadas satisfaciendo unos requisitos de potencia y ancho de banda aceptables.

Con la implementación de las comunicaciones y el procesado digitales, que se introdujeron en las redes móviles en el estándar GSM (sistemas 2G) [4], se abrió un nuevo mundo de posibilidades en las comunicaciones adaptativas. El procesado digital permitió la implementación de filtros adaptativos, y en los sistemas 3.5G se presentó la posibilidad de adaptar la modulación empleada a las condiciones del canal [5]. Otro ejemplo de evolución en las comunicaciones móviles más allá de la telefonía, son los sistemas de televisión y radiodifusión por medio de radioenlaces, con los estándares digitales de DAB y DVB respectivamente, que actualmente predominan en Europa. Dichos estándares permiten la configuración de distintos modos de funcionamiento según el entorno, y además incorporan diversos mecanismos para la prevención, detección y corrección de errores [6], [7] y [8].

En la actualidad, con el drástico aumento de la demanda de redes de datos vía dispositivos móviles y el desarrollo de nuevos estándares como 5G, el estudio de las comunicaciones adaptativas se ha vuelto indispensable en el avance de estas nuevas tecnologías. Entre ellas, se incluye también el nuevo paradigma de Internet of Things (IoT) y los estándares que utiliza, tales como Bluetooth, ZigBee o Wi-Fi. Todos estos estándares de redes inalámbricas, que por otro lado 5G busca intercomunicar y emplear [9], dependen cada vez más de su capacidad para adaptarse al entorno. Esto se debe a que las tasas de transmisión requeridas son cada vez más altas, a la vez que se busca reducir las situaciones que puedan propiciar la pérdida de la comunicación.

Ante esta realidad, los sistemas de SDR se presentan como una alternativa barata, potente y versátil, fácilmente implementable en todo tipo de dispositivos con independencia del

---

<sup>1</sup>Considerando un enlace por fibra óptica.



estándar empleado en las capas superiores. Además, se añade la ventaja crucial que supone su reutilización, pudiendo implementarse en millones de dispositivos con un bajo coste. Esta última característica es posiblemente la más importante de cara a la comercialización masiva que ya es una realidad para los dispositivos móviles, y cuyas perspectivas son de aún mayor crecimiento en el corto y medio plazo.

Por último, la versatilidad, configurabilidad y bajo coste de las herramientas de SDR resultan también características especialmente atractivas en el entorno investigador, en el que su popularidad ya manifiesta es cada vez mayor. Algunos ejemplos claros de esta realidad se encuentran en campos como el diseño de sistemas Multiple-Input Multiple-Output (MIMO) [10], la cancelación digital de interferencias [11], o la aplicación de técnicas de Deep Learning a las comunicaciones inalámbricas [12].

## 1.4.- ESTRUCTURA DE LA MEMORIA

La presente memoria se divide en siete capítulos. Tras un primer capítulo de introducción, el segundo se centra en presentar los fundamentos teóricos básicos de las comunicaciones digitales sobre los que se va a trabajar. A su vez, el tercer capítulo sirve de descripción de las principales herramientas empleadas, a saber, GNU Radio en la parte de software y los dispositivos USRP en la de hardware.

El cuarto capítulo está dedicado a la implementación de distintos subsistemas que serán utilizados en el desarrollo del sistema final. El quinto capítulo describe la combinación de los subsistemas y otros elementos adicionales en un único sistema integrado. Las medidas y resultados obtenidos se detallan en el sexto capítulo. Por último, el séptimo capítulo se reserva para las conclusiones y las líneas de trabajo futuro.

Junto con esta memoria se adjunta además el código fuente de todos los bloques de procesamiento digital programados, así como los archivos ejecutables de cada uno de los sistemas presentados.

## 2. CONCEPTOS BÁSICOS DE COMUNICACIONES DIGITALES

En este capítulo se introducen algunos fundamentos básicos de las comunicaciones digitales. Se busca así recoger las bases teóricas de cara a abordar el diseño del sistema desarrollado. No obstante, los conceptos aquí expuestos se presentan de manera breve y esquemática, siendo estos descritos en detalle en las dos referencias principales aquí utilizadas, [13] y [14].

### 2.1.- ESQUEMA GENERAL DE UN SISTEMA DE COMUNICACIONES DIGITALES

Con el objetivo de introducir el funcionamiento de un sistema de comunicaciones digitales, se partirá del esquema general presentado en la Figura 2.1, que se corresponde con el descrito en [13]. Por simplicidad, dicho esquema será particularizado para un sistema monoportadora y un canal gaussiano. La modulación escogida para este caso es una Quadrature Phase Shift Keying (QPSK). Como se desarrollará más adelante, durante este trabajo no se usarán modulaciones de más de dos dimensiones, lo que convierte a QPSK en una modulación adecuada para este ejemplo.

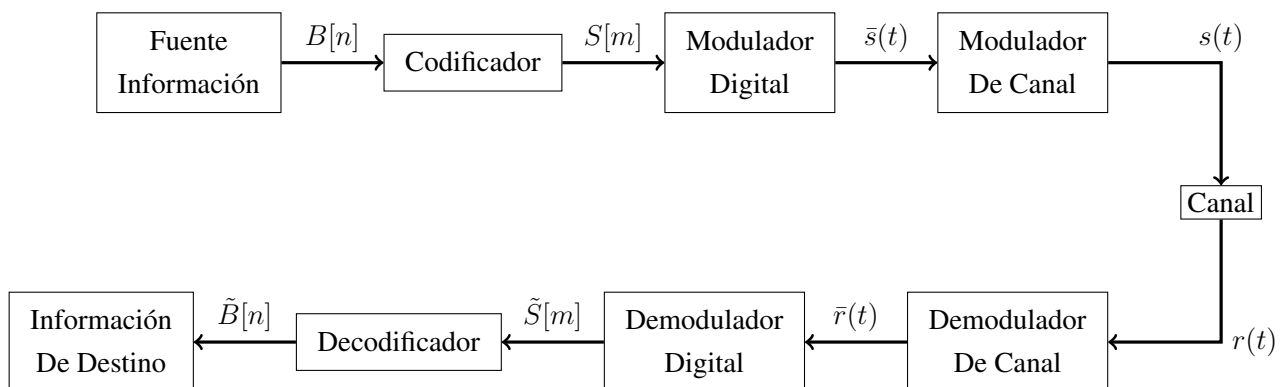


Figura 2.1.- Esquema general de un sistema de comunicaciones digitales.

En primer lugar, se parte de la Fuente de Información, la cual genera una secuencia binaria de dígitos 0 y 1, denotada por  $B[n]$ . El objetivo de la comunicación es transmitir dicha secuencia a través de un canal, de manera que el mensaje binario recibido se corresponda con la secuencia original. El segundo y penúltimo bloque en la Figura 2.1 sirven para la codificación de la información, tratándose del Codificador y Decodificador respectivamente. Si bien las aplicaciones pueden variar enormemente según el código empleado, en este caso se considerará que su única función es la corregir y/o detectar errores en la secuencia recibida. Para ello, empleará un código conocido por el Decodificador, realizando este el proceso in-



verso. El Modulador Digital tomará los bits de salida del codificador en agrupaciones de  $N$  bits, correspondiéndose cada agrupación con un símbolo de la secuencia  $S[m]$ .

El Modulador Digital del transmisor puede dividirse desde el punto de vista teórico en dos bloques: el Modulador Discreto y el Modulador de Señal. La función del primero consiste en asociar símbolos a coordenadas. Las coordenadas, a su vez, también vienen definidas por la modulación empleada. Cada símbolo se asigna a una coordenada y el número de símbolos  $M$  es igual a  $2^N$ . La Figura 2.2 muestra un esquema de la constelación QPSK [13], de  $N = 2$ . Es importante destacar también que la modulación QPSK es una modulación de dos dimensiones,  $L = 2$ , lo que es independiente del número de bits por símbolo. Esto significa que son necesarios dos números reales, o uno complejo, para expresar las coordenadas de cada uno de sus símbolos. En estos casos, es habitual hablar de componente en cuadratura (Q) para referirse a la parte imaginaria, denotándose la real como componente en fase (I) [14].

De esta manera, se genera a la salida del Modulador Discreto una secuencia discreta de números complejos que se denotará por:

$$A[m] = A_I[m] + jA_Q[m] \quad (2.1)$$

Dicha secuencia se introduce en el Modulador de Señal, bloque que genera la señal continua a partir de  $A[m]$  y una señal (o función) base, que en este caso se considera real,  $\phi(t)$  según (2.2), donde  $T$  es el período de símbolo, es decir, el tiempo que pasa entre que se transmite un símbolo y el siguiente <sup>2</sup>.

$$\bar{s}(t) = \sum_{m=-\infty}^{m=\infty} A[m]\phi(t - mT) \quad (2.2)$$

Es importante destacar que en (2.2) se representa lo que se conoce como la envolvente compleja de la señal. Durante este trabajo se utilizará frecuentemente dicha notación. Para más información acerca de la envolvente compleja y otras formas de representar las señales se recomienda consultar [14].

El último bloque del transmisor es el Modulador de Canal, cuya función es desplazar la señal en banda base a la banda de frecuencias deseada. La señal paso banda puede expresarse a partir de sus componentes I-Q según (2.3). La pulsación angular de la frecuencia central paso banda se denota por  $\omega_c = 2f_c$ , siendo  $f_c$  la frecuencia de canal, mientras que  $\theta$  hace referencia a una fase aleatoria introducida por el Modulador de Canal. De esta forma, la señal modulada se puede expresar como:

$$s(t) = \sum_{m=-\infty}^{m=\infty} A_I[m]\phi(t - mT) \cos(\omega_c t + \theta) - \sum_{m=-\infty}^{m=\infty} A_Q[m]\phi(t - mT) \sin(\omega_c t + \theta) \quad (2.3)$$

<sup>2</sup>Una forma de ver el Modulador Digital es como un conversor digital-analógico de la secuencia de números complejos  $A[m]$  a una tasa  $T$  empleando un filtro de interpolación de respuesta  $\phi(t)$ .





El primer bloque del receptor, el Demodulador de Canal, realiza justo la tarea inversa, desplazando la señal recibida a banda base.

El canal más básico en un sistema de comunicaciones digitales es conocido como canal gaussiano, escogido para este ejemplo. En él, la única perturbación considerada es el ruido térmico modelado como ruido blanco gaussiano. Canales más complejos que introducen un mayor número de perturbaciones serán tratados en el Apartado 2.3. El ruido térmico (Apartado 2.3.2), también conocido como ruido de Johnson–Nyquist, se genera por la agitación térmica de los portadores de carga. Dicho ruido es modelable mediante un proceso aleatorio cuyas muestras son variables aleatorias gaussianas de media cero, y es aditivo respecto de la señal. De esta manera, la señal que llega a la entrada del receptor  $r(t)$  se puede expresar según:

$$r(t) = s(t) + n(t), \quad (2.4)$$

donde  $n(t)$  es la señal de ruido.

Tras el Demodulador de Canal, el Demodulador Digital convierte la señal continua de nuevo a una secuencia de símbolos,  $\tilde{S}[m]$ , que el Decodificador transforma en una secuencia de bits,  $\tilde{B}[n]$ . El Demodulador Digital consta también de dos partes, análogas a las del Modulador Digital. En primer lugar, el Demodulador de Señal transforma la señal a coordenadas expresadas en el mismo espacio de señal donde se define la modulación (Figura 2.2). Para ello, es necesario un filtro adaptado a la señal base empleada en transmisión, a cuya salida se muestrea en los múltiplos del período de símbolo,  $mT$ . Dicha operación equivale al producto escalar entre la señal recibida y la señal base desplazada:

$$R[m] = A[m] + n[m] = \bar{r}(t) * \phi^*(-t)|_{t=mT} = \int_{-\infty}^{\infty} \bar{r}(\tau) \phi^*(\tau - mT) d\tau = \langle \bar{r}(t), \phi(t - mT) \rangle \quad (2.5)$$

De manera que las muestras de salida del filtro adaptado del receptor en los instantes  $mT$  se reducen, en canal gaussiano, a la secuencia de símbolos transmitidos  $A[m]$  más una secuencia de muestras de ruido  $n[m]$  que se puede modelar como una secuencia de variables aleatorias gaussianas independientes e idénticamente distribuidas. Más información acerca de sus posibles valores se recoge en el Apartado 2.3.2. Este proceso resulta así en una secuencia  $R[m]$  a la salida, que la segunda parte del Demodulador Digital, el Decisor, convertirá a una secuencia de símbolos.

Llegados a este punto, es importante hacer varias aclaraciones respecto de la señal base empleada y el ruido. En primer lugar, se ha supuesto que el sistema está libre de interferencia entre símbolos (ISI), o equivalentemente que la transmisión de un símbolo no afecta ni a las observaciones anteriores, ni a las observaciones posteriores. A tal fin, la función base empleada debe cumplir (2.6). En otras palabras, el conjunto de funciones generado a partir

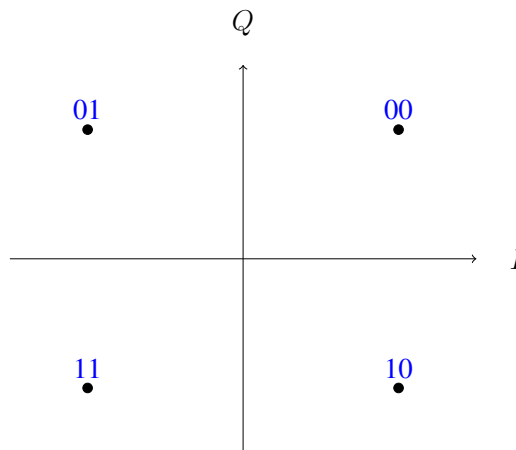


Figura 2.2.- Representación de una posible constelación para la modulación QPSK.

de desplazar la señal  $\phi(t)$  deben formar una base ortonormal<sup>3</sup>.

$$\langle \phi(t), \phi(t - mT) \rangle = \delta_{0,m} \begin{cases} 1 & \text{si } m = 0 \\ 0 & \text{si } m \neq 0 \end{cases} \quad (2.6)$$

El cumplimiento de la condición (2.6) permite garantizar también que las muestras de ruido son independientes dos a dos, tal y como se ha supuesto inicialmente. Ambas propiedades permiten que la toma de decisiones (acción del Decisor) se pueda efectuar de manera independiente para cada símbolo  $R[m]$ . En el caso de que la probabilidad a priori de los símbolos transmitidos sea desconocida, se consideran equiprobables. En este TFM se asumirá siempre equiprobabilidad, lo que implica que el criterio más adecuado para la decisión es el criterio de Máxima Verosimilitud (ML) [13]. Adicionalmente, el funcionamiento del Decisor puede verse condicionado por el código empleado por el Codificador.

Además, para el caso de una modulación de fase, como es QPSK, hay que tener en cuenta la ambigüedad de fase, obviada en este análisis. Esta proviene de la señal portadora, generada en el Modulador de Canal, generalmente de naturaleza senoidal, frecuencia  $f_c$  y fase inicial aleatoria  $\theta$ . Dicha ambigüedad se puede expresar en términos de la envolvente compleja como el producto de  $\bar{s}(t)$  por una exponencial  $e^{j\theta}$ . Para que la comunicación no se vea afectada, es necesario un sistema de recuperación de portadora que estime la fase de la señal recibida. A efectos del esquema de la envolvente compleja, la recuperación de la señal moduladora en banda base se consigue multiplicando la señal recibida  $\bar{r}(t)$  por la exponencial de la fase estimada cambiada de signo  $e^{-j\hat{\theta}}$ . Más información acerca de las modulaciones de fase, como es QPSK, se presenta en el Apartado 2.2.2.

Por último, si bien se han establecido las condiciones que debe de cumplir la función base (2.6), no se ha definido ninguna posible función que las reúna. Es posible demostrar [13] que la función sinc normalizada (2.7) cumple (2.6) y además ocupa el mínimo ancho de

<sup>3</sup>Según el álgebra de un espacio de Hilbert de señales definido en [13].



banda posible para un período de símbolo,  $T$ , dado.. Nótese que, aplicando la Transformada de Fourier (TF), en el dominio frecuencia la función sinc se corresponde con un pulso, en este caso de duración  $T$ .

$$\phi(t) = \frac{1}{\sqrt{T}} \text{sinc} \left( \frac{t}{T} \right) = \frac{1}{\sqrt{T}} \frac{T \sin(\frac{\pi t}{T})}{\pi t} \quad (2.7)$$

No obstante, esta función no es físicamente realizable<sup>4</sup>, por lo que se puede recurrir, sacrificando ancho de banda, a emplear como base la función comúnmente conocida como raíz de coseno alzado [13]. Esta se define según (2.8), donde  $\alpha \in [0, 1]$  denota lo que comúnmente se denomina factor de roll-off o de ensanchamiento espectral.

$$\phi(t) = \frac{4\alpha}{\pi\sqrt{T}} \frac{\cos \left( (1 + \alpha) \frac{\pi t}{T} \right) + \frac{T \sin \left( (1 - \alpha) \frac{\pi t}{T} \right)}{4\alpha t}}{1 - (4\alpha t/T)^2} \quad (2.8)$$

Puede observarse que, mientras que un  $\alpha = 0$  daría lugar a una sinc, cuanto mayor sea el factor de roll-off, más rápido decrece la función en el tiempo a cambio de ensanchar su espectro. La Figura 2.3 muestra la representación temporal de la función raíz de coseno alzado para distintos factores de roll-off.

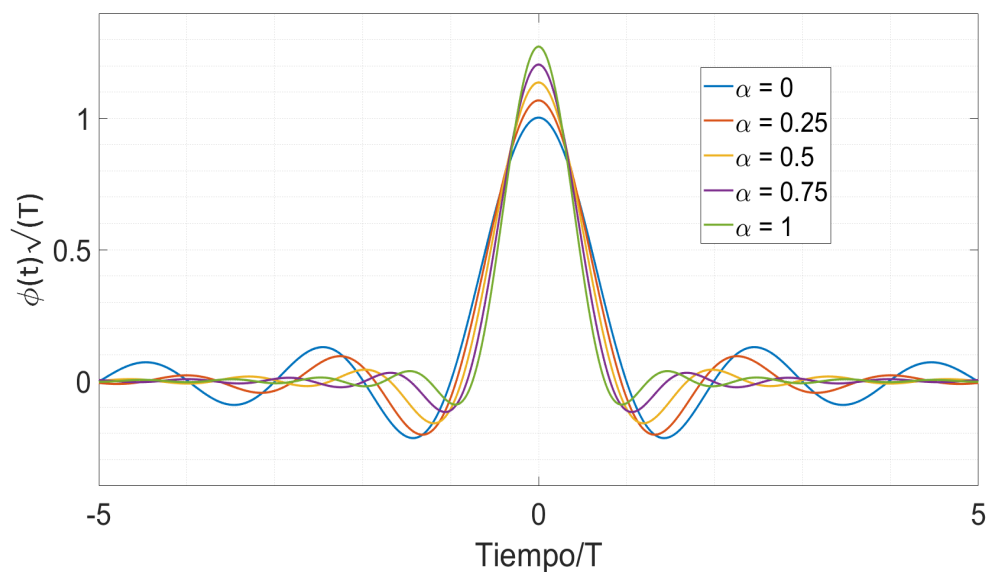


Figura 2.3.- Representación temporal de la función raíz de coseno alzado para distintos factores de roll-off,  $\alpha$ .

En realidad, la función coseno alzado tampoco sería físicamente realizable, pero dado que decrece mucho más rápidamente que la función sinc, un truncado temporal adecuado apenas afectará a su comportamiento.

<sup>4</sup>Esto es debido a que es una función no causal y de soporte infinito (contiene valores no nulos en todo el eje de tiempos).



En el dominio de la frecuencia, el ancho de banda de la raíz de coseno alzado viene dado por:

$$W = \frac{1 + \alpha}{2T}, \quad (2.9)$$

donde se observa que  $\alpha$  indica el ensanchamiento espectral, en tanto por uno, de la forma de onda de tipo raíz de coseno alzado respecto de la función sinc para un  $T$  dado.

## 2.2.- MODULACIONES LINEALES

Este apartado se centra en el estudio de las modulaciones lineales que se emplearán durante el presente TFM. En primer lugar, se analizarán los parámetros básicos una modulación digital genérica, así como sus límites, para después analizar y comparar las modulaciones M-Quadrature Amplitude Modulation (M-QAM) y M-Phase Shift Keying (M-PSK).

Como ya se había adelantado en el Apartado 2.1, todas las constelaciones de las modulaciones aquí descritas pueden definirse sobre un espacio real de dos dimensiones  $L = 2$ . Sus coordenadas, por tanto, pueden ser descritas por medio de un único número complejo. La notación empleada para referirse a cada una de sus componentes será la I para la componente en fase, o componente real, y Q para la componente en cuadratura o componente imaginaria.

### 2.2.1.- PARÁMETROS DE LOS SISTEMAS DE MODULACIÓN DIGITAL

A continuación, se enumeran algunos de los parámetros más característicos de los sistemas de modulación digital [13]:

- $N$ : Promedio de bits por símbolo.
- $\mathcal{E}_{av}$ : Energía media de la constelación. Se calcula según (2.10), donde  $P_i$  es la probabilidad del símbolo  $i$ -ésimo y  $\mathcal{E}(a_i)$  la energía de cada símbolo considerando  $a_i$  la representación en espacio de señal del símbolo  $i$ -ésimo. Cada  $a_i$  se corresponde con una realización de la envolvente compleja señal  $\bar{s}_i$  para una señal paso banda, o de la propia señal  $s_i$  para una señal en banda base. En realidad, la energía media de la señal se puede definir como un promedio de las distancias al origen al cuadrado de cada uno de sus símbolos sobre la constelación.

$$\mathcal{E}_{av} = \sum_{i=0}^{M-1} P_i |a_i|^2 \quad (2.10)$$



- $\mathcal{E}_0$ : Energía media de las señales transmitidas:

$$\mathcal{E}_0 = \begin{cases} \mathcal{E}_{av} & \text{para señales reales} \\ \frac{1}{2}\mathcal{E}_{av} & \text{para señales complejas} \end{cases} \quad (2.11)$$

- $P_0$ : Potencia media de las señales transmitidas:

$$P_0 = \frac{\mathcal{E}_0}{T} \quad (2.12)$$

- $\mathcal{E}_b$ : Energía media por bit transmitido:

$$\mathcal{E}_b = \frac{\mathcal{E}_0}{N} \quad (2.13)$$

- $\gamma_b$ :  $\mathcal{E}_b$  normalizada o relación señal a ruido por bit:

$$\gamma_b = \frac{\mathcal{E}_b}{N_0}, \quad (2.14)$$

donde  $N_0$  representa la densidad espectral de potencia del ruido (W/Hz) (Apartado 2.3.2). Es común expresar  $\gamma_b$  en dB, en cuyo caso se denota como  $\Gamma_b$ :

$$\Gamma_b = 10 \log_{10}(\gamma_b) \text{ [dB]} \quad (2.15)$$

- $\beta$ : Ganancia geométrica o factor de mérito de la constelación. Se entiende como un parámetro de calidad acerca de cómo la constelación aprovecha su energía para separar las señales (2.16). La distancia mínima entre símbolos se representa por  $d$ .

$$\beta = \frac{d^2}{\mathcal{E}_{av}} \quad (2.16)$$

- $\nu$ : Eficiencia espectral. Se define según (2.17), donde  $R$  es la tasa binaria y  $W$  el ancho de banda empleado.

$$\nu = \frac{R}{W} \left[ \frac{\text{bits}}{\text{s Hz}} \right] \quad (2.17)$$

- $\rho$ : Eficiencia en potencia. Se define según (2.18). Hay que tener en cuenta que una determinada eficiencia en potencia siempre se asocia a una probabilidad de error en canal gaussiano determinada.

$$\rho = \frac{1}{\gamma_b} \quad (2.18)$$



A la hora de comparar dos modulaciones distintas, aquí denotadas como 1 y 2, se define la ganancia en potencia de una modulación frente a otra  $g_{1,2}$  según (2.19).  $g_{1,2}$  es el ahorro en potencia al emplear la modulación 1 en vez de 2 para, bajo el mismo canal gaussiano (misma  $N_0$ ), y ofreciendo la misma tasa binaria  $R$ , obtener una misma probabilidad de error de símbolo  $P_E$ . La ganancia en expresada en dB se define en (2.20).

$$g_{1,2} = \frac{\gamma_{b,2}}{\gamma_{b,1}} \quad (2.19)$$

$$G_{1,2} = 10 \log_{10}(g_{1,2}) \text{ [dB]} \quad (2.20)$$

Se puede demostrar [13] que  $\gamma_b$  está acotada según (2.21), donde  $Q^{-1}()$  denota la función  $Q$  inversa,  $k$ , el promedio de símbolos vecinos que tiene cada uno de los símbolos y  $P_E$ , la probabilidad de error de símbolo:

$$\underbrace{\frac{2}{\beta N} [Q^{-1}(P_E)]^2}_{\gamma_{b,inf}} \leq \gamma_b \leq \overbrace{\frac{2}{\beta N} \left[ Q^{-1} \left( \frac{P_E}{k} \right) \right]^2}_{\gamma_{b,sup}} \quad (2.21)$$

Además, también se puede comprobar (2.22), lo que permite definir la ganancia en potencia de una modulación frente a otra de manera muy sencilla cuando  $P_E \rightarrow 0$  (2.23).

$$Si P_E \rightarrow 0 \Rightarrow \gamma_b = \gamma_{b,inf} \quad (2.22)$$

$$Si P_E \rightarrow 0 \Rightarrow g_{1,2} = \frac{\beta_1 N_1}{\beta_2 N_2} \quad (2.23)$$

Por otro lado, también se puede calcular, en base al Criterio de Nyquist [14] la eficiencia espectral máxima para una modulación libre de ISI en canal gaussiano (2.24), donde  $L$  representa el número de dimensiones de la modulación.

$$\nu_{max} = \frac{R}{W_{min}} = \frac{R}{\frac{L}{2T}} = 2 \frac{N}{L} \quad (2.24)$$

Además, la Capacidad de Canal de Shanon [13] establece un límite para el régimen binario en función de la relación señal a ruido,  $\frac{S}{N}$ :

$$R \leq W \log_2 \left( 1 + \frac{S}{N} \right) \quad (2.25)$$

Dicho límite, desarrollado utilizando (2.26), donde  $T_b$  es el período de bit, se traduce en un límite para las eficiencias  $\rho$  y  $\nu$  (2.27).

$$\frac{S}{N} = \frac{\frac{\mathcal{E}_b}{T_b}}{N_0 W} = \gamma_b \nu \quad (2.26)$$



$$\rho \leq \frac{\nu}{2^\nu - 1} \quad (2.27)$$

Por último, a partir de la probabilidad de error de símbolo  $P_E$ , se puede calcular la probabilidad de error de bit,  $P_b$ . Resulta trivial establecer una sencilla relación a modo de cotas (2.28) entre ambos valores. Por otro lado, también es lógico el que si los símbolos adyacentes entre sí se diferencia en un solo bit, la probabilidad de error de bit disminuirá. Esto es conocido como Codificación Gray. Un ejemplo se puede encontrar en la Figura 2.2. Al emplear la Codificación Gray, la probabilidad de error de bit se asemeja a la de símbolo dividida entre el valor de  $N$  (2.29).

$$\frac{1}{N}P_E \leq P_b \leq P_E \quad (2.28)$$

$$\text{Con Codificación Gray } P_b \approx \frac{P_E}{N} \quad (2.29)$$

En relación a las tasas de error medidas empíricamente, estas son denotadas como Symbol Error Rate (SER) y Bit Error Rate (BER), respectivamente

## 2.2.2.- MODULACIONES M-PSK

Las modulaciones de fase [14] se caracterizan por transportar información únicamente en la fase de la señal transmitida. Esto se debe a que todos los posibles símbolos de la constelación poseen el mismo módulo respecto del origen de coordenadas. Siguiendo esta premisa, se pueden generar múltiples constelaciones que reciben el nombre de M-PSK, donde  $M$  es potencia de 2 e indica el número de símbolos de la modulación. La Figura 2.4 muestra una posible constelación para 2-PSK o BPSK (Binary Phase Shift Keying), y la Figura 2.5 para una 8-PSK. Se usa en ambos casos Codificación Gray. Se recuerda además que la modulación 4-PSK o QPSK fue ya presentada en la Figura 2.2 .

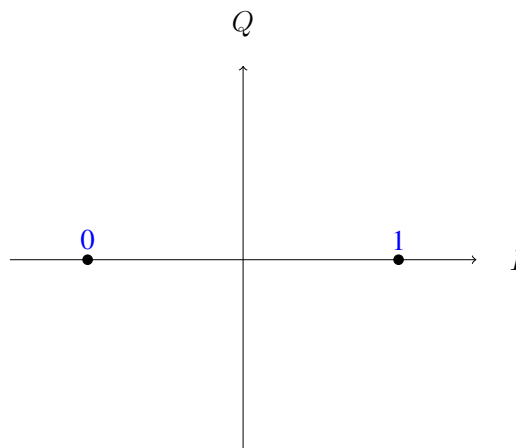


Figura 2.4.- Representación de una posible constelación para la modulación 2-PSK o BPSK.

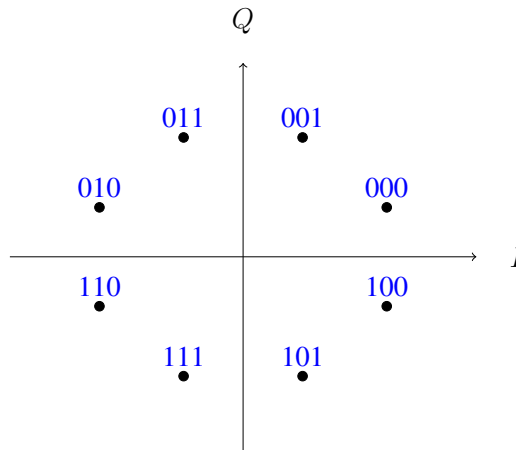


Figura 2.5.- Representación de una posible constelación para la modulación 8-PSK con Codificación Gray.

Matemáticamente, es posible expresar la envolvente compleja de la señal transmitida empleando una modulación M-PSK según (2.30). El módulo de todos los símbolos se denota por  $a$ , mientras que  $\varphi[m]$  es una secuencia de variables aleatorias independientes, que toma valores  $\{\varphi_i\}_{i=0,1,\dots,M-1}$ . En general,  $\varphi_i = \frac{2\pi}{M}i$ . Como se había introducido en el Apartado 2.1, la envolvente compleja se define respecto de la frecuencia central  $f_c$  y una fase aleatoria  $\theta$ , ambas fruto del Modulador de Canal. El receptor deberá incorporar un esquema recuperación de portadora, como se adelantó al final del Apartado 2.1. Por otro lado, se recuerda que para que la transmisión esté libre de ISI, la señal base  $\phi(t)$  debe cumplir lo expuesto en (2.6). Según todo lo expuesto, la envolvente compleja de la señal puede expresarse como:

$$\bar{s}(t) = \sum_{m=-\infty}^{m=\infty} a\phi(t - mT)e^{j\varphi[m]} \quad (2.30)$$

Para una modulación M-PSK con  $N = \log_2(M)$  bits por símbolo puede calcular fácilmente la ganancia geométrica (Apartado 2.2.1):

$$\beta_{M-PSK} = 4 \sin^2\left(\frac{\pi}{M}\right) \quad (2.31)$$

A partir de la cual se pueden obtener las cotas de la SER (2.21), y una estimación de la BER usando Codificación Gray (2.29):

$$Q\left(\sqrt{2\gamma_b N} \sin\left(\frac{\pi}{M}\right)\right) \leq P_{E,M-PSK} \leq 2Q\left(\sqrt{2\gamma_b N} \sin\left(\frac{\pi}{M}\right)\right) \quad (2.32)$$

Con Codificación Gray  $P_b \approx \frac{P_E}{N}$

Asumiendo la condición (2.22), se puede calcular también la eficiencia en potencia (2.18):

$$\rho_{M-PSK} \approx \frac{2 \sin^2\left(\frac{\pi}{M}\right)N}{[Q^{-1}(P_E)]^2} \quad (2.33)$$





Además, a partir de (2.24), es directo obtener la eficiencia espectral máxima:

$$\nu_{max,M-PSK} = N \left[ \frac{\text{bits}}{\text{s Hz}} \right] \quad (2.34)$$

Por último, se conoce una modulación M-PSK codificada diferencialmente [14], como aquella en la que se transmite la fase absoluta, pero con la información contenida en la diferencia de fase,  $\Delta\varphi[m]$ :

$$\varphi[m] = \varphi[m-1] + \Delta\varphi[m] \quad (2.35)$$

Esto permite eliminar la ambigüedad de fase generada en los esquemas de recuperación de portadora convencionales. M-PSK codificada diferencialmente presenta unos parámetros idénticos a M-PSK, salvo por la probabilidad de error de símbolo, si bien la energía normalizada de bit es similar en ambos casos:

$$P_{E,M-PSK,dif} \approx 2P_{E,M-PSK} \quad \gamma_{b,M-PSK,dif} \approx \gamma_{b,M-PSK} \quad (2.36)$$

También es posible emplear la codificación diferencial para eliminar el esquema de recuperación de portadora en el receptor. Esto se conoce como M-PSK diferencial. No obstante, aunque disminuye su complejidad, empeora las prestaciones del sistema respecto a una M-PSK convencional, por lo que no se hará uso de esta posibilidad durante este TFM.

### 2.2.3.- MODULACIONES M-QAM

Las modulaciones de amplitud en cuadratura [13] se caracterizan por transportan información tanto en la fase como en la amplitud de la señal transmitida. En dichas modulaciones los símbolos se distribuyen siguiendo un patrón de cuadrados cuyo centro coincide con el origen de coordenadas. Siguiendo esta premisa, se pueden generar múltiples constelaciones que reciben el nombre de M-QAM, donde  $M$  igual a  $2^N$ , siendo  $N$  un número par<sup>5</sup> y mayor que cero. En el caso de 4-QAM, la constelación es idéntica a una QPSK, por lo que su representación se puede encontrar en la Figura 2.2. La Figura 2.6 muestra un ejemplo de la constelación 16-QAM empleando Codificación Gray.

Matemáticamente, es posible expresar la envolvente compleja de la señal transmitida empleando una modulación M-QAM según (2.2)<sup>6</sup>. La secuencia  $A[m]$  se define en (2.1). Cada una de sus componentes es una secuencia de variables aleatorias independientes  $\{a_i\}_{i=0,1,\dots,M-1}$ , siendo  $a_i$  unas de las dos coordenadas que toman los posibles símbolos. Las coordenadas de una M-QAM se establecen con el objetivo de separar los símbolos lo máximo posible empleando la mínima energía [13]. La envolvente compleja se define respecto

<sup>5</sup>Esto en realidad no es necesario para las M-QAM generales, pero sí para las M-QAM cuadradas, que son a las que se limita este TFM.

<sup>6</sup>Nótese que en el Apartado 2.1 la modulación QPSK se definió matemáticamente con una notación más propia de QAM por simplicidad.

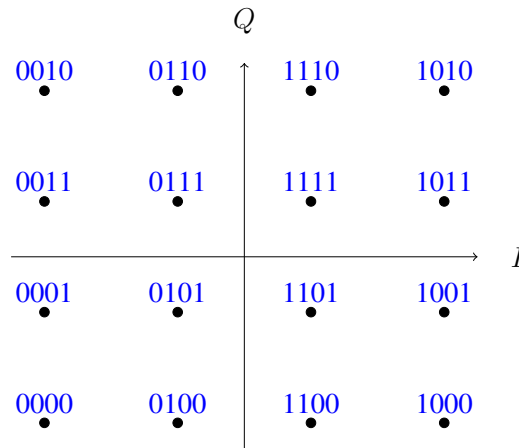


Figura 2.6.- Representación de una posible constelación para la modulación 16-QAM con Codificación Gray.

de la frecuencia central  $f_c$  y una fase aleatoria  $\theta$ , ambas fruto del Modulador de Canal. El receptor deberá incorporar un esquema recuperación de fase, al igual que en el caso de una M-PSK. Por otro lado, se recuerda que para que la transmisión esté libre de ISI la señal base  $\phi(t)$  debe cumplir lo expuesto en (2.6).

Para una modulación M-QAM con  $N = \log_2(M)$  bits por símbolo, siendo  $N$  un número par, puede calcular fácilmente la ganancia geométrica (Apartado 2.2.1):

$$\beta_{M-QAM} = \frac{6}{M-1} \quad (2.37)$$

A partir de la cual se pueden obtener las cotas de la SER (2.21), y una estimación de la BER usando Codificación Gray (2.29):

$$P_{E,M-QAM} = 1 - \left[ 1 - \frac{2\sqrt{M}-2}{\sqrt{M}} Q \left( \sqrt{\frac{3\gamma_b N}{M-1}} \right) \right]^2 \quad \text{Con Codificación Gray } P_b \approx \frac{P_E}{N} \quad (2.38)$$

Asumiendo la condición (2.22), se puede calcular también la eficiencia en potencia (2.18):

$$\rho_{M-QAM} \approx \frac{3N}{(M-1)[Q^{-1}(P_E)]^2} \quad (2.39)$$

Por último, a partir de (2.24), es directo obtener la eficiencia espectral máxima:

$$\nu_{max,M-QAM} = N \quad (2.40)$$

## 2.2.4.- COMPARATIVA M-PSK Y M-QAM

A la hora de comparar las modulaciones M-PSK y M-QAM, así como otras dos modulaciones cualquiera, es interesante empezar por las eficiencias. A partir de (2.33) y (2.39) se



puede concluir que la eficiencia de una modulación  $\rho_{\sqrt{M}\text{-PSK}}$  y la de  $\rho_{M\text{-QAM}}$  decrecen de igual forma, con un comportamiento asintótico proporcional a  $\frac{N}{M}$ . Además, (2.34) y (2.40) demuestran que la máxima eficiencia espectral es igual en ambos casos.

De esta manera, y centrándose únicamente en las eficiencias conseguidas, se podría concluir que los esquemas QAM son mejores que los PSK. Así, para un  $M > 4$  dado, la eficiencia espectral en ambos esquemas es la misma, pero la eficiencia en potencia de M-PSK será inferior a la de M-QAM. No obstante, las modulaciones M-PSK presentan una serie de ventajas vinculadas principalmente a la transmisión de información basándose únicamente en la fase de la señal, y no la amplitud.

Entre dichas ventajas, la más destacable es la robustez frente a desvanecimientos en potencia. Desde el punto de vista de la constelación, estos se interpretarían como una disminución del radio de la circunferencia que conforman los símbolos, pero no cambiarían las regiones de decisión de cada símbolo, lo que sí sucede para cualquier M-QAM que no sea 4-QAM. Además, las modulaciones M-PSK ofrecen mayor robustez frente a la distorsión no lineal de amplificadores cuando se emplea como función base  $\phi(t)$  un pulso rectangular, si bien dicha condición no se dará en este TFM.

## 2.3.- TIPOS DE CANALES Y PERTURBACIONES EN UN SISTEMA DE COMUNICACIONES DIGITALES

En este apartado se analizan los tipos de canales y perturbaciones más comunes en un sistema de comunicaciones digitales en general, y un sistema móvil en particular. Se expondrán las distintas problemáticas desde el punto de vista teórico, reservándose las posibles soluciones para capítulos posteriores.

### 2.3.1.- TIPOS DE CANALES

Se denomina canal al entorno físico, bien sea radio o guiado, a través del que se propaga la señal transmitida hasta su recepción. Desde el punto de vista matemático, es posible modelar con mayor o menor precisión los efectos que el canal produce en la señal. Así, se define un canal por su respuesta al impulso  $\bar{h}_c(t)$  [14], usando la notación de la envolvente compleja. Es importante señalar que la descripción de un canal mediante su respuesta al impulso es solo válida para canales lineales, descritos en este apartado. No obstante, aunque no todos los canales se comporten de esa manera, es muy habitual suponer que los canales radio sí lo sean, o al menos, que se pueden modelar de manera precisa como tal.

A la respuesta al impulso es necesario añadir el ruido de Johnson-Nyquist, que se puede considerar como parte del canal y es inherente a todas las comunicaciones. Como resultado,



la señal recibida se puede expresar matemáticamente como (2.41)<sup>7</sup>

$$\bar{r}(t) = \bar{s}(t) * \frac{1}{2} \bar{h}_c(t) + \bar{n}(t) \quad (2.41)$$

En función de su respuesta al impulso, un canal se puede clasificar como:

- Canal gaussiano: El canal más sencillo matemáticamente, pero generalmente poco realista. Su respuesta al impulso es una delta de Dirac [14], lo que resulta en que la señal recibida es idéntica a la transmitida salvo por el ruido<sup>8</sup>:

$$\bar{r}(t) = \bar{s}(t) * \delta(t) + \bar{n}(t) = \bar{s}(t) + \bar{n}(t) \quad (2.42)$$

- Canal lineal invariante en el tiempo: Aunque incluye otro tipo de canales, en este TFM se utilizará este término para referirse a los canales multitrayecto. Estos se puede expresar como una combinación lineal de distintas deltas de Dirac ponderadas:

$$\bar{h}_c(t) = \sum_{n=1}^N a_n e^{j\phi_n} \delta(t - t_n), \quad (2.43)$$

donde tanto los valores de atenuación,  $a_n$ , desfase,  $e^{j\phi_n}$ , y retardo,  $t_n$ , permanecen constantes en el tiempo.

- Canal lineal variante en el tiempo: Un canal variante en el tiempo es aquel que no puede ser definido mediante una única respuesta al impulso  $\bar{h}_c$ , sino que esta varía según el momento en el que se calcule. Así, un canal variante puede ser descrito mediante un conjunto de canales lineales e invariantes. En este tipo de canales, es frecuente definir un tiempo de coherencia,  $T_C$ , para referirse al periodo durante el cual el canal permanece aproximadamente constante. En líneas generales, se considera que la variabilidad temporal presenta un problema serio para el sistema cuando el tiempo de coherencia es menor al periodo de símbolo  $T_c < T$ . El efecto resultante es comúnmente conocido como desvanecimiento temporal o selectividad en el tiempo [13].

Merecen una mención especial, aunque fuera de la clasificación anterior, los canales selectivos en frecuencia. Se denomina canal selectivo en frecuencia a aquel en el que, para un instante temporal fijo, la función de transferencia (la Transformada de Fourier (TF) de la respuesta al impulso del sistema  $TF \{ \bar{h}_c(t) \} = H_c(f)$ ) no es constante en un determinado ancho de banda. Es decir, en un determinado instante, la atenuación que sufren cada una de las componentes frecuenciales de la señal no es la misma. Un canal lineal e invariante puede ser selectivo en frecuencia, como también lo puede ser un canal variante. En estos casos se define el ancho de banda de coherencia  $B_c$ , como

<sup>7</sup>El factor  $\frac{1}{2}$  se debe al uso de la notación en envolvente compleja [14].

<sup>8</sup>Se ha obviado el retardo que sufrirá la señal fruto del tiempo de propagación por el canal.



el ancho de banda en el que la respuesta al impulso sí se mantiene constante. En líneas generales, se considera que la variación frecuencial presenta un problema serio para el sistema cuando el ancho de banda de coherencia es menor que el ancho de banda del sistema  $B_c < W$ . El efecto resultante es comúnmente conocido como desvanecimiento frecuencial o selectividad en frecuencia [13].

De nuevo, es importante destacar que una clasificación excluyente definiría solo tres tipos de canales: canal gaussiano, canal lineal e invariante y canal lineal variante en el tiempo. No obstante, se ha tratado la característica de selectividad frecuencia como un tipo aparte debido a su importancia en comunicaciones radio.

## 2.3.2.- PERTURBACIONES

Algunas de las perturbaciones más comunes en un sistema de comunicaciones vía radio y móviles son:

- **Atenuación:** Cuando la atenuación que sufre la señal es constante, o presenta una escasa variación, no suele representar un problema para la comunicación siempre y cuando esta se diseñe con la potencia adecuada. No obstante, en un entorno móvil la atenuación suele variar con el tiempo. Por ejemplo, cuando el receptor se aleja del transmisor esta por lo general aumenta.
- **Ruido:** Ya introducido desde el Apartado 2.1, el ruido térmico o ruido de Ruido de Johnson-Nyquist se debe a la agitación de los portadores de carga y es inherente a todo sistema de comunicaciones. Dicho ruido se puede modelar como una secuencia de variables aleatorias gaussiana de media cero y varianza  $\sigma^2$  [13]. Para un canal gaussiano, la secuencia de símbolos ya muestreados,  $R[m]$ , se puede expresar como (2.5), siendo  $n[m]$ :

$$n[m] = n_I[m] + jn_Q[m] \quad (2.44)$$

Teniendo cada una de las muestras  $n[m]$  una varianza  $\sigma^2$  que viene dada por:

$$\sigma^2 = 2N_0, \quad (2.45)$$

donde  $2N_0$  es el nivel de la densidad espectral de potencia (W/Hz) de la envolvente compleja del ruido en el ancho de banda de la señal. Desde el punto de vista unidimensional, la varianza de cada una de las componentes, real y compleja, es la mitad de la varianza total, esto es,  $N_0$ . La expresión resultante es entonces (2.46). Las componentes del ruido muestreado son independientes entre sí y dos a dos [13].

$$f_{n_{I(n)}} = f_{n_{Q(n)}} = \frac{1}{\sqrt{2\pi N_0}} e^{\frac{-n}{2N_0}} \quad (2.46)$$



- **Multicamino:** El multicamino es una de las perturbaciones principales en la mayoría de los canales radio. Este se debe a la llegada al receptor de la señal transmitida por más de un camino distinto, fruto de fenómenos como la reflexión y refracción. Dichas componentes alcanzan el receptor con retardos y atenuaciones distintas según el camino recorrido. Este fenómeno se suele describir como dispersión temporal<sup>9</sup>. El multicamino se puede expresar como un canal lineal según (2.43). Sin embargo, su problema principal radica en que según el retardo y la frecuencia, la suma de las distintas componentes puede ser constructiva (las señales se suman en fase) o destructiva (las señales se suman en contrafase). El multicamino puede dar lugar por tanto a un canal selectivo en frecuencia.

Además, en un canal móvil el multicamino variará según el instante, modificándose los retardos, atenuaciones y número de componentes en la expresión (2.43). Esto dará lugar a un canal variante en el tiempo. El resultado se suele denominar como fast-fading, o pérdidas por desvanecimientos rápidos.

- **Efecto Doppler:** Probablemente el efecto más característico de los canales móviles. El Efecto Doppler es un conocido efecto físico por el cual el receptor experimenta un cambio aparente de la frecuencia de una señal consecuencia del movimiento relativo del transmisor [13]. La consecuencia es un desplazamiento positivo o negativo en frecuencia conocido comúnmente como dispersión frecuencial<sup>10</sup>. Dicho desplazamiento se calcula según (2.47), donde  $c$  denota la velocidad de la luz y  $v$  la velocidad relativa entre emisor y receptor (positiva si se están acercando). El ángulo entre la dirección de llegada y la dirección de movimiento se denota por  $\alpha$ .

$$\Delta f = \frac{v}{c} f \cos \alpha \quad (2.47)$$

Por su naturaleza, el Efecto Doppler aparece en canales variantes en el tiempo, siendo el caso más complejo de analizar cuando la velocidad es variable.

- **Fallos de sincronismo en el reloj:** Aunque no se trata de una perturbación del canal propiamente dicho, se presenta aquí como una posible fuente de fallo en las comunicaciones. Como su nombre indica, se debe a la desincronización entre el reloj del transmisor y el receptor. Los relojes digitales encargados de generar el período de símbolo pueden presentar ligeras variaciones que afecten a la sincronización de la comunicación. El caso en el que el reloj del receptor genera un período distinto al del transmisor suele ser el más grave. Otro caso alternativo es que ambos relojes sean idénticos pero presenten ligeras oscilaciones. La realidad suele ser una combinación de ambas: los dos relojes

<sup>9</sup>No confundir con varianza en el tiempo.

<sup>10</sup>No confundir con selectividad en frecuencia.



son distintos y presentan a su vez variaciones. Por último, incluso cuando los relojes sean idénticos y tengan el mismo período, estos pueden actuar en distintos instantes temporales. El resultado matemático es que (2.5) se transforma en (2.48), generando ISI.  $T'$  denota el período de muestreo del transmisor,  $T''$  el del receptor, y  $\delta$  el offset temporal entre el funcionamiento de ambos.

$$R[m] = \left( \sum_{m=-\infty}^{m=\infty} A[m]\phi(t - mT') + \bar{n}(t) \right) * \phi^*(-t) \Big|_{t=mT''+\delta} \quad (2.48)$$

- Pérdidas por desvanecimientos lentos (Shadowing): Se conocen así a las pérdidas adicionales producidas por la existencia de elementos de gran escala (como una montaña) en el canal. No obstante, esta perturbación no presenta una variación rápida con el tiempo, ni siquiera en un canal móvil convencional, y se corrige inyectando potencia adicional a la señal transmitida. Esto se debe a que puede ser modelada por una sencilla componente de pérdidas adicionales de distribución logarítmico-normal [13]. Desde el punto de vista de este trabajo será tratada como una atenuación variable.

La combinación de todas las perturbaciones aquí descritas dará lugar a desvanecimientos en tiempo y frecuencia desde el punto de vista de la potencia recibida. Además, se producirá ISI, ya que la condición (2.6) solo aseguraba su no existencia en canales gaussianos. Como consecuencia, todo sistema de comunicaciones móvil, en el que aparecerán en mayor o menor medida estos fenómenos, debe incorporar mecanismos de protección frente a ellos.

A modo de referencia, se presenta la respuesta de un canal variante en el tiempo y con dispersión frecuencial en un instante  $t$  a un impulso transmitido  $\tau$  segundos antes:

$$h_c(t, \tau) = \sum_{i=1}^{N(t)} a_i(t)\delta(\tau - \tau_i(t))e^{-j\omega\tau_i(t)}e^{-j\nu_i(t,\omega)t} \quad (2.49)$$

, donde  $N(t)$  indica el número de componentes que conforman la respuesta al impulso,  $a_i(t)$  denota una atenuación compleja,  $\tau_i(t)$  el retardo de cada componente y  $\nu_i(t, \omega)$  su desplazamiento en pulsación por efecto Doppler.

Por último, es necesaria hacer una aclaración respecto de la detección de símbolos en un canal interferente. Si bien es común que la decisión se realice en bloques de varios bits con algoritmos destinados a disminuir la probabilidad de error, durante el presente trabajo se decidirá siempre símbolo a símbolo de acuerdo al criterio de Máxima Verosimilitud (ML) [13]. No obstante, el diseño de los sistemas aquí presentados es tal que, para implementar algún algoritmo de detección de secuencias, no sería necesario modificar la estructura general de bloques en absoluto. Por otro lado, según la codificación de corrección de errores



empleada, la detección de secuencias es requerida obligatoriamente para un funcionamiento óptimo. Más información acerca de estos conceptos puede encontrarse en [13].

## 2.4.- IGUALACIÓN LINEAL DE CANAL

En este apartado se introducirán los conceptos básicos de la igualación lineal de canal como mecanismo de corrección frente a algunas de las perturbaciones presentadas en el Apartado 2.3.2

Sea  $p(t)$  el resultado de la convolución entre la señal base  $\phi(t)$ , la respuesta al impulso del Canal,  $h_c(t)$ , y del del filtro adaptado  $\phi^*(-t)$ :

$$p(t) = \phi(t) * h_c(t) * \phi^*(-t) \quad (2.50)$$

Este se denota como Canal Continuo Equivalente (CCE) [14], del que se puede obtener a su vez el Canal Discreto Equivalente (CDE),  $p[m]$ , según:

$$p[m] = p(t)|_{t=mT}, \quad (2.51)$$

a partir del cual se puede expresar la secuencia de números complejos tras el muestreo en el receptor,  $R[m]$ , como la convolución discreta entre la secuencia generada por el transmisor,  $A[m]$ , y el CDE o  $p[m]$ , más las componentes de ruido muestreadas:

$$R[m] = \sum_{j=-\infty}^{j=\infty} A[j]p[m-j] + n[m] \quad (2.52)$$

Nótese que, en comparación con (2.5), en este caso no se ha supuesto un canal gaussiano. Es posible demostrar [14] que desde el punto de vista del CDE se puede asegurar la no existencia de ISI con la condición:

$$p[m] = K_s \delta[m], \quad (2.53)$$

donde  $K_s$  representa una constante de amplitud. Hay que tener en cuenta que (2.6) daba por supuesto un canal gaussiano, con lo que deja ser condición suficiente para la no existencia de ISI en el momento en el que esto no se cumple, por ejemplo, para un canal lineal invariante en el tiempo (2.43). Para tratar de mitigar el efecto de la ISI, es posible añadir un filtro digital denotado como igualador a la salida del Demodulador de Señal dentro del Demodulador Digital (Figura 2.1). De esta manera, la secuencia de números complejos que recibe el Decisor pasa a ser:

$$R[m] = (A[m] * p[m] + n[m]) * c[m], \quad (2.54)$$





donde  $n[m]$  representa la secuencia de muestras de ruido (2.44), y  $c[m]$ , los coeficientes de un filtro digital de respuesta al impulso finita, o filtro Finite Impulse Response (FIR). Si bien existen múltiples técnicas de igualación que se clasifican atendiendo a distintos criterios [13], esta explicación se centrará únicamente en aquellas empleadas en el presente TFM. Por tanto, se considerará siempre una igualación lineal y dada por un filtro FIR como la presentada en (2.54). Esta clase de igualadores se denotan como Linear Transversal Equalizer (LTE). Por otro lado, la igualación será siempre ciega, esto es, no se conoce a priori la respuesta al impulso del CDE. En el caso de la igualación no ciega, una estimación de  $p[m]$  suele ser calculada por medio de la transmisión de una secuencia de entrenamiento. Además, la igualación será adaptativa, es decir, los coeficientes del igualador se modificarán en cada iteración para adaptarse a posibles cambios del canal en el tiempo. Sea  $x[m]$  la suma de la convolución discreta entre  $A[m]$  y  $p[m]$  más las componentes de ruido  $n[m]$ :

$$R[m] = \sum_{j=0}^{j=N} c[j]x[m-j] \quad (2.55)$$

Considerando ahora  $c_m$  como el valor que toman el conjunto de coeficientes del filtro en el momento en el que se calcula la muestra  $R[m]$ , estos se pueden expresar como un vector columna:

$$c_m = \{c_0, c_1, \dots, c_N\}^T \quad (2.56)$$

Y sea  $x_m$  el conjunto de muestras de  $x[m]$  ponderadas por el filtro  $c[m]$  en la muestra  $m$ , de acuerdo con (2.55):

$$x_m = \{x_m, x_{m-1}, \dots, x_{m-N}\}^T, \quad (2.57)$$

esto permite expresar (2.55) como un producto vectorial:

$$R[m] = c_m^T x_m \quad (2.58)$$

Como la igualación es adaptativa, los coeficientes del filtro se deben adaptar periódicamente, en este caso, muestra a muestra, lo que resulta en la expresión:

$$c_{m+1} = c_m + \mu \cdot x_m^* \cdot e[m], \quad (2.59)$$

donde  $\mu$  denota una constante conocida como ganancia de actualización,  $x_m^*$  las muestras conjugadas de  $x_m$  (2.57) y  $e[m]$  el error calculado en la muestra  $m$  en base a una función. Este proceso de actualización conduce a la minimización del error cuadrático medio o criterio Minimum Mean Squared Error (MMSE) [13]. La función de cálculo de error determina el comportamiento del igualador, o dicho de otra forma, el criterio que este sigue para tratar de corregir el efecto de la ISI. Un desarrollo detallado de una amplia gama de igualadores



se puede encontrar en [13]. En este TFM se considerarán únicamente dos: el criterio Least Mean Square (LMS) y el criterio Constant Modulus Algorithm (CMA), ambos igualadores LTE ciegos y adaptativos de acuerdo con lo expuesto hasta ahora.

Para el criterio LMS el error se calcula como la diferencia entre la muestra transmitida,  $A[m]$ , y la recibida,  $R[m]$ . Sin embargo, nótese que el receptor no conoce la muestra transmitida  $A[m]$ , a no ser que se utilice una secuencia de entrenamiento, lo que no se considera en este caso. En cambio, se considera la salida del Decisor  $\tilde{A}[m]$  como la muestra correcta, con lo que el cálculo del error se realiza como:

$$e_{LMS}[m] = \tilde{A}[m] - x[m] \quad (2.60)$$

Nótese que la convergencia del igualador adaptativo LMS en unos coeficientes óptimos depende en gran medida de un correcto arranque del proceso, o dicho de otra forma, que los primeros  $\tilde{A}[m]$  sean una estimación fiable de  $A[m]$  para que la actualización de los coeficientes sea calculada de manera correcta.

Por su parte, el criterio CMA busca que el módulo de las muestras recibidas sea constante, para lo que calcula el error de la forma:

$$e_{CMA}[m] = (R_2 - R[m]^2) \cdot R[m], \quad (2.61)$$

donde  $R_2$  representa un cociente entre la esperanza del módulo de  $A[m]$  elevado a la cuarta y elevado al cuadrado:

$$R_2 = \frac{E[|A[m]|^4]}{E[|A[m]|^2]^2} \quad (2.62)$$

Nótese que en este caso el receptor sí conoce la esperanza matemática de  $A[m]$  siempre y cuando conozca la probabilidad a priori de los símbolos transmitidos. Por otro lado, y como se puede intuir, el algoritmo CMA está orientado a corregir los efectos de la ISI para modulaciones de amplitud constante. Tal es el caso de las modulaciones M-PSK (Apartado 2.2.2) o la 4-QAM, al corresponderse esta con QPSK (Apartado 2.2.3). Sin embargo, su actuación puede ser favorable en la corrección de la ISI incluso en el caso de modulaciones que no cumplan dicha condición.

Por último, y dada su utilización en este TFM, resulta adecuado introducir el concepto de igualador espaciado fraccionalmente del inglés Fractional Spaced Equalizer (FSE) [13]. Todo explicado hasta ahora supone, si bien un canal de respuesta al impulso desconocida, un muestreo por parte del receptor en los instantes óptimos  $t = mT$ . No obstante, esto no es una suposición realista, ya que el receptor desconoce el retardo introducido por el canal, que además puede ser variable, ni ha de estar necesariamente sincronizado con el transmisor. Este fenómeno tiene un efecto análogo a la perturbación introducida en el Apartado 2.3.2 como fallos de sincronismo en el reloj, este es, que (2.5) se transforme en (2.48), pudiéndose



producir nulos en el espectro de  $R[m]$  que afectarán gravemente al igualador al tratar de compensarlos.

La estrategia FSE consiste en el muestreo del receptor en un submúltiplo del período de símbolo de la manera:

$$R[k] = \bar{r}(t) |_{t=mT}, \quad k = \frac{T}{n}, \quad \text{Siendo } n \text{ un número entero mayor que } 1, \quad (2.63)$$

con lo que se obtienen  $n$  muestras por símbolo que serán ponderadas por el igualador en su acción, que generará a su salida la secuencia  $R[m]$  con una única muestra por símbolo para ser transformada por el Decisor en  $\tilde{A}[m]$ .



## 3. HERRAMIENTAS EMPLEADAS

El objetivo de este capítulo es el de introducir las herramientas utilizadas en el desarrollo del sistema en el que se basa este TFM. Estas son fundamentalmente dos: GNU Radio, como herramienta de software empleada para diseñar e implementar la parte digital; y los dispositivos USRP como la parte de hardware, los cuales funcionan de manera compatible con GNU Radio y constituyen el back-end y front-end en el transmisor y receptor respectivamente.

### 3.1.- GNU RADIO

En este apartado se presenta la herramienta GNU Radio, introduciendo sus fundamentos y funcionalidades básicos. También se presentará su herramienta gráfica de desarrollo GNU Radio Companion (GRC). Como nota importante, cabe destacar que el desarrollo de este TFM se ha llevado a cabo empleando la versión más reciente, que se corresponde con la 3.9 tanto para GNU Radio como para GRC. El sistema operativo empleado ha sido Ubuntu 20.04 Long Term Support (LTS), conocido también bajo el nombre Focal Fossa [15].

#### 3.1.1.- INTRODUCCIÓN

GNU Radio [1] es una herramienta de desarrollo de software libre y código abierto que proporciona bloques de procesamiento de señal digital para la implementación de sistemas de comunicaciones. Un sistema generado de esta manera puede ser conectado a un equipo de hardware de RF para crear un transmisor o receptor reales, o sencillamente empleado dentro del entorno software a modo de simulación.

En su núcleo, GNU Radio se puede entender como un entorno desarrollado en el lenguaje de programación C++ que define un amplio conjunto de bloques de procesamiento de señal cada uno de los cuales realiza una tarea concreta. Lo que es más importante, GNU Radio define también la interacción entre dichos bloques y, como herramienta de código abierto, añade la posibilidad de que cualquier usuario programe los suyos propios o modifique los ya existentes. El código fuente actualizado se puede encontrar en el servicio online de alojamiento de repositorios GitHub [15], y su manual y página de referencia se recogen en [16]. Además, GNU Radio cuenta con una interfaz gráfica llamada GRC, la cual permite el manejo y conexión de bloques de manera más amigable para el usuario. Aunque útil, dicha interfaz está limitada en su funcionalidad, pero combinada con el manejo del código permite agilizar la configuración de muchos bloques y parámetros.

Por otro lado, GNU Radio también hace uso del lenguaje de programación Python, basando en él gran parte de las funcionalidades de usuario. Por ejemplo, los esquemas diseñados por medio de GRC generan un ejecutable en Python, encargado de controlar la capa



más alta del programa basándose en la invocación de los bloques que lo componen. Si bien existe la opción de programar bloques propios en Python, encargándose GNU Radio de su posterior compilación a C++, no se empleará dicha característica en la realización de este TFM, ya que, si bien pueden permitir la simplificación de algunas definiciones, pueden resultar en código no óptimo. Se seguirá siempre la política de programar cualquier bloque nuevo o modificación en C++, utilizando GRC y, por ende, Python para la definición de las conexiones y variables de los archivos ejecutables. Esta estrategia se corresponde con la seguida y recomendada por los desarrolladores de GNU Radio.

Como conclusión, las características fundamentales que han propiciado la elección de GNU Radio como herramienta de software son las siguientes: el ser un proyecto de software libre y código abierto; el empleo de un lenguaje de bajo nivel multiplataforma como C++, lo que brinda una gran configurabilidad y versatilidad en el procesado digital; la existencia de una amplia comunidad consecuencia de su creciente popularidad en los ámbitos de educación, investigación y la empresa; y su compatibilidad con una amplia gama de dispositivos de hardware de distintos fabricantes [17].

### **3.1.2.- BLOQUES EN GNU RADIO Y HERRAMIENTA GRC**

La palabra bloque se utiliza para denotar la estructura básica de GNU Radio. Un bloque es un subsistema independiente integrado dentro del entorno que funciona como fuente, sumidero o procesador de valores de distinto tipo. Todos los bloques derivan de una clase abstracta base en C++ que define la estructura y funcionalidades básicas a cumplir. Por ende, tanto los bloques por defecto como los desarrollados expresamente como parte de este TFM son clases hijas de la clase madre.

A partir de esta consideración, se pueden realizar distintas clasificaciones entre los bloques atendiendo a distintos criterios. Por ejemplo, partiendo de que un bloque básico recibe muestras en su entrada/s que procesa para generar muestras a su salida/s, se denominan bloques fuente a aquellos que generan información a su salida/s sin ninguna entrada; y bloques sumidero, a los que no cuentan con ninguna salida. Otra distinción importante, aplicada a las entradas y salidas, es la de síncronas y asíncronas. Las entradas y salidas síncronas demandan y generan muestras respectivamente en base a una tasa de muestreo fija. Por su parte, las entradas asíncronas no esperan la recepción de información de manera periódica, ya que las salidas asíncronas no generan dicha información periódicamente. Se entiende de esta definición la incompatibilidad en la conexión entre ambos casos. Es decir, una salida síncrona solo puede ir conectada a una entrada del mismo tipo y una salida asíncrona solo será admitida por una entrada igual. Esta incompatibilidad también se da entre entradas y salidas que manejen valores de distinto tipo, por ejemplo, entre una salida que genere valores de tipo int y una entrada que espere muestras de tipo float.

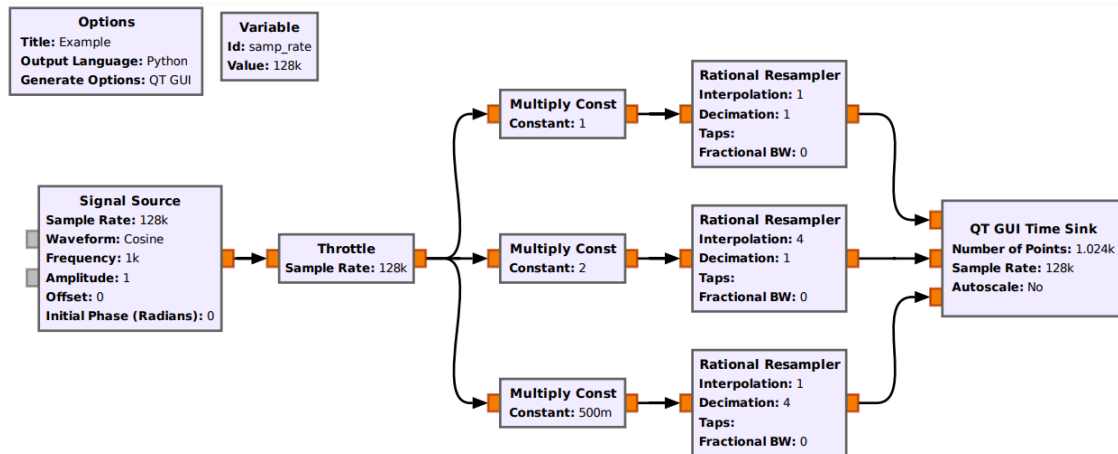


Figura 3.1.- Ejemplo de un esquema en GNU Radio de comparativa de una señal coseno diezmada e interpolada con la original.

Por otro lado, los bloques en GNU Radio se agrupan en módulos de acuerdo con su función, siendo algunos ejemplos los módulos gr-digital o gr-analog, los cuales se pueden encontrar en [16]. El primer paso para la programación de bloques, funciones o clases propios consiste en crear lo que se conoce como un out-of-tree-module [18]. Tras su programación, dicho módulo puede ser integrado en una instancia cualquiera de GNU Radio siempre y cuando la versión de esta coincida con la del entorno de desarrollo. Junto con esta memoria se adjunta el out-of-tree-module gr-TFMv5 que contiene todo el código necesario para la ejecución de los sistemas realizados. Los pasos concretos para la instalación de este se pueden encontrar en [18].

La conexión de una serie de bloques para conformar un sistema en GNU Radio, así como la configuración de los parámetros generales del mismo, es, a menudo, una tarea engorrosa. La herramienta gráfica GRC permite generar un archivo de Python que defina dichas conexiones y variables de una manera intuitiva y rápida. Además, es posible configurar los atributos de cada bloque desde GRC por medio de una interfaz definida en un archivo YAML. En su ejecución, el programa resultante instanciará un objeto de la clase correspondiente por cada uno de los bloques y gestionará los flujos de información intercambiados entre ellos.

Como ejemplo de lo explicado hasta ahora se presenta la Figura 3.1. En este esquema diseñado en GRC se muestra una señal digital de tipo coseno generada por el bloque Signal Source que es introducida en tres instancias distintas del bloque Rational Resample con el objetivo de visualizar la señal original, interpolada y diezmada. Para hacer más clara la distinción se añade el bloque Multiply Const configurado con una constante distinta en cada caso. El bloque Throttle sirve para el control de la congestión en base a la tasa de muestreo, definida en la variable samp\_rate. La visualización por pantalla, mostrada en la Figura 3.2, corre a cargo del bloque QT GUI Time Sink.

Por último, cabe destacar que GRC incorpora una leyenda de colores para identificar

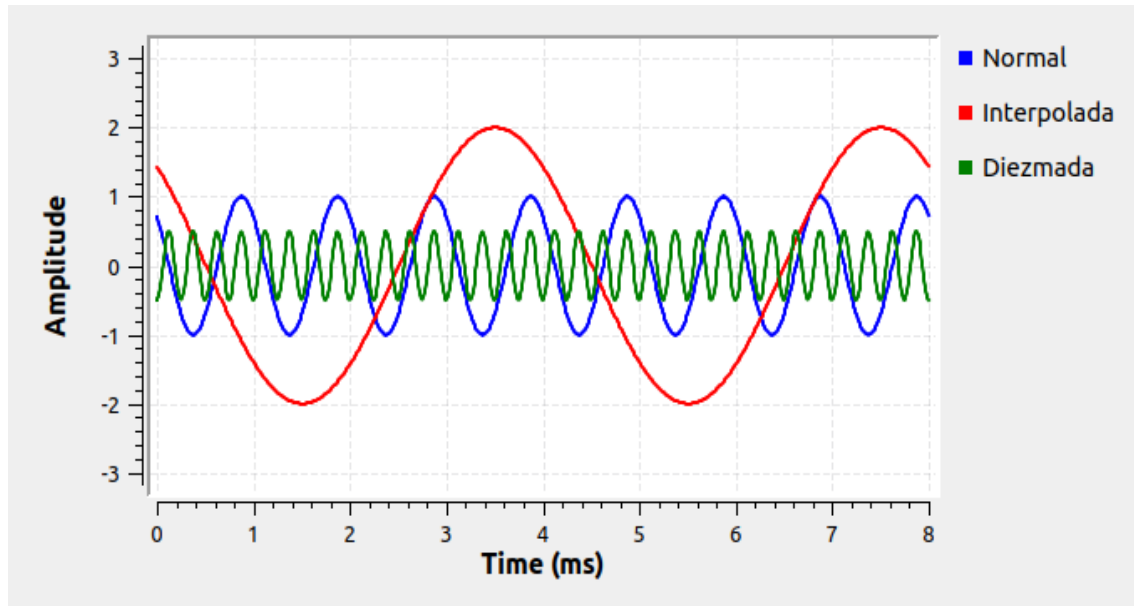


Figura 3.2.- Representación gráfica de la salida del esquema presentado en la Figura 3.1.

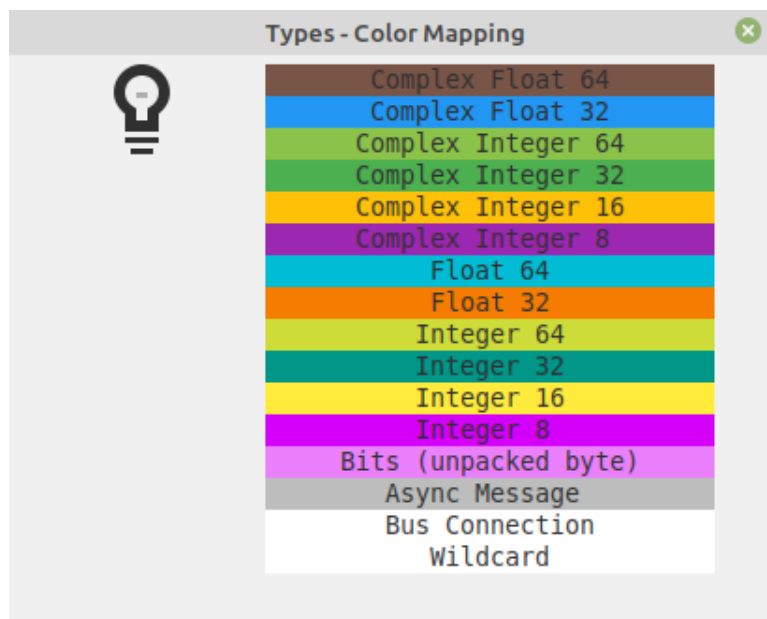


Figura 3.3.- Leyenda de tipos de valores en GRC.

los tipos de valor de cada entrada y salida, tal y como se muestra en la Figura 3.3. Nótese que las entradas y salidas asíncronas poseen un identificador propio, Async Message, lo que será tratado más en detalle a continuación, en el Apartado 3.1.3.

### 3.1.3.- COMUNICACIÓN ENTRE BLOQUES

Con el objetivo de que los distintos bloques que integran un único esquema puedan intercambiar información de control entre sí y modificar el comportamiento de otros o el suyo propio en base a esta, GNU Radio define dos modos de comunicación entre bloques:

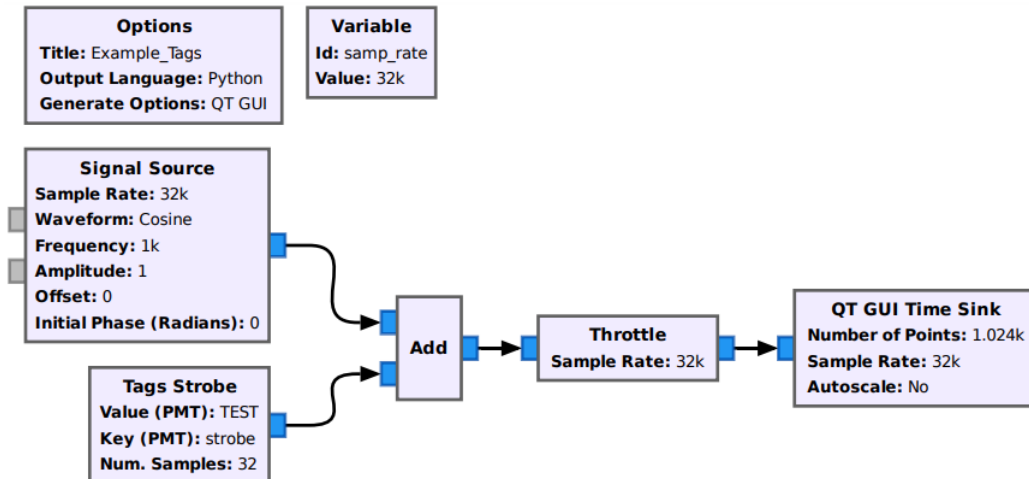


Figura 3.4.- Ejemplo de un esquema en GNU Radio en el que se hace uso de las etiquetas de flujo (stream tags).

las etiquetas de flujo o stream tags; y el envío de mensajes o message passing [16].

Las etiquetas de flujo son datos que se propagan a través de conexiones síncronas, funcionando de manera paralela al flujo de datos principal, este es, el intercambio de muestras de un determinado tipo y con una frecuencia fija entre bloques. Una etiqueta de flujo va siempre asociada a una muestra del flujo principal, siendo generada por un bloque encargado de dicha asociación. Los bloques aguas abajo del esquema pueden interpretar los datos contenidos en la etiqueta, modificarlos, suprimirlos, o simplemente propagarlos de forma transparente. Las etiqueta de flujo se definen mediante un par clave-valor ambos de tipo polimórfico o Polymorphic Type (PMT) [16].

En la Figura 3.4 se puede apreciar un ejemplo de funcionamiento de las etiquetas de flujo, donde el bloque Tags Strobe genera etiquetas de clave strobe y valor TEST cada 32 muestras. Como la tasa de muestreo definida en samp\_rate es de 32k muestras por segundo y la frecuencia del coseno generado por Signal Source es de 1 kHz, esto implica 32 muestras y una etiqueta por período. Como se puede observar, el bloque Add es el encargado de esta asociación, mientras que el bloque Throttle controla la congestión en base a la tasa de muestreo. El bloque QT GUI Time Sink permite la visualización de las muestras, en este caso complejas (véase la leyenda en la Figura 3.3) junto con las etiquetas. El resultado se muestra en la Figura 3.5.

Por otro lado, el envío de mensajes se presenta como la alternativa asíncrona de comunicación entre bloques. Este paradigma permite que bloques aguas abajo se comuniquen con bloques aguas arriba del esquema, facilitando además la comunicación del propio ejecutable de GNU Radio con una aplicación externa. Tanto las entradas como salidas que generan y reciben respectivamente estos mensajes se identifican con el tipo especial Async Message, incluido en la Figura 3.3. El contenido de los mismos, como el de las etiquetas de flujo, es de tipo polimórfico, y puede, aunque no necesariamente, responder también a la estructura



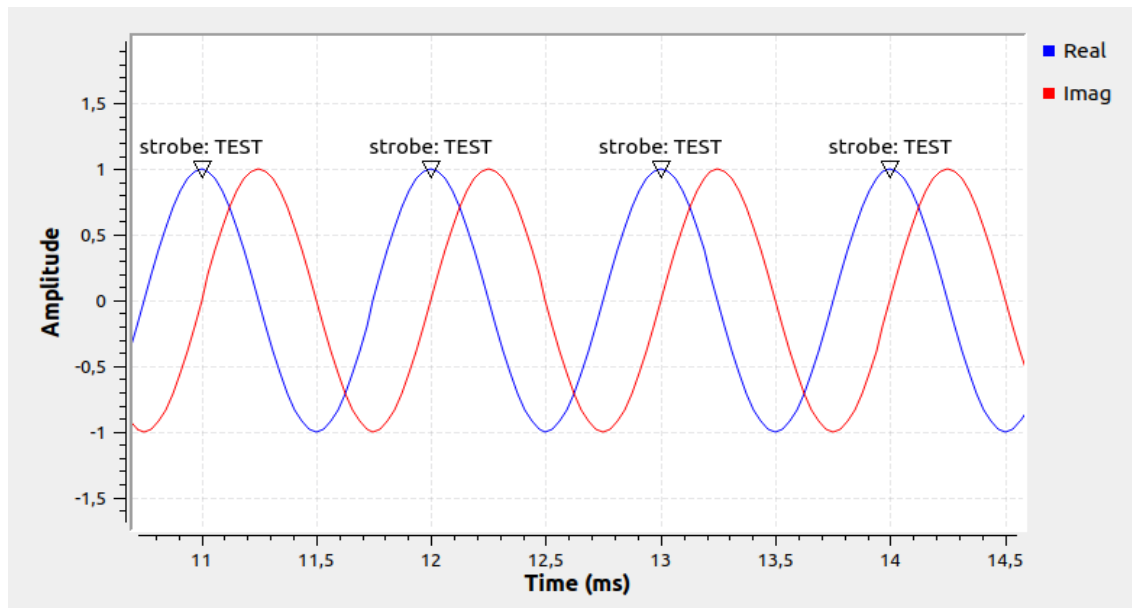


Figura 3.5.- Representación gráfica de la salida del esquema mostrado en la Figura 3.4

de par clave-valor. En concreto, GNU Radio define el Protocol Data Unit (PDU) como un tipo de PMT especial designado para el envío de mensajes, compuesto por el par metadatos-datos. Los metadatos son representados por un diccionario que a su vez está formado por un conjunto de claves y valores, y los datos, por un vector cuyos valores son de un tipo a determinar.

Nótese por último que el carácter asíncrono de este paradigma hace que los instantes temporales en los que un mensaje es generado o interpretado no sean deterministas. Esta es la diferencia más sustancial respecto de las etiquetas de flujo, las cuales son transmitidas, recibidas e interpretadas junto a una muestra del flujo principal. Por su parte, el tiempo de respuesta en el envío de mensajes dependerá del estado del programa general y los hilos y procesos en ejecución de la CPU del sistema de ejecución en cada momento. Si bien el tiempo entre que un mensaje es generado recibido e interpretado suele ser del orden de milisegundos, esta realidad ha de ser tomada en cuenta a la hora de escoger el modo de comunicación entre bloques adecuado según la aplicación.

En la Figura 3.6 se puede apreciar un ejemplo básico de funcionamiento del envío de mensajes. El bloque Message Strobe genera un mensaje cada segundo con el string TEST como contenido, que es imprimido por la terminal por el bloque Message Debug. Como nota adicional, nótese que la conexión entre una salida y una entrada asíncrona se representa en GRC con una línea discontinua.

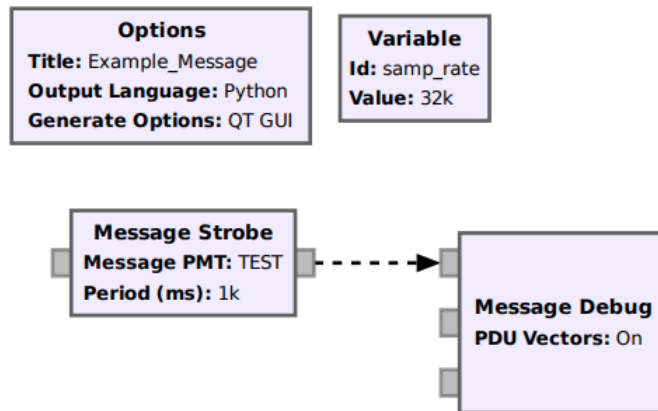


Figura 3.6.- Ejemplo de un esquema en GNU Radio en el que se hace uso del envío de mensajes (message passing).

## 3.2.- DISPOSITIVOS USRP

En este apartado se presenta la gama de dispositivos USRP del fabricante Ettus Research, introduciendo de manera esquemática su estructura y prestaciones. Tras una introducción y revisión de la arquitectura general que todos los USRPs comparten, se describirá más detenidamente el modelo empleado en la realización de este TFM: el B200. Por último, se explicará la compatibilidad de estos dispositivos con la herramienta de software GNU Radio descrita en el Apartado 3.1.

### 3.2.1.- INTRODUCCIÓN

Los USRP del fabricante Ettus Research [2] son una gama de dispositivos de SDR que incluyen en un sistema integrado todos los elementos de hardware necesarios para la implementación de aplicaciones en esta tecnología. Esto supone incluir no solo los elementos físicos del front-end y back-end para la transmisión y recepción respectivamente, sino también que los parámetros más destacables de estos puedan ser modificados dentro de un rango. Dicha capacidad de configuración es clave de cara a aprovechar al máximo la versatilidad y capacidad de adaptación del paradigma SDR. En esta misma línea, el ajuste de estos parámetros debe ser llevado a cabo por una unidad central de forma coordinada y en tiempo real. En el caso de los USRP dicha unidad es una Field Programmable Gate Array (FPGA), la cual se encarga no solo de la configuración de los elementos de hardware, sino de una parte del procesamiento de las muestras digitales.

Por otro lado, los USRP permiten la comunicación de su sistema con una unidad de procesamiento externa, como puede ser la CPU de un ordenador personal genérico. La razón de ser de esta característica es la de aumentar la accesibilidad y versatilidad del sistema, una vez más en línea con el paradigma SDR. Dicha comunicación se implementa físicamente a



Figura 3.7.- USRP modelo B200 recubierto con la carcasa diseñada para dicho dispositivo.

través de una interfaz que varía según el modelo y se define a nivel de software mediante la librería USRP Hardware Driver (UHD), programada en los lenguajes C y C++.

En la Figura 3.7 se muestra el modelo USRP B200, descrito más en detalle en el Apartado 3.2.3. El código fuente de UHD, que en su mayoría es de software libre y código abierto, se recoge en [19]. El manual tanto de los USRP como de UHD se puede encontrar en [20].

Por último, al igual que se hizo con GNU Radio en el Apartado 3.1.1, es conveniente justificar la elección de los dispositivos USRP como herramienta de hardware. Además de su compatibilidad con GNU Radio, los USRP están, sin lugar a dudas, entre los dispositivos de SDR más extendidos en el mercado, con la consiguiente fiabilidad que ello supone, así como la existencia de una amplia comunidad. Por otro lado, la oferta de una amplia gama de modelos con prestaciones y precios muy diversos facilita la optimización de recursos a la hora de escoger una de las opciones según la aplicación a desarrollar. Además, esta característica es especialmente atractiva debido a que los distintos modelos guardan entre sí cierta compatibilidad basada en UHD. Esto se debe a que, en general, se puede emplear esta librería de forma transparente al modelo de USRP utilizado. El resultado es que el código generado para el desarrollo de una aplicación con un modelo determinado pueda ser reutilizado con pocos cambios (o incluso, ninguno) para la implementación de dicha aplicación utilizando otro modelo.

### 3.2.2.- ARQUITECTURA GENERAL

La arquitectura general de un dispositivo USRP conectado a una CPU externa se muestra en la Figura 3.8 [21]. Aunque algunos modelos cuentan con varios transmisores y/o receptores, en este esquema se considera el caso más simple en el que hay un un solo transmisor y un solo receptor. Para todos los modelos, las interfaces de entrada/salida de señal se imple-

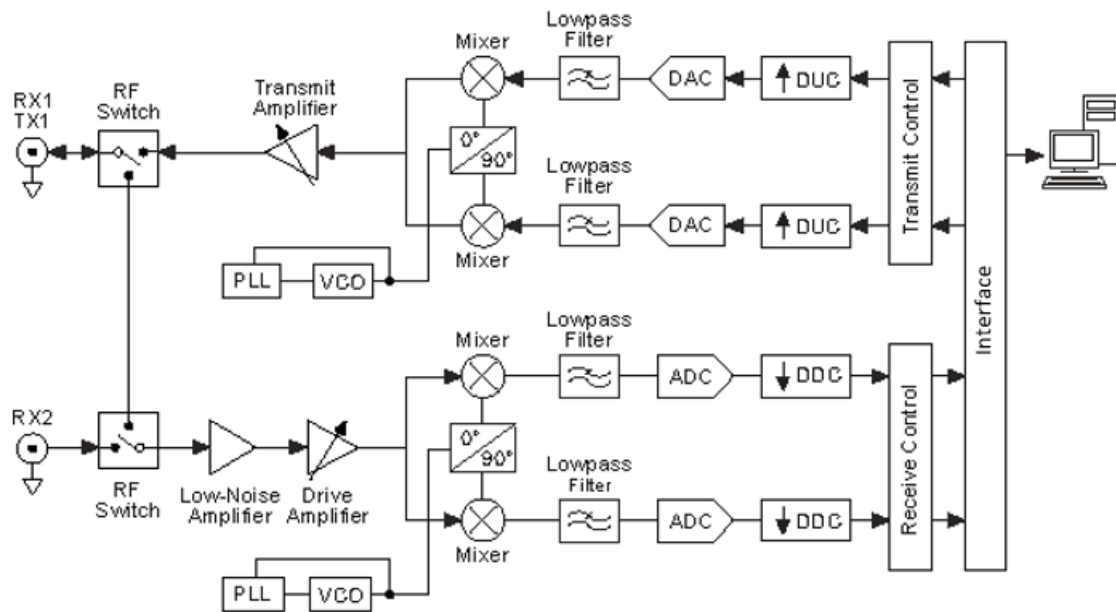


Figura 3.8.- Arquitectura general de un dispositivo USRP.

mentan por medio de conectores de tipo SubMiniature version A (SMA). Además, se puede apreciar que una de las dos interfaces puede conmutar entre transmisión y recepción gracias al elemento RF Switch. Esta se representa por RX1 y TX1, mientras que la otra, solo de entrada, se denomina RX2.

Pasando al análisis del transmisor, y situándose en la parte superior derecha de la Figura 3.8, la cadena se origina con la CPU externa enviando un flujo de muestras digitales a través de una interfaz física designada como Interface, cuya implementación varía según el modelo. Dichas muestras serán complejas, esto es, cada una de ellas incluye un valor para la componente en fase y otro para la componente en cuadratura. La separación de las componentes I y Q, así como otras funciones, corren a cargo del bloque Transmit Control. Ambas secuencias son tratadas posteriormente por un Digital Up Converter (DUP) antes de entrar al conversor analógico digital o Digital To Analog Converter (DAC). Una vez generadas las señales analógicas correspondientes, estas son filtradas paso-bajo por el Lowpass Filter, y moduladas según un esquema homodino por el mezclador o Mixer. La señal portadora a la frecuencia de canal,  $f_c$ , es generada por medio de un Voltage Controlled Oscillator (VCO), cuya fase está sincronizada por un Phase Locked Loop (PLL). Por último, al final de la cadena se incluye un amplificador de ganancia variable representado por el Transmit Amplifier.

En segundo lugar se analizará el receptor. Partiendo de la esquina inferior izquierda de la Figura 3.8, la señal recibida es amplificada por un Low Noise Amplifier (LNA), o amplificador de bajo ruido, y un amplificador de ganancia ajustable representado por el Drive Amplifier. Tras una demodulación homodina análoga a la modulación del transmisor, las componentes I y Q de la señal pasan por el Analog To Digital Converter (ADC), o conversor analógico digital, y un Digital Down Converter (DDC). Finalmente, el bloque Receive Con-

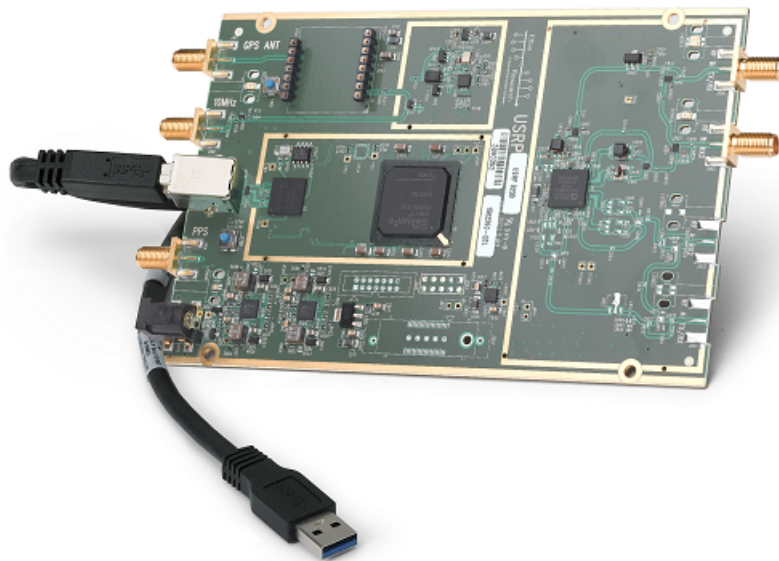


Figura 3.9.- Placa del USRP modelo B200.

trol acondicionará las muestras digitales para ser enviadas a la CPU a través de la Interface.

Los bloques DUC y DDC permiten realizar un ajuste fino de frecuencia, ya que el proporcionado por el VCO puede no ser suficientemente preciso para la aplicación desarrollada.

Nótese que esta descripción no ha utilizado valores cuantitativos de parámetros de interés como el rango de ganancia del Transmit Amplifier o la tasa de muestreo del ADC. Esto se debe a que son parámetros variables según el modelo empleado, como también lo es la implementación física de los bloques aquí presentados de forma conceptual.

Por último, y aunque no incluido en el esquema de la Figura 3.8, los USRP también incluyen entradas para señales de referencia que permiten la sincronización con otros dispositivos. Si bien estas entradas pueden resultar fundamentales para algunas aplicaciones, por ejemplo, en sistemas MIMO, no se han usado en el desarrollo del presente trabajo y, por ello, no se describen en profundidad.

### 3.2.3.- USRP B200

En el desarrollo de este TFM se emplean dos dispositivos modelo USRP B200 [22]. En la Figura 3.7 se puede observar este modelo recubierto por su carcasa, mientras que la placa en sí se muestra en la Figura 3.9. El diagrama de bloques del modelo B200 se puede ver en la Figura 3.10, en el cual se pueden identificar los elementos de la arquitectura general de un USRP explicados en el Apartado 3.2.2 y mostrados en la Figura 3.8.

En primer lugar, el bloque general de Interface de un USRP es implementado en el B200 mediante una conexión USB 3.0, que además sirve para la alimentación del dispositivo. Dicha conexión también puede ser USB 2.0, pero se empeoran las prestaciones en cuanto a

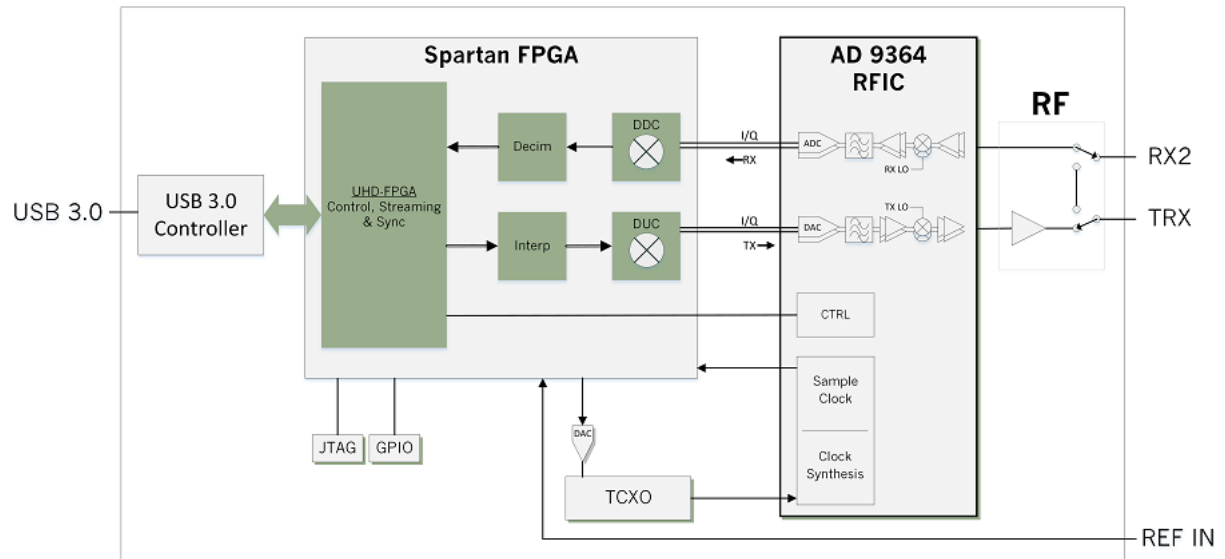


Figura 3.10.- Diagrama de bloques del USRP modelo B200.

la máxima tasa de muestreo [21]. Los bloques de la arquitectura general Transmit Control y Receive Control corren a cargo de un circuito integrado diseñado para la conexión USB 3.0, denotado como USB 3.0 Controller. La FPGA, modelo Xilinx Spartan 6 XC6SLX75 [23], sirve de unidad de control e implementa los bloques DDC y DUC, entre otras funciones. Los bloques de diezmado e interpolación, Decim e Interp, son representados en este esquema de manera independiente, si bien eran incluidos dentro del DDC y DUC respectivamente en la Figura 3.8. Los conversores ADC, DAC y elementos de radiofrecuencia (RF) como los filtros, mezcladores, amplificadores, PLLs y VCOs son implementados mediante el uso del chip de Analog Devices AD9364 [24]. La única excepción es la de un amplificador de ganancia fija al final de la etapa transmisora, que desde el punto de vista de la arquitectura se considera parte del Transmite Amplifier.

Póngase ahora la atención en las interfaces entre la FPGA y el integrado AD9364. En primer lugar, las líneas I/Q sirven para el intercambio de muestras digitales. La interfaz conectada al bloque CTRL designa las conexiones por medio de las cuales la FPGA controla los parámetros de configuración del chip y, por ende, el ajuste de sus elementos. Por último, la señal de reloj del AD9364 es generada en el bloque Clock Synthesis en base a la referencia del oscilador Temperature Compensated Crystal Oscillator (TCXO), a su vez controlado por la FPGA a través de un DAC. La señal resultante, representada en Sample Clock, es enviada de vuelta a la FPGA y pasa a ser la referencia común entre ambos sistemas.

Las interfaces REF IN y GPIO sirven para la sincronización de la señal de reloj con otros dispositivos, pero como se adelantó en el Apartado 3.2.2, estas no son empleadas en este TFM. En cuanto a la interfaz JTAG, su propósito es la programación y depuración de la FPGA por medio del estándar con el mismo nombre.

Por último, se comentan algunas de las prestaciones más relevantes del modelo B200



[22]. En primer lugar, la tasa de muestreo máxima del ADC y DAC es de 61.44 MS/s, con una resolución de 12 bits y considerando muestras complejas (cada muestra tiene un valor de la componente I y otro de la componente Q). Esto permitiría, según el límite del criterio de Nyquist, un ancho de banda máximo de 61.44 MHz de la señal paso banda. No obstante, el fabricante asegura un rango desde 200 kHz hasta los 56 MHz. El rango de la frecuencia de canal,  $f_c$ , es de 70 MHz a 6 GHz, tratándose además de un dispositivo full-duplex en el que su único transmisor y receptor pueden funcionar simultáneamente. En cuanto a los parámetros de RF, los valores máximos de potencia son de 20 dBm para la salida y de -15 dBm en la entrada, estando todos los puertos adaptados a 50  $\Omega$ . La ganancia disponible del transmisor es de 76 dB, y la del receptor, de 89.8 dB. La figura de ruido del receptor es siempre inferior a 8 dB y el punto de intercepción de tercer orden a la entrada es de -20 dBm. Por último, el valor RMS del ruido de fase es de 1° medido a 3.5 GHz, y de 1.5° grados a 6 GHz.

Para el lector interesado, una comparativa general de las prestaciones de los distintos modelos de USRP se recoge en [25].

### 3.2.4.- COMPATIBILIDAD CON GNU RADIO

Como ya se adelantó en el Apartado 3.2.1, la compatibilidad de los dispositivos USRP con GNU Radio se basa en UHD, una librería programada en los lenguajes C y C++ cuyo código puede ser ejecutado en un procesador genérico y que es capaz de comunicarse con los dispositivos USRP. Concretamente, UHD permite la configuración de la FPGA, cuyo código está escrito en Verilog, en los parámetros que afectan de manera más crítica en la transmisión y recepción del USRP. Esta característica no excluye la posibilidad de que el usuario programe la FPGA de manera libre, si bien esta opción no es usada en este TFM. No obstante, es una característica a tener en cuenta, especialmente en aplicaciones de alto coste computacional. En estos casos, la traslación de las funciones de procesamiento digital más críticas de un procesador genérico a la FPGA puede ser una solución altamente eficiente. Por otro lado, y aunque no es el caso del B200, existen algunos USRP que incorporan un procesador genérico integrado, lo que permite la utilización de UHD sin necesidad de un sistema externo.

Adicionalmente, UHD incorpora una API capaz de interactuar con otros entornos, entre los que se encuentra GNU Radio. De hecho, GNU Radio incorpora bloques por defecto que hacen uso de esta API, y por ende, permiten la configuración de cualquier modelo USRP. Un ejemplo básico de esta característica se puede observar en la Figura 3.11, donde el bloque UHD: USRP Source configura el USRP conectado para que en su interfaz RX2 demodule la señal de entrada con una frecuencia de canal,  $f_c$ , de 800 MHz, una tasa de muestreo del ADC de 1 MS/s, y una ganancia total de la cadena receptora de 20 dB. Además, el bloque incluye una entrada de tipo Async Message que permite la configuración de sus parámetros durante

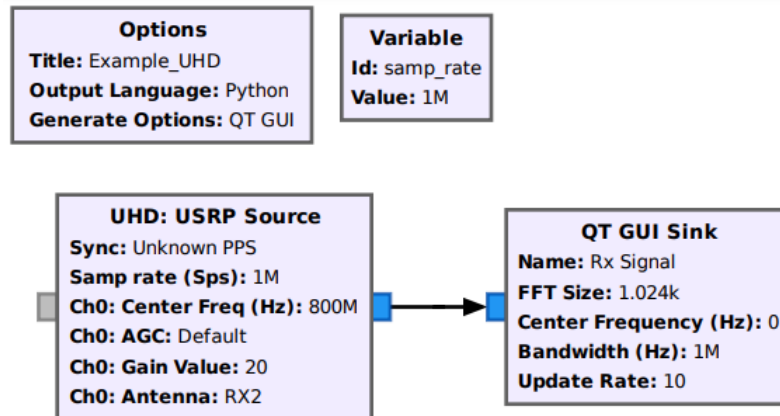


Figura 3.11.- Ejemplo de un esquema en GNU Radio que incluye el bloque UHD: USRP Source.

la ejecución del programa. Las muestras digitales recibidas desde el USRP son enviadas al bloque QT GUI Sink, que permite la visualización temporal y espectral de las mismas.

Como apunte final, cabe señalar que el bloque UHD: USRP Source permite configurar muchos otros parámetros aparte de los aquí expuestos, todo ello de forma transparente al modelo concreto de USRP utilizado. En el caso de que alguno de estos parámetros no puedan ser configurados para el modelo de USRP conectado, o que los valores pedidos estén fuera del rango adecuado, UHD ignorará la petición o la adaptará dentro de los rangos posibles. Esta característica denota la compatibilidad de UHD con cualquier modelo de USRP, y como consecuencia, la compatibilidad de un mismo esquema de GNU Radio para ser ejecutado con distintos modelos, eso sí, siendo siempre consciente de las prestaciones ofrecidas en cada caso.





## 4. DISEÑO DE SUBSISTEMAS

En este capítulo se aborda el diseño de los subsistemas que posteriormente permiten el desarrollo del sistema final. Para la integración de este se emplean los subsistemas aquí presentados como herramienta básica de trabajo, realizando en ellos las modificaciones pertinentes y añadiendo otros elementos complementarios. Sirva por tanto este capítulo como introducción conceptual al diseño de un sistema de comunicaciones en GNU Radio basado en la transmisión y recepción de paquetes.

### 4.1.- TRANSMISOR BASE

En este apartado se muestra el diseño básico de un transmisor en GNU Radio para una comunicación basada en paquetes compuestos por cabecera y carga útil, el cual será denotado como Transmisor Base. Si bien el transmisor del sistema final (Apartado 5.1) podrá modular la carga útil de los paquetes utilizando distintas constelaciones, para esta explicación inicial se partirá del empleo exclusivo de las modulaciones BPSK y QPSK para la cabecera y carga útil respectivamente. Ambas son modulaciones de tipo M-PSK, descritas en el Apartado 2.2.2. La Figura 4.1 muestra el esquema de bloques del Transmisor Base en GNU Radio, generado con ayuda de GRC. En la descripción del esquema se indicará la correspondencia de cada una de sus partes con los bloques teóricos descritos en el Apartado 2.1 y mostrados en la Figura 2.1.

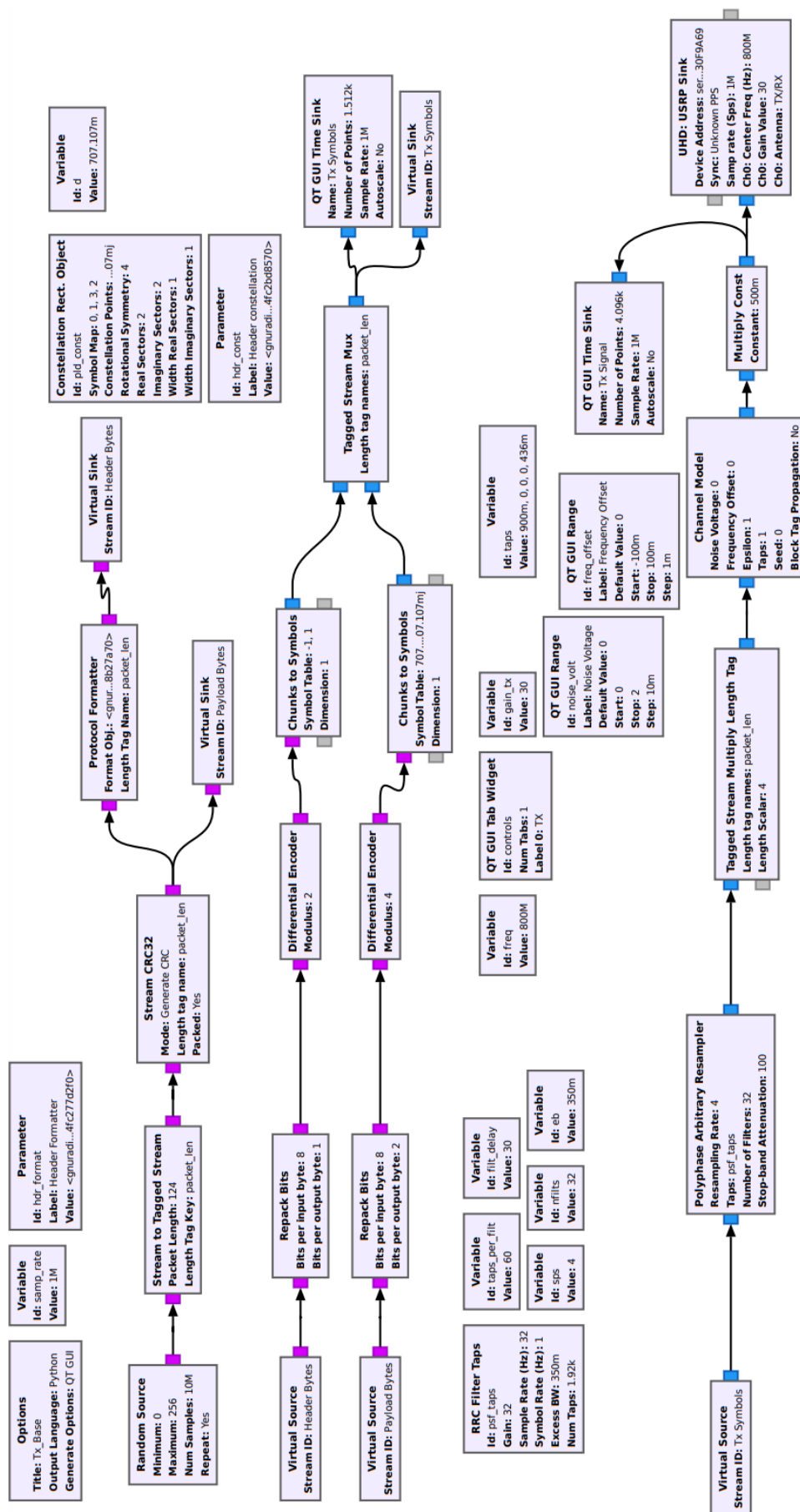


Figura 4.1.- Esquema del Transmisor Base en GNU Radio (GRC).



La Fuente de Información viene dada en este caso por el bloque Random Source, el cual genera una secuencia continua de bytes aleatorios, esto es, valores enteros comprendidos entre 0 y 255 que conformarán la carga útil de la comunicación. El flujo de salida atraviesa el bloque Stream to Tagged Stream. En él se configura la variable Packet Length con el valor 124, y Length Tag Key con packet\_len. Esto implica que el bloque introducirá cada 124 bytes una etiqueta (Apartado 3.1.3), que contendrá justamente la el valor 124 y cuya clave será packet\_length. A continuación, el flujo etiquetado resultante se introduce en el bloque Stream CRC32. La función de este bloque es la de añadir el prefijo de 4 bytes propio del código de detección de errores de redundancia cíclica CRC32 [13]. Para realizar esta acción se observa la etiqueta cuya clave coincida con el valor introducido en Length tag name, lo que en este caso conduce a añadir el prefijo cada bloque de 124 bytes. En este punto del esquema se realiza la división entre la carga útil, denotada como Payload bytes, y la cabecera, designada por Header bytes. Los única función de los bloques Virtual Sink y Virtual Source es la de conexionar salidas y entradas de manera limpia y ordenada desde un punto de vista gráfico. Los bytes de la cabecera son generados gracias al bloque Protocol Formatter. Este se configura también con una clave de etiqueta y, además, con un objeto de tipo cabecera. Los objetos de tipo cabecera son instancias de una clase derivada de la clase abstracta header\_format\_base [16]. De esta manera, el bloque Protocol Formatter generará a su salida la cabecera correspondiente a cada bloque de información delimitado por la etiqueta packet\_len. Es decir, en el presente esquema se generará una cabecera cada 124 bytes. En este caso la clase escogida para la cabecera es header\_format\_default, programada en el core de GNU Radio dentro del módulo gr-digital, e instanciada en el esquema con el nombre de hdr\_format en su parte superior izquierda. La estructura de la cabecera consta de dos partes: un código de acceso de 8 bytes; y un campo de 2 bytes que indica la longitud de la cabecera y que se encuentra duplicado (4 bytes en total).

La diferenciación en dos flujos separados de la cabecera y carga útil permite el empleo de modulaciones distintas en cada una. Ambos flujos pasan primero por el bloque Repack Bits, donde los valores enteros entre 0 y 256 se descomprimen para representar valores dentro de los límites de cada constelación. Por ejemplo, para el caso de QPSK, entre 0 y 3, o lo que es lo mismo, 2 bits de información que dan lugar a 4 posibles símbolos. A continuación, el bloque Differential Encoder realiza la codificación diferencial (Apartado 2.2.2) de la información. Hasta este punto la acción de los bloques descritos después del Random Source se corresponde con la del Codificador descrito en el Apartado 2.1. El bloque Chunk to Symbols se corresponde directamente con el Modulador Discreto, realizando la asignación de coordenadas a cada símbolo para generar a su salida un flujo de muestras complejas. Las constelaciones de cada modulación se generan mediante objetos pertenecientes a una clase derivada de la clase abstracta constellation, dentro del módulo gr-digital [16].

En la siguiente etapa se debe realizar la multiplexación de los flujos cabecera y carga

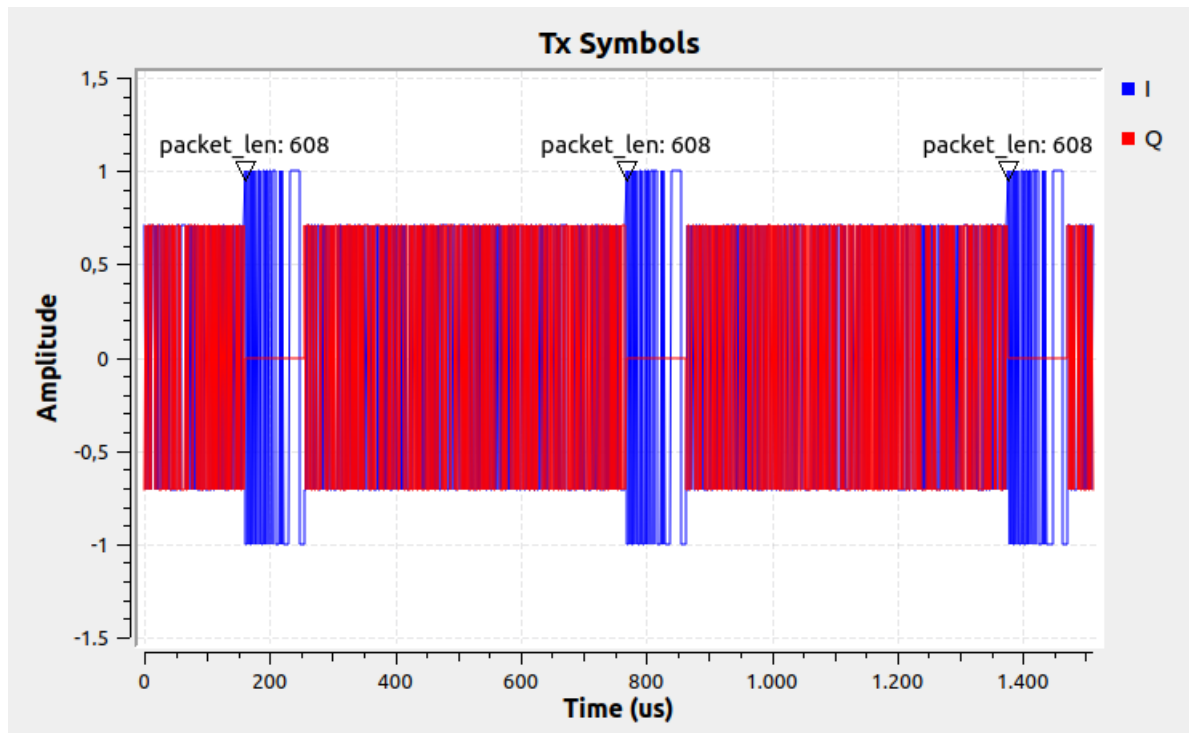


Figura 4.2.- Muestras complejas del flujo de símbolos transmitido en el Transmisor Base.

útil, tarea realizada por el bloque Tagged Stream Mux. Nótese que esta se hace también en base al valor de la etiqueta `packet_len`, para mantener la composición diseñada en los primeros bloques. Además, los distintos bloques de la cadena han ido actualizando el valor de dicha etiqueta de manera adecuada. Por ejemplo, al atravesar el bloque Repack Bits de QPSK su valor se cuadruplica, ya que por cada byte de entrada se generan 4 agrupaciones de 2 bits a la salida. En la Figura 4.2 se puede observar la representación temporal del flujo etiquetado como Tx Symbols, dada por el bloque QT GUI Time Sink. El hecho de que la componente en cuadratura sea nula para las muestras de BPSK permite la diferencia clara entre cabecera y carga útil. Además, la etiqueta `packet_len` designa el inicio de cada paquete y especifica la longitud en muestras de los mismo. Recuérdese que se empleaban 12 bytes para la cabecera, que codificados en BPSK dan lugar a 96 muestras. Por su parte, la carga útil empleaba 128 bytes (124 de información y un prefijo de 4 del código CRC32), que con QPSK dan lugar a 512 muestras. La suma da el valor indicado en la etiqueta: 608 muestras.

El siguiente paso consiste en la generación de la señal a transmitir, lo que se corresponde teóricamente con el Modulador de Señal (Apartado 2.1). El bloque Poyphase Arbitrary Resampler se configura para implementar un filtro digital de interpolación de respuesta igual a la función raíz de coseno alzado (2.8) con factor de roll-off  $\alpha = 0.35$ . El factor de interpolación escogido es de 4 muestras por símbolo, utilizándose el bloque Tagged Stream Multiply Length Tag para actualizar el valor de la etiqueta `packet_len`, función que en este caso no realiza el bloque anterior. La representación temporal de la señal digital a transmitir se muestra en la Figura 4.3. Como se puede observar, el número de muestras por paquete es

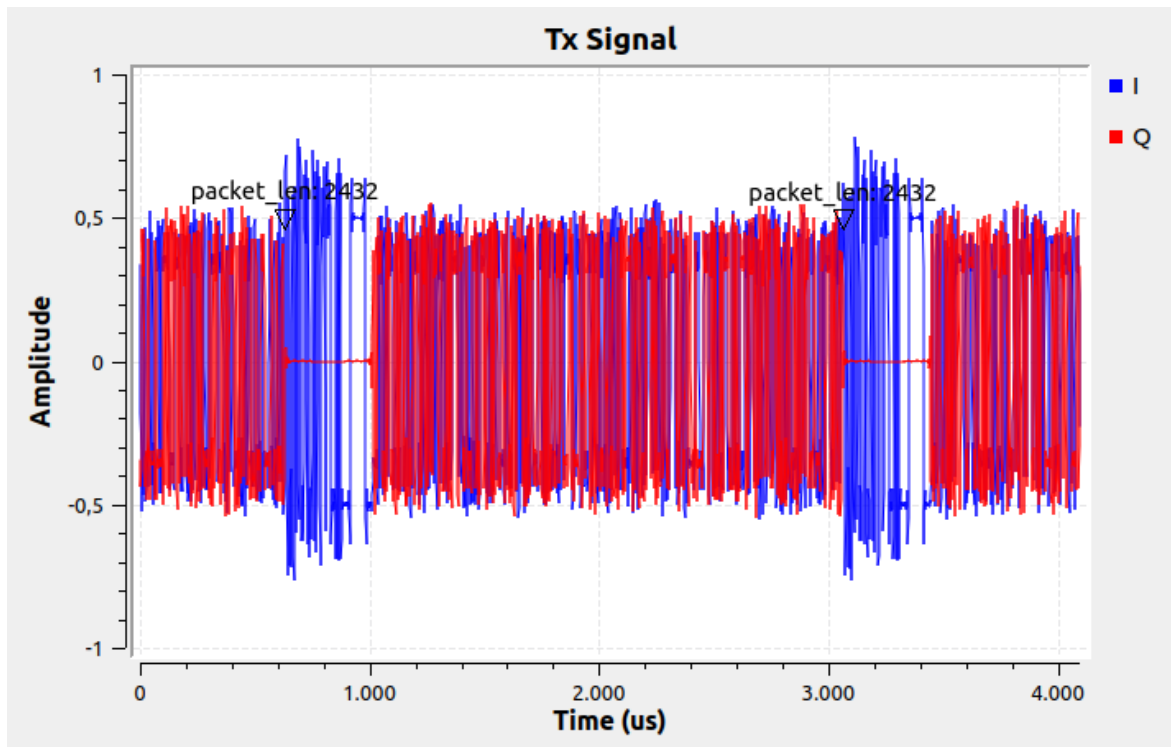


Figura 4.3.- Señal digital a transmitir enviada al USRP en el Transmisor Base.

finalmente de 2432. Nótese que la variable `samp_rate`, en este esquema con un valor igual a 1 MS/s, designa el número de muestras por segundo que se envían al USRP para su transmisión. Con esta información se puede calcular, por ejemplo, el número de paquetes, símbolos, o bytes de la carga útil transmitidos por segundo.

Por último, el bloque Channel Model sirve para la simulación de un canal variante en el tiempo controlado digitalmente, lo que permite realizar pruebas con las perturbaciones introducidas en el Apartado 2.3.2, y que junto con el canal físico representa el bloque Canal (Figura 2.1). El bloque UHD:USRP Sink, introducido en el Apartado 3.2.4, controla el USRP conectado para realizar la conversión digital-analógica y modular la señal de acuerdo con la función del Modulador de Canal (Apartado 2.1). En este caso se configura una frecuencia de canal,  $f_c$ , de 800 MHz, y una ganancia de 30 dB. Por otro lado, la función del bloque Multiply Const no es otra que la realizar un escalado digital para evitar que las muestras digitales enviadas al USRP estén fuera de su rango de trabajo, este es, entre -1 y 1. Las muestras con amplitud mayor que 1 saturarán a +1 o -1 en función de su signo.

A modo de resumen, la Tabla 4.1 recoge la estructura de los paquetes generados por el Transmisor Base. Recuérdese que el bloque Polyphase Arbitrary Resample genera 4 muestras a su salida por cada muestra de entrada.



Paquete, 140 bytes, 2432 muestras				
Cabecera, BPSK, 12 bytes, 384 muestras			Carga útil, 128 bytes, 2048 muestras	
CA (8 bytes)	LCU (2 bytes)	LCU (2 bytes)	Info (124 bytes)	CRC32 (4 bytes)

Tabla 4.1.- Estructura de los paquetes generados por el Transmisor Base. CA indica Código de Acceso y LCU Longitud de la Carga Útil.

## 4.2.- RECEPTOR BASE

En este apartado se detalla el diseño del receptor, denotado como Receptor Base. La Figura 4.4 recoge el esquema de bloques en GNU Radio, generado con GRC. El bloque UHD:USRP Source controla el USRP receptor y genera su flujo de salida con las muestras que recibe de este. Se correspondería así con el Demodulador de Canal explicado en el Apartado 2.1 y mostrado en la Figura 2.1. Para este caso se configura el receptor con un frecuencia de canal,  $f_c$ , de 800 MHz, y una ganancia de 30 dB. En el Apartado 5.3 se profundizará más en detalle en el resto de parámetros del bloque. A continuación, Energy Scale es el primer bloque visto hasta ahora que no pertenece al core de GNU Radio, sino que ha sido programado expresamente para la realización de este TFM, incluyéndose en el módulo out-of-tree gr-TFMv5 (Apartado 3.1.2). Su función es la de realizar un escalado digital adaptativo que asegure que el valor medio de las muestras a su salida se aproxime a una constante. Esta función cumple dos objetivos. El primero, que las muestras enviadas desde el USRP estén dentro de los márgenes de trabajo de GNU Radio. El segundo, que estas se adecuen a las fronteras de decisión de la constelaciones. Nótese que este último punto no es necesario para modulaciones de fase como las M-PSK (Apartado 2.2.2), pero sí para las modulaciones de amplitud como M-QAM (Apartado 2.2.3), que se utilizarán en el sistema final. La energía media,  $\mathcal{E}_{av}$ , (2.10) de todas las constelaciones empleadas en este TFM será siempre 1, asumiendo equiprobabilidad de los símbolos y como se puede observar en la Figura 4.2 para BPSK y QPSK. Como se tratan de muestras digitales, no llevan asociadas una unidad física. El valor de la energía de la señal transmitida o recibida en, por ejemplo, Julios, dependerá de la configuración de los USRP. No obstante, la dispersión de las muestras digitales es suficiente como para realizar una estimación de la probabilidad de error, como se mostrará en capítulos posteriores. El valor de la energía media se introduce en la variable Energy Avg, Init Scale indica el factor de escala inicial y Length (samples) indica el período de actualización en muestras.

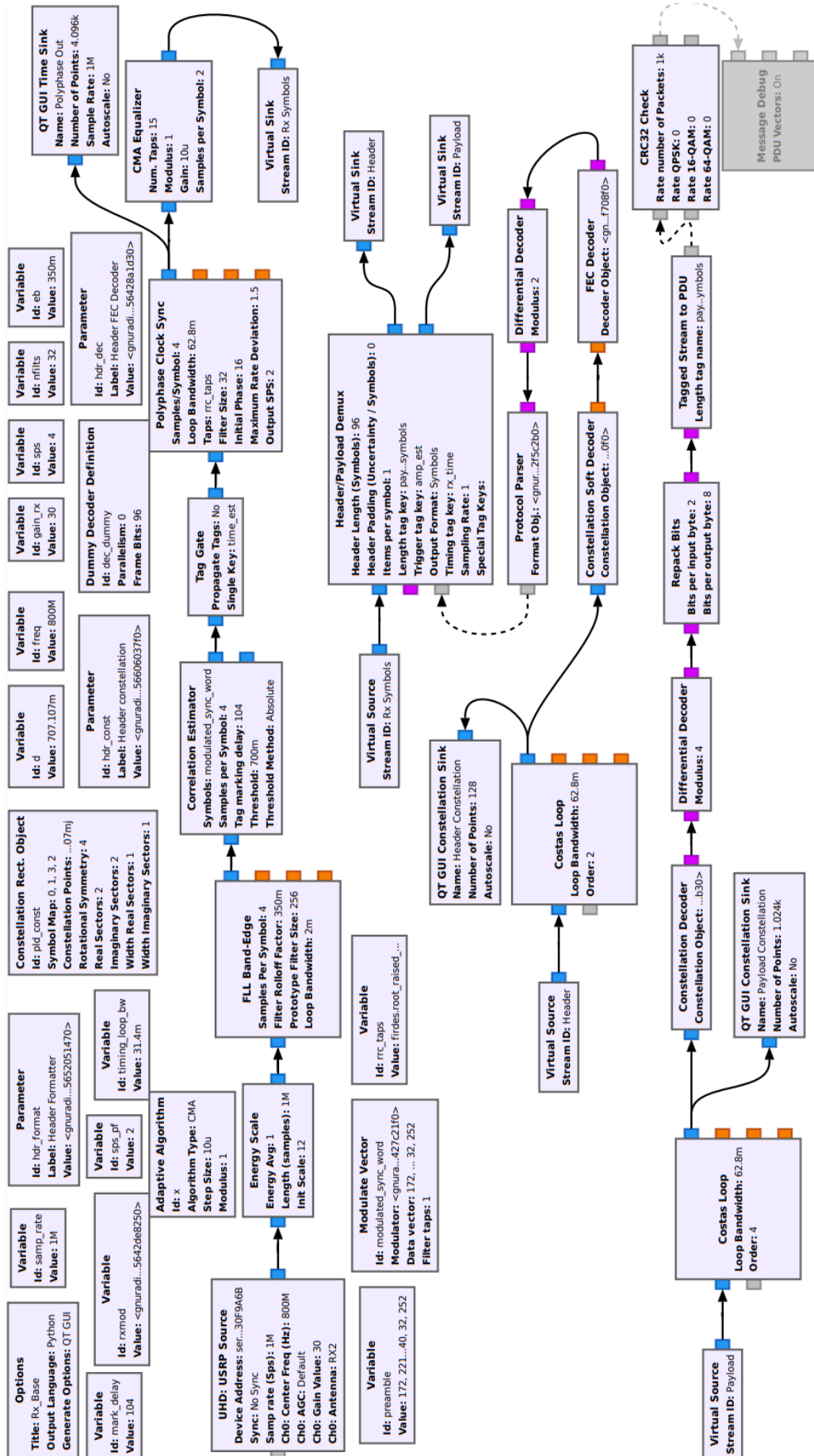


Figura 4.4.- Esquema del Receptor Base en GNU Radio (GRC).



Las muestras ya escaladas son introducidas en el bloque FLL Band-Edge, el cual realiza una corrección digital del desplazamiento en frecuencia debido al desajustes entre el oscilador del transmisor y el receptor. La técnica empleada se explica en detalle en [27]. El bloque Correlation Estimator genera una réplica de la señal digital resultante de modular el código de acceso de la cabecera con los parámetros correspondientes (BPSK, raíz de coseno alzado como función base, etc.). La señal recibida es correlada con esta réplica de manera que cuando el resultado supera un umbral se asume el inicio de la cabecera de un paquete. Se parte de la suposición de que el receptor conoce a priori la longitud de la cabecera, pero no la de la carga útil, que se obtendrá al demodular la cabecera (véase la Tabla 4.1). Además, se añaden una serie de etiquetas (Apartado 3.1.3) que identifican el inicio de la cabecera y que ofrecen información de utilidad para el resto de bloques. El bloque Tag Gate sirve justamente para eliminar una de esas etiquetas, `time_est`, por no cumplir esta última condición.

El Polyphase Clock Sync tiene una doble función. La primera, implementa el filtro digital adaptado a la función base raíz de coseno alzado. La segunda, es el bloque encargado de la sincronización temporal. Nótese que en todo el sistema se emplean 4 muestras digitales por símbolo, lo que supone en realidad un sobremuestreo. Sin embargo, el hecho de que el canal introduzca, entre otros efectos, un retardo desconocido, implica que el muestreo realizado por el USRP receptor no será óptimo. Este efecto ya se presentó en el Apartado 2.3.2 y el Apartado 2.4, hablando de la desincronización temporal entre los relojes del transmisor y receptor y el efecto de ISI que ello supone (2.48). El sobremuestreo permite que el Polyphase Clock Sync estime la muestra óptima que se obtendría en caso de sincronización a partir de las 4 muestras de las que se dispone. Para ello se emplea la técnica descrita en [28], basada en un banco de filtros digitales que introducen distintos desfases y entre los cuales se escoge en cada momento aquel para el cual la derivada de su salida es mínima. En este caso concreto la salida generada no es de 1 muestra por símbolo, sino de 2, para favorecer la acción del igualador. Este es implementado por el CMA Equalizer. La acción de este bloque es la igualación de canal lineal mediante el criterio CMA y la técnica FSE, tal y como se describe en el Apartado 2.2.4. Si se particulariza (2.62) para BPSK y QPSK se obtiene el mismo resultado, lo que permite que ambas modulaciones sean igualadas de manera conjunta.

La Figura 4.5 muestra la salida del Polyphase Clock Sync en un escenario en el que se ejecutan Transmisor Base y Receptor Base configurados según lo explicado hasta ahora y con un canal entre ellos formado por un cable Bayonet Neill-Concelman (BNC). Más información acerca del montaje se puede encontrar en el Apartado 6.2.1. En la gráfica se observan las etiquetas introducidas por el bloque Correlation Estimator, entre las que destacan `corr_start` y `phase_est`. La primera se sitúa en la muestra de inicio del paquete y su valor se corresponde con el resultado de la correlación. La segunda marca la muestra de inicio de la información después del código de acceso, y su valor indica una estimación del desfase de la señal recibida respecto de la transmitida en radianes. Observando detenidamente la Figura



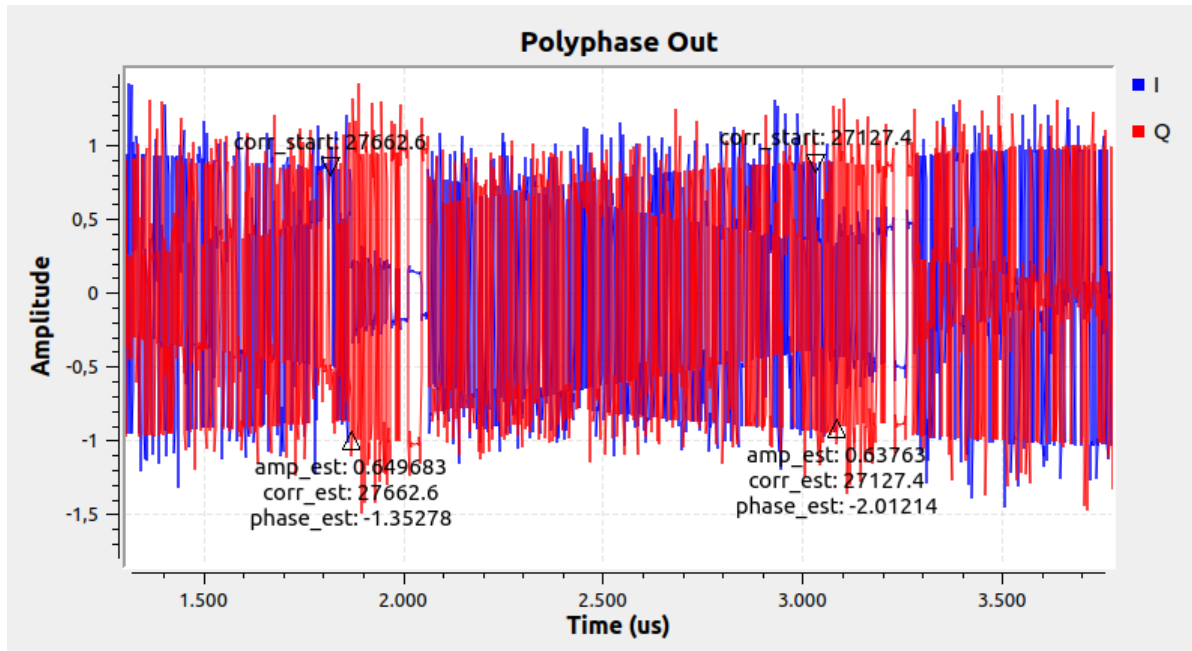


Figura 4.5.- Señal digital a la salida del bloque Polypahse Clock Syn en el Receptor Base.

4.2, se aprecia el efecto de este desfase sobre la cabecera, ya que sus muestras tienen componentes I y Q, a pesar de estar modulada en BPSK (sólo componente I). El que el desfase sea variable se debe a múltiples motivos. En primer lugar, el propio canal, formado no solo por el cable BNC sino por los componentes de RF de los USRP. En segundo lugar, los bloques digitales de la cadena receptora vistos ahora, cuya función es en muchos casos adaptativa, tal es el caso por ejemplo del FLL Band-Endge o el propio Polyphase Clock Sync . Este efecto será corregido más adelante con el Costas Loop. Como apunte importante, el desfase no afecta a la función del Correlation Estimator porque la correlación se calcula, en realidad, con el módulo de las señales al cuadrado.

El bloque Header/Payload Demux es un elemento central de la recepción de paquetes. Este bloque espera una etiqueta que marquen el inicio de la cabecera y, como conoce a priori la longitud de esta, mueve sus muestras a su salida superior y se queda a la espera. Las muestras de la cabecera son demoduladas para la extracción de su información, que se envía de vuelta al Header/Payload Demux, el cual utiliza la información recibida para conocer la longitud de la carga útil y mover esas muestras a su salida inferior, añadiendo una etiqueta de clave payload\_symbols y valor el número de muestras correspondiente. En este punto de la cadena el número de muestras coincide con el número de símbolos. Este proceso se repite para cada paquete.

En lo que se refiere a la cadena receptora de la cabecera, esta se inicia con el bloque Costas Loop. Como ya se había adelantado, su función es la corrección de fase de forma adaptativa, usando como valor inicial el proporcionado por la etiqueta phase\_est. El fundamento teórico de este bloque se describe en [29]. Todos los bloques descritos hasta ahora,

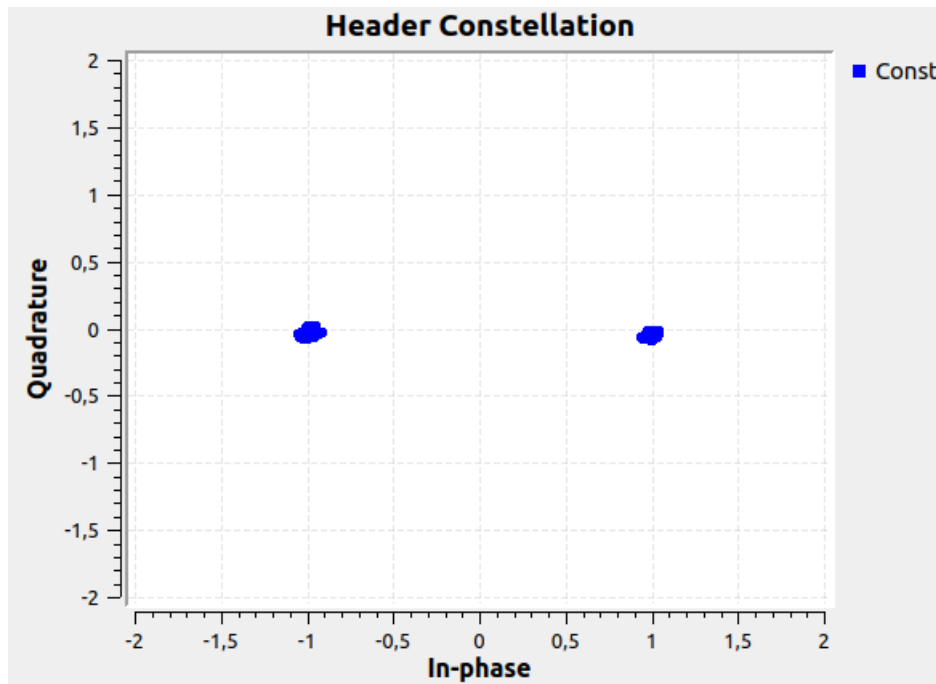


Figura 4.6.- Constelación BPSK recibida para la cabecera en el Receptor Base.

desde el Energy Scale hasta el Costa Loop, pueden ser considerados parte del Demodulador de Señal (Apartado 2.1). A continuación se incluyen los bloques Constellation Soft Decoder, FEC Decoder y Differential Decoder. El primero representa el Decisor, que completa el Demodulador Digital (Figura 2.1). El segundo y tercero conforman el Decodificador. En realidad, en este caso en el que no se incluye un código de corrección y/o detección de errores en la cabecera, la decisión se realiza símbolo a símbolo, lo que se conoce también como hard decoding, o decodificación dura. Además, la única función del FEC Decoder es la de asignar los símbolos ya decididos con su agrupación de bits correspondiente. El bloque Differential Decoding realiza la decodificación diferencial (Apartado 2.2.2). Por último, el bloque Protocol Parser recibe la información de la cabecera y decide si esta es válida o no. Para este caso se comprueba que su longitud sea la correcta y que los campos duplicados de la longitud de la carga útil sean idénticos (ver la Tabla 4.1). En caso positivo crea un mensaje (Apartado 3.1.3) con la información correspondiente que envía al Header/Payload Demux. En caso negativo envía un mensaje de error y el Header/Payload Demux dará el paquete por perdido y esperará a la recepción de la siguiente cabecera. La constelación recibida para la cabecera en el escenario de transmisión y recepción descrito anteriormente se observa en la Figura 4.6. Esta refleja una constelación BPSK como la mostrada en la Figura 2.4, con una ligera dispersión de los símbolos debida a los efectos del canal.

Para la cadena demoduladora de la carga útil las diferencias con la cabecera son el uso del bloque Constellation Decoder, que realiza directamente la decisión dura de los símbolos; y el bloque Repack Bits, que empaqueta la información de estos en bytes. La constelación QPSK recibida sobre la que se toman las decisiones de símbolos se muestra en la Figura

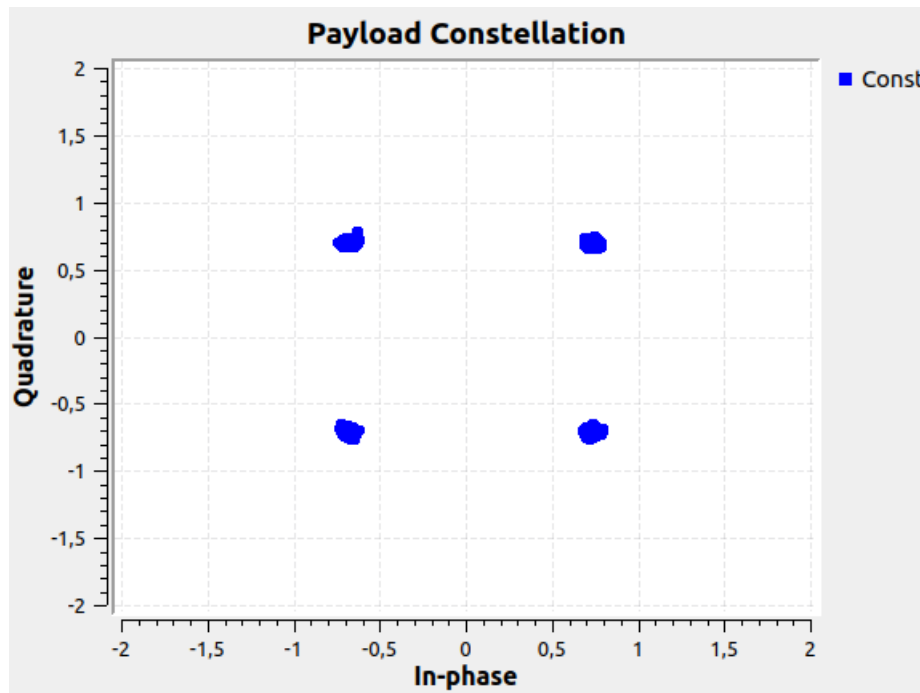


Figura 4.7.- Constelación QPSK recibida para la carga útil en el Receptor Base.

4.7. El bloque Tagged Stream to PDU utiliza la etiqueta `payload_symbols` para conocer la longitud de la carga útil de cada paquete, agrupar sus bytes y formar un mensaje de tipo PDU (Apartado 3.1.3). Por último, el bloque CRC32 Check, que como Energy Scale pertenece al módulo `gr-TFMv5`, realiza la detección de errores utilizando el campo CRC32 (Tabla 4.1). En caso de que se detecte algún error, el PDU es descartado, mientras que, si la comprobación es positiva, se elimina el campo CRC32 y se propaga en su salida un PDU que únicamente contiene el campo de la Información. Este bloque está basado en el bloque Async CRC32 del core de GNU Radio, pero incluye una cuenta de los paquetes totales recibidos y los paquetes válidos, o libres de error<sup>11</sup>. En el ejemplo se configura de manera que cada 1000 paquetes totales se indique cuántos de ellos han resultado válidos. El bloque cuenta también con una entrada, una salida, y otras variables adicionales no utilizadas en este caso, cuya utilidad se describe en el Apartado 5.3.1.

El bloque de la Información de Destino (Figura 2.1) viene representado en este caso por el Message Debug, que imprime en la terminal los PDU que recibe. Aunque útil para el testeo, cuando la tasa de muestreo es elevada conviene deshabilitarlo para evitar la congestión de la terminal, tal es el caso de este esquema.

Para el sistema formado por el Transmisor Base y el Receptor Base funcionando conjuntamente, cabe señalar, en primer lugar, que para el escenario descrito en el que el canal viene dado por un cable BNC se comprobó por medio del CRC32 Check que, tras un estado inicial de arranque, se recibía de manera correcta la totalidad de los paquetes. La puesta

<sup>11</sup>Como se comentará en el Apartado 6.1, existe la posibilidad de que el paquete contenga errores que no sean detectados por el código CRC32, pero se considera despreciable



a prueba del sistema para canales más complejos será desarrollada en el Apartado 6.2 y el Apartado 6.3 para el sistema final, que se basa en el Transmisor Base y el Receptor Base aquí descritos. En segundo lugar, destacar que el diseño tiene como ventaja fundamental la total independencia de los paquetes entre sí. Esto implica que no se necesita de una recepción de información continua, o un protocolo inicial que sincronice transmisor y receptor. Al contrario, cada paquete puede tener una modulación y/o longitud distintas en su carga útil, siempre y cuando su cabecera lo indique de manera adecuada. Además, la pérdida la comunicación en un momento dado no afectará en absoluto a la recepción de futuros paquetes cuando esta se reanude. Estas ventajas serán explotadas por el sistema final.

## 5. SISTEMA FINAL

En este capítulo se presenta el diseño del sistema final, objetivo último de este TFM. Este consiste en un sistema de comunicaciones adaptativo de SDR, implementado en GNU Radio en su parte software y dos USRP en su parte hardware. Se expone aquí una explicación detallada de su diseño, así como una descripción de los elementos que lo componen. Los subsistemas Transmisor Base y Receptor Base presentados en el Apartado 4.1 y el Apartado 4.2 respectivamente serán piezas básicas para la construcción del mismo.

### 5.1.- EXPLICACIÓN GENERAL DEL SISTEMA

El esquema general del sistema final, cuya notación se emplea durante este capítulo, se muestra en la Figura 5.1.

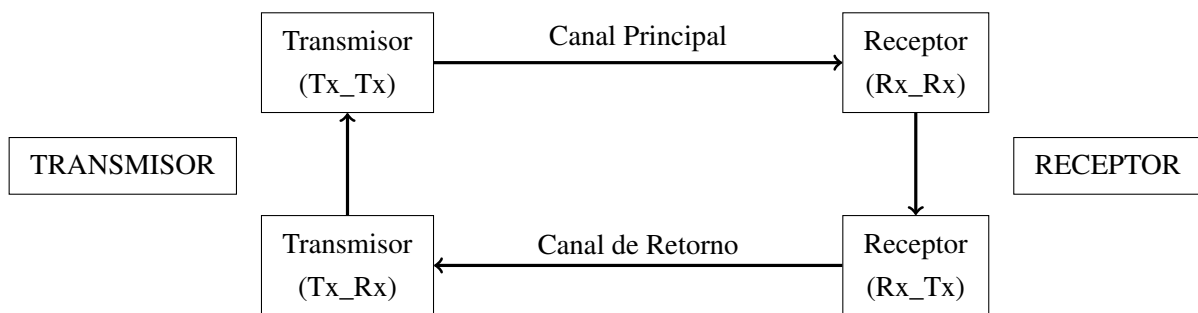


Figura 5.1.- Esquema general del sistema de comunicaciones adaptativo.

Como se puede observar, el sistema se divide en dos componentes principales denotados como Transmisor y Receptor. En realidad, ambos componentes están formados por sendos transceptores, pero su nombre hace referencia a su función en el flujo principal de información. El Transmisor genera dicho flujo desde su módulo Tx\_Tx y lo envía a través del Canal Principal, siendo recibido por el módulo Rx\_Rx del Receptor. Esta comunicación se basa en el uso de paquetes independientes, formados una cabecera y una carga útil, de acuerdo a lo explicado en el Apartado 4.1 y el Apartado 4.2. Sin embargo, aunque la información de la cabecera irá siempre modulada en BPSK para una robustez máxima frente a interferencias, la información de la carga útil podrá usar las modulaciones BPSK, QPSK, 16-QAM o 64-QAM descritas en el Apartado 2.2.2 y el Apartado 2.2.3. La modulación escogida para la carga útil será parte de la información contenida en la cabecera de cada paquete, de manera que puede variar en cada uno de ellos. El Transmisor tomará la decisión de qué modulación emplear en cada momento en base a la información que le llega del Receptor a través del Canal de Retorno. El Rx\_Rx analiza la calidad de la comunicación del Canal Principal en cada momento y se comunica con el Rx\_Tx, el cual transmite dicha información, que recibe



y demodula el Tx\_Rx para comunicársela a su vez al Tx\_Tx. La comunicación del Canal de Retorno se realiza también en forma de paquetes en los que cabecera y carga útil están modulados con BPSK. La información contenida en la carga útil será justamente la indique al Transmisor la necesidad de cambiar la modulación o, por el contrario, mantener la actual. Por otro lado, la comunicación del Canal de Retorno se realizará por norma general con una tasa de símbolo inferior a la del Canal Principal, ocupando así un menor ancho de banda.

Lo explicado hasta ahora describe de manera general la implementación de una modulación adaptativa, pero no justifica la necesidad o ventajas de esta técnica. Para ello es conveniente introducir primero las condiciones del escenario para el cual el sistema es diseñado. En él, el Transmisor está limitado por una potencia media,  $P_0$  (2.12), y un ancho de banda,  $W$ , siendo ambos valores fijos. Dicho ancho de banda impone además un período de símbolo,  $T$ , que, suponiendo el uso de la función raíz de coseno alzado como función base, vendrá dado a través de (2.9). Por otro lado, el sistema se implementa para la realización de una aplicación que requerirá de una tasa de error de bit menor a una cota máxima,  $BER_{max}$ , directamente relacionada con una probabilidad de error de bit máxima,  $P_{b,max}$ . En realidad el requisito de que  $P_0$  sea constante implica que la energía media de la constelación,  $\mathcal{E}_{av}$  (2.10), también ha de serlo. A partir de  $\mathcal{E}_{av}$  se puede obtener la energía media de las señales transmitidas,  $\mathcal{E}_0$  (2.11), y la energía media por bit transmitido,  $\mathcal{E}_b$ , en función del número de bits transmitidos por símbolo,  $N$  (2.13). Nótese que se trabaja con señales complejas para el cálculo de  $\mathcal{E}_0$ , y que  $N$  es igual a 1, 2, 4 y 6 para las modulaciones BPSK, QPSK, 16-QAM y 64-QAM respectivamente. Esto implica que, dada la misma  $\mathcal{E}_{av}$ ,  $\mathcal{E}_b$  es 6 veces menor para 64-QAM que para BPSK. Asumiendo además la misma densidad espectral de potencia de ruido,  $N_0$  (Apartado 2.3.2),  $\Gamma_b$  sería aproximadamente 7.78 dB menor para 64-QAM (2.15) que para BPSK. A partir de  $\Gamma_b$  y suponiendo Codificación Gray, resulta directo calcular la cota superior de  $P_b$  para las modulaciones de fase BPSK y QPSK (2.32) y las modulaciones de amplitud 16-QAM y 64-QAM (2.38). De este modo se puede también conocer el valor de  $\Gamma_b$  necesario en cada modulación para cumplir el requisito impuesto por  $P_{b,max}$ .

El funcionamiento del sistema será por tanto el siguiente. Los valores de  $P_0$ ,  $W$  han de ser constantes, lo que fija los valores de  $\mathcal{E}_{av}$  y  $T$  respectivamente. Tanto la cabecera en el Canal Principal como todo el paquete en el Canal de Retorno usarán siempre BPSK. La carga útil en el Canal Principal podrá usar BPSK, QPSK, 16-QAM y 64-QAM, pero se iniciará siempre con BPSK. Durante la comunicación, el Rx\_Rx realizará una estimación de  $\Gamma_b$  en tiempo real, que le permitirá realizar una estimación de la  $P_b$  en función de la modulación utilizada en este momento. No solo eso, sino que Rx\_Rx también puede estimar la  $P_b$  en caso de que se estuviese empleando cualquiera de las otras modulaciones. De esta manera, si el valor actual es inferior a  $P_{b,max}$ , y además se estima que esta condición se seguirá cumpliendo si se pasa a utilizar una modulación con  $N$  mayor, Rx\_Rx trasladará esta información a Rx\_Tx, que enviará una petición de cambio al Transmisor. De forma



análoga, si el valor estimado se acerca al límite  $P_{b_{max}}$ , la petición será la de emplear una modulación de menor  $N$  que proporcione un  $\Gamma_b$  mayor. El objetivo es así tender en cada momento a emplear aquella modulación que permita una mayor tasa binaria, pero que a la vez asegure que se cumple la condición de  $P_{b_{max}}$  manteniendo  $P_0$  y  $W$  constantes.

Como apunte final, señalar que un valor de  $\Gamma_b$  más elevado no implica necesariamente una menor probabilidad de error de bit para modulaciones distintas. Sin embargo, el escenario descrito y las modulaciones escogidas sí aseguran una disminución de  $P_b$  al cambiar a una modulación de menor  $N$  para un mismo canal y  $\mathcal{E}_{av}$  constante.

## 5.2.- TRANSMISOR

En este apartado se presenta el Transmisor, formado por el Tx\_Tx y el Tx\_Rx según lo mostrado en la Figura 5.1. Se explicarán ambos elementos por separado, así como la interacción entre ellos.

### 5.2.1.- Tx\_Tx

El esquema del Tx\_Tx en GNU Radio, generado con ayuda de GRC, se muestra en la Figura 5.2, la Figura 5.3 y la Figura 5.4. Si se observa el diseño, este consiste en realidad en la multiplexación de 4 instancias del Transmisor Base, explicado en el Apartado 4.1. En concreto, se tiene un Transmisor Base para cada una de las posibles modulaciones de la carga útil: BPSK, QPSK, 16-QAM y 64-QAM, según lo explicado en el Apartado 5.1. No solo eso, sino que cada instancia debe generar una cabecera distinta, ya que esta es la encargada de indicar el tipo de modulación de la carga útil. Esta necesidad conduce a la primera modificación clara respecto del Transmisor Base: el tipo de cabecera. En el primero se utilizaba la clase `header_format_default`, cuya estructura se observa en la Tabla 4.1. La naturaleza del sistema requiere en este caso la programación de una clase propia: `header_format_counter_dif`, incluida en el módulo `gr-TFMv5`. Esta clase deriva de `header_format_counter`, cuya estructura se basa en la de `header_format_default`, pero añadiendo dos campos adicionales. El primero de estos campos consiste en dos bytes para indicar el valor de bits por símbolo,  $N$ , en la constelación de la carga útil. El segundo se compone de otros 2 bytes para señalar el número de paquete. Adicionalmente, `header_format_counter_dif` añade dos bytes extras que devuelven la codificación diferencial a su estado inicial, lo que es necesario para que la decodificación se pueda realizar correctamente. Por otro lado, la longitud en bytes de la carga útil es diferente para cada modulación. El propósito de ello es que la longitud del paquete en número de muestras sea la misma en todos los casos, lo que implica que, si se mantienen constantes la tasa de muestreo y de símbolo, el tiempo de transmisión de un paquete será igual para las cuatro modulaciones. Las distintas estructura de los paquetes en función de la modulación empleada en la carga útil se muestran en la Tabla 5.1.



Paquete, (82/146/274/402), 2624 muestras							
C, BPSK, (18), 576 muestras						CU, (64/124/256/384), 2048 muestras	
CA (8)	LCU (2)	LCU (2)	BPS (2)	CP (2)	RE (2)	Info (60/120/252/380)	CRC32 (4)

Tabla 5.1.- Estructura de los paquetes generados por el Tx\_Tx. La leyenda de campos es la siguiente: C - Cabecera; CU - Carga Útil; CA - Código de Acceso; LCU - Longitud de la Carga Útil; BPS - Bits Por Símbolo en la carga útil; CP - Contador de Paquete; RS - Reseteo de Estado. Los valores entre paréntesis indican longitud en bytes. En los casos en los que se indican 4 valores estos se corresponden con la utilización en la carga útil de BPSK, QPSK, 16-QAM y 64-QAM respectivamente.



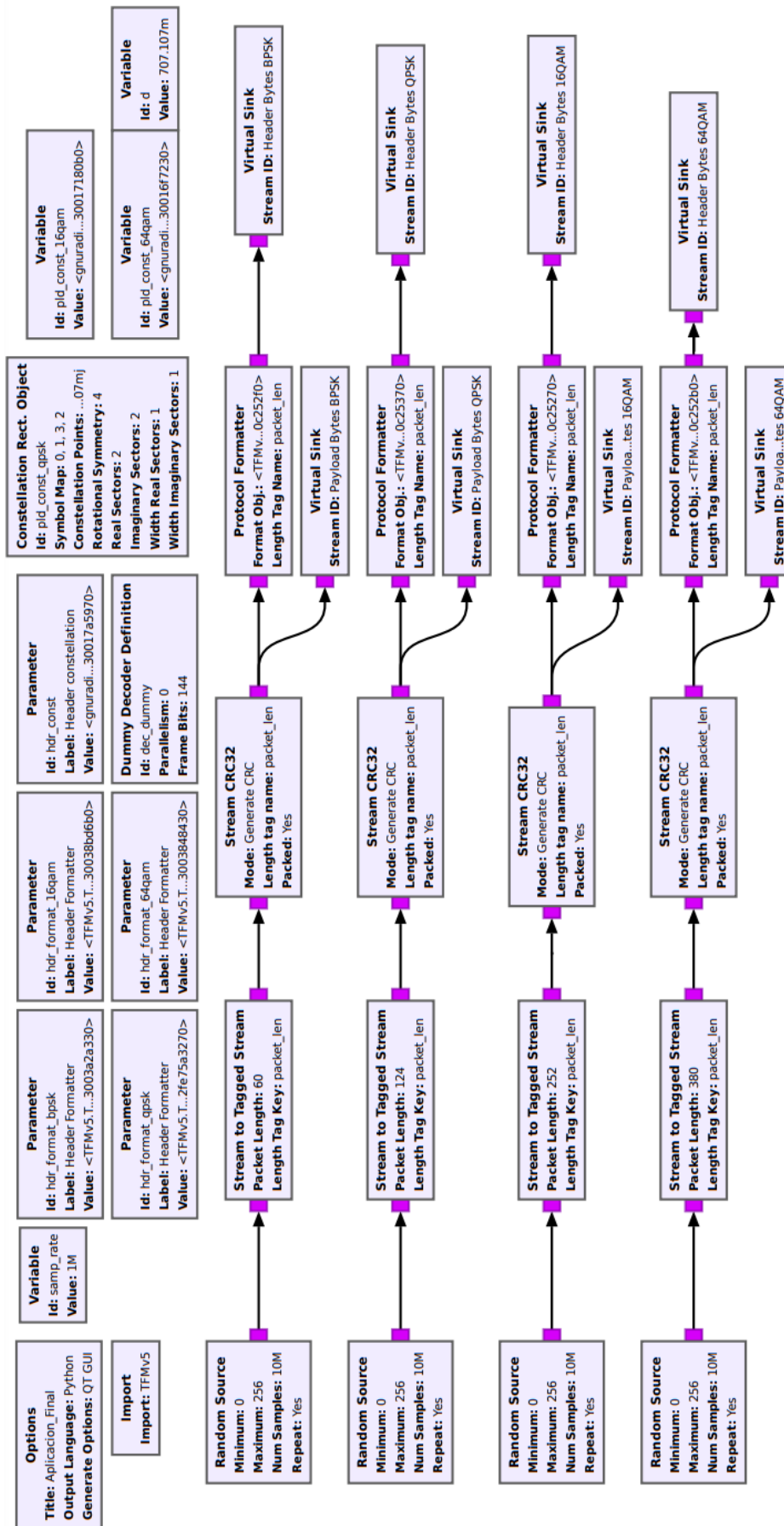


Figura 5.2.- (a) Esquema del Tx\_Tx en GNU Radio (GRC).

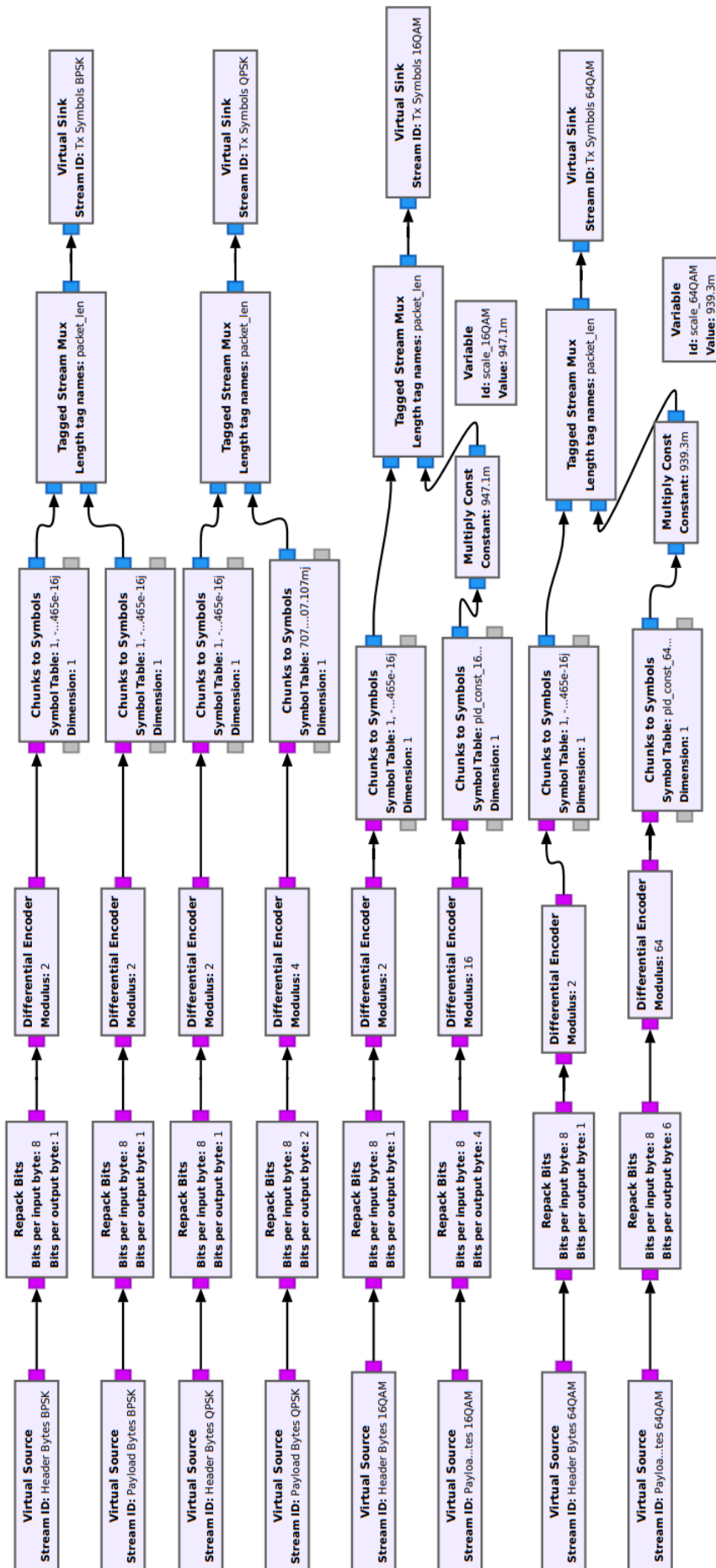


Figura 5.3.- (b) Esquema del Tx\_Tx en GNU Radio (GRC).

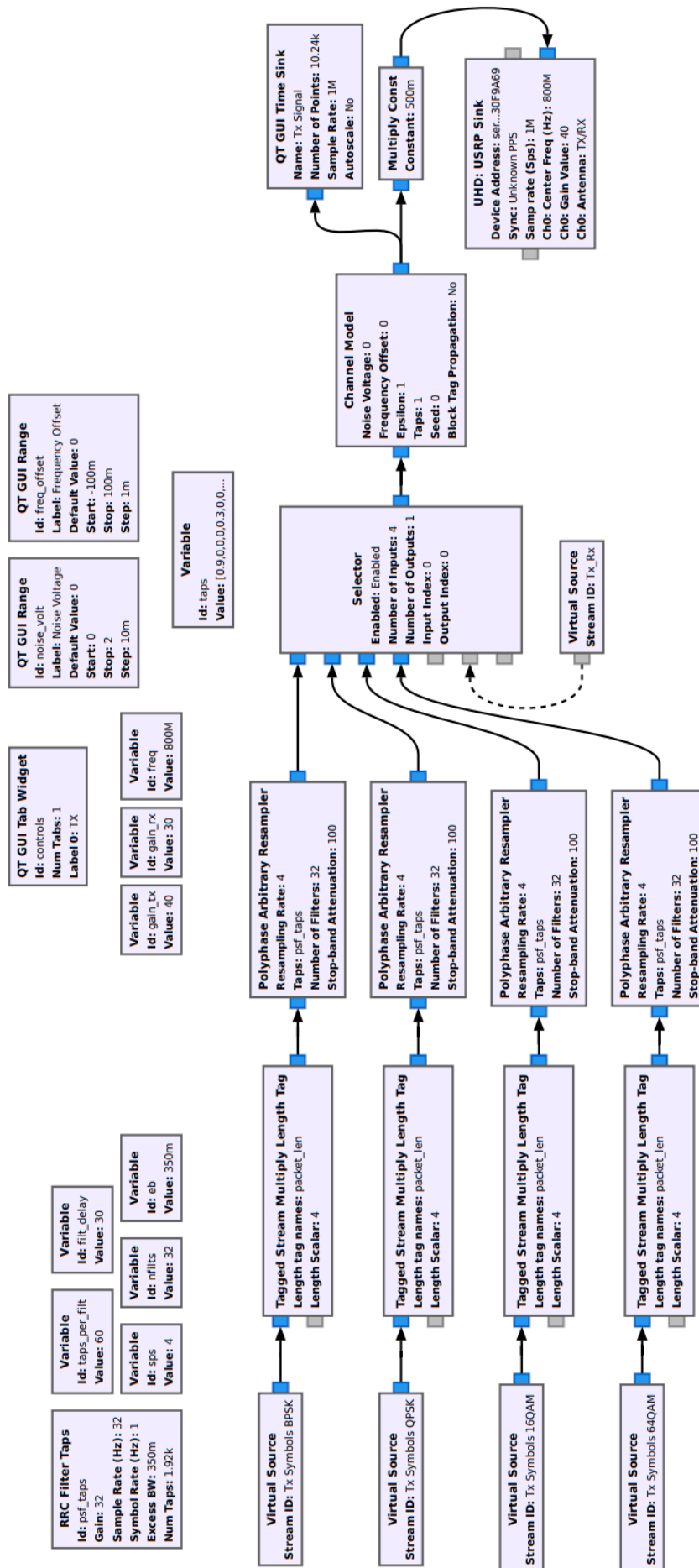


Figura 5.4.- (c) Esquema del Tx\_Tx en GNU Radio (GRC).



Llegados a este punto, procede explicar la necesidad del campo de reseteo de estado en la cabecera. En primer lugar, se recuerda que la codificación diferencial que añade el Transmisor Base se realiza de forma independiente entre cabecera y carga útil. A su vez, el Receptor Base (Apartado 4.2) debe conocer a priori los bits y la modulación del código de acceso al inicio de la cabecera para poder implementar la correlación que identifique el inicio del paquete de manera correcta. Al emplear codificación diferencial es necesario que el inicio de cada cabecera sea fijo y conocido por el receptor. Partiendo del uso de una constelación binaria en la que  $N = 1$  como BPSK, esto implica que todas las cabeceras contengan un número par de unos. La clase `header_format_counter_dif` se programa para que en cada cabecera se realice el conteo de los bits puestos a uno para ser compensado, en caso de ser impar, modificando uno de los bits en los dos bytes del campo de reseteo, que por defecto se encuentran a cero. Según lo explicado no serían necesarios dos bytes para esta función, sino un único bit añadido al final de la cabecera, pero la programación de las clases de tipo cabecera en GNU Radio tiende a asumir que su longitud es siempre igual a un número par de bytes. Nótese que esta característica no es necesaria en el Transmisor Base, incluso usando codificación diferencial. Esto se debe a que la cabecera en ese caso tomaba siempre el mismo valor, que tenía un número par de unos de por sí. Esto no es aplicable a este caso, pues los campos de la longitud del paquete en bytes, el valor de  $N$  en la carga útil y el contador de paquete son variables.

Los cuatro flujos producidos (uno por cada posible modulación) son introducidos en el bloque Selector que, como su nombre indica, realiza en cada momento la selección de una de sus entradas para ser trasladada a su salida. El bloque se configura para empezar seleccionando la entrada correspondiente a BPSK, pero cuenta con una entrada de tipo mensaje que permite el cambio de selección desde fuera, y que está conectada al Tx\_Rx. De esta manera, en cada momento se escogerá el flujo a transmitir adecuado en función de la información que se reciba desde el Receptor. La última etapa tras el Selector se corresponde nuevamente con el final del Transmisor Base. En este punto, cabe señalar el motivo del escalado de las constelaciones 16-QAM y 64-QAM a la salida del Chunk to Symbols mediante el bloque Multiply Const. Este se realiza para asegurar que  $\mathcal{E}_{av}$  se mantiene constante para todas las modulaciones (Apartado 5.1), e igual a 1 (Apartado 4.1). En la Figura 5.5 se muestra la señal digital a transmitir enviada al USRP en caso de emplear BPSK en la carga útil, mientras que la Figura 5.6 hace lo propio con 16-QAM.

Por último, es necesario aclarar dos aspectos que pueden haber sido ya percibidos por el lector. En primer lugar, el uso de cuatro Transmisores Base casi completos con modificaciones conduce a la redundancia de algunos bloques que pudiera ser vista como innecesaria. Por ejemplo, ¿por qué no incluir el bloque Selector antes del Polyphase Arbitrary Resampler y así utilizar una sola instancia del mismo en lugar de cuatro. Los motivos principales son dos. Por un lado, este diseño asegura que para cada modulación se puedan ajustar los parámetros

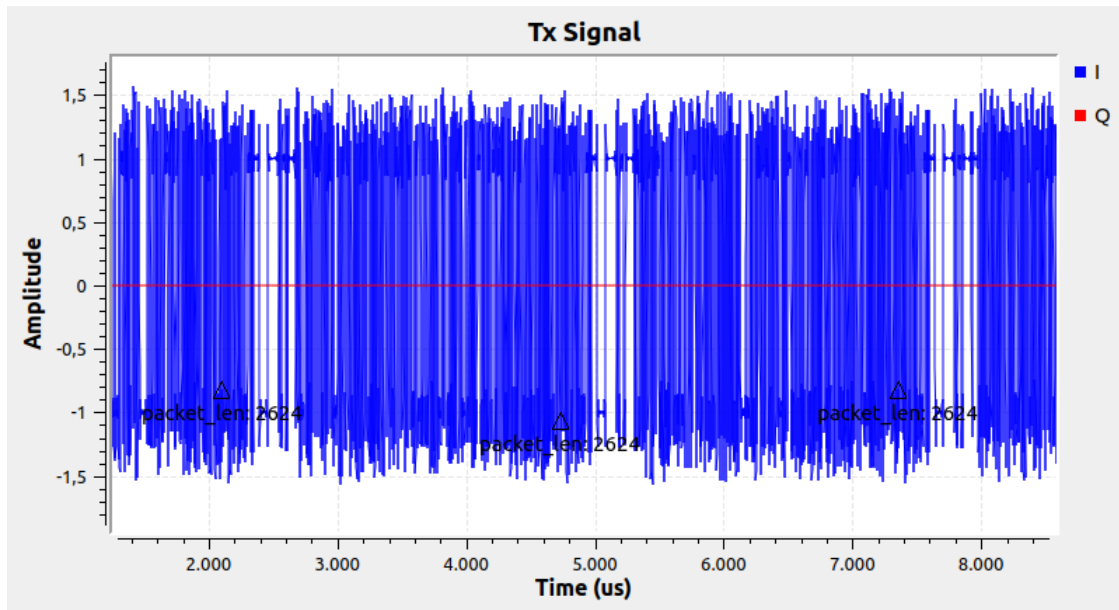


Figura 5.5.- Señal digital a transmitir enviada al USRP utilizando BPSK en la carga útil.

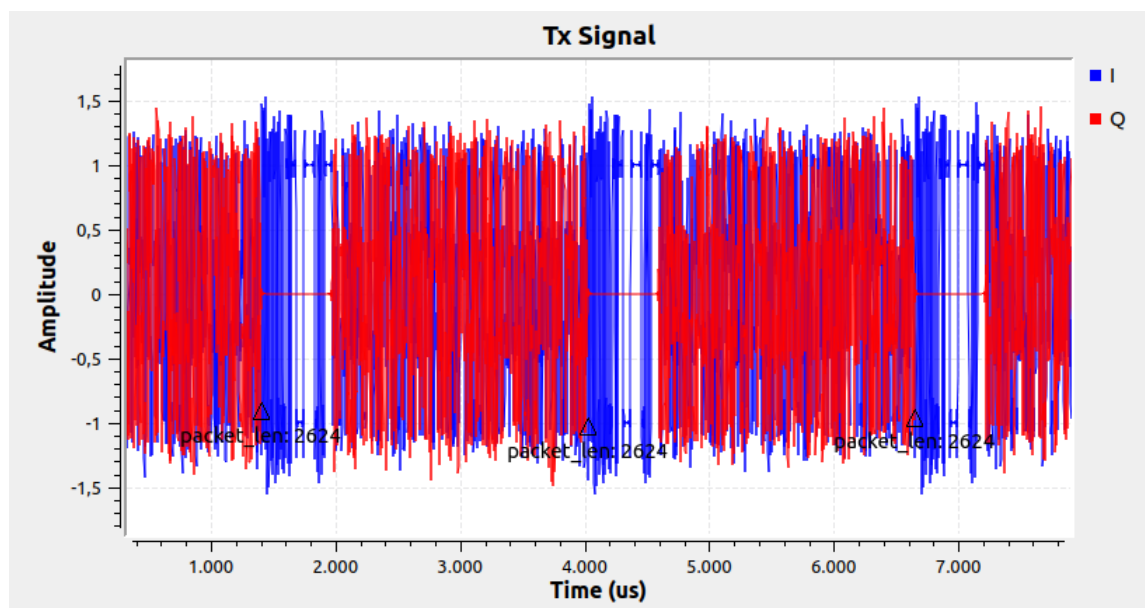


Figura 5.6.- Señal digital a transmitir enviada al USRP utilizando 16-QAM en la carga útil.



de transmisión de manera independiente, por ejemplo, el factor de roll-off,  $\alpha$ , de la función raíz de coseno alzado. Por otro, el incluir el Selector lo más próximo al bloque UHD: USRP Sink es lo más correcto desde el punto de vista de la gestión de buffers de GNU Radio cuando existen varias ramas de bloques en paralelo. La explicación detallada se escapa del ámbito de este TFM, aunque para el lector interesado en [30] se presenta una primera introducción al tema. Respecto al coste computacional, este no sería un problema, pues idealmente solo una de las cuatro ramas se encontraría activa, es decir, con sus bloques procesando información en cada momento. No obstante, esto no es así para el caso concreto aquí presentado, lo que enlaza con una segunda aclaración necesaria, expuesta en el siguiente párrafo.

Si se observa el inicio del esquema, la Fuente de Información (Figura 2.1) se ha implementado con cuatro bloques Random Source. El motivo es la simplicidad, pues el contenido de los bytes de la información transmitida no es el foco de interés de cara a la prueba del sistema. No obstante, en caso de utilizar este diseño como soporte de capa física para una aplicación particular, sería necesario la programación de un bloque en GNU Radio que gestione la comunicación entre capas de manera adecuada. La programación de dicho bloque sería distinta en función de la aplicación, por ejemplo, para una aplicación de transmisión de ficheros u otra de vídeo en streaming. El bloque resultante contaría con cuatro salidas conectadas a cada uno de los subsistemas y en cada momento escogería por cuál de ellas enviar su información, permaneciendo las otras tres inactivas. Para la sincronización con el bloque Selector en el cambio de modulación, bastaría como incluir una entrada de tipo mensaje en el bloque que estuviese también conectada al Tx\_Rx. El carácter asíncrono de la comunicación por mensajes, descrita en el Apartado 3.1.3, supondría la pérdida de un paquete de información en cada cambio, lo que se solucionaría con la programación de una retransmisión.

## 5.2.2.- Tx\_Rx

El esquema del Tx\_Rx en GNU Radio, generado con ayuda de GRC, se muestra en la Figura 5.7. Se trata de una implementación directa del Receptor Base descrito en el Apartado 4.2, con dos diferencias importantes. La primera, la recepción de la carga útil se configura para una BPSK, que es la única modulación empleada en el Canal de Retorno, según lo explicado en el Apartado 5.1. La segunda, el bloque que representa la Información de Destino (Figura 2.1), que en este caso se denota como Tx Controller. Se trata de un bloque programado durante el desarrollo de este TFM y, por ende, dentro del módulo gr-TFMv5. Su función es la de interpretar la información enviada por el Rx\_Tx (Apartado 5.3.2) y generar en su salida mensajes para su comunicación al Tx\_Tx. Cabe señalar que el Tx Controller solo interpretará los mensajes que hayan superado la detección de errores del código CRC32, de cara a evitar posibles fallos que conduzcan a un cambio de modulación no deseado.

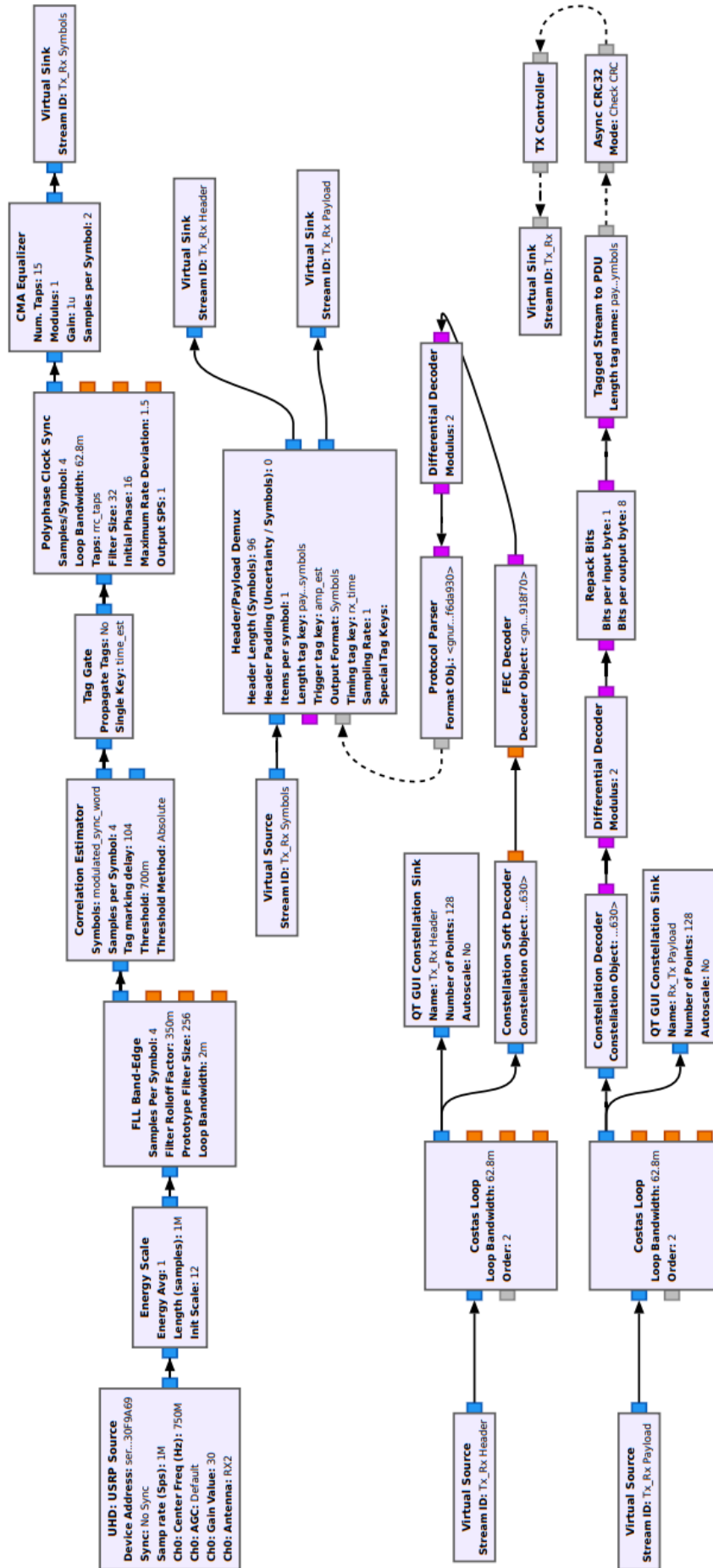


Figura 5.7.- (a) Esquema del Tx\_Rx en GNU Radio (GRC).



## 5.3.- RECEPTOR

En este apartado se presenta el Receptor, formado por el Rx\_Rx y el Rx\_Tx según lo mostrado en la Figura 5.1. Se explicarán ambos elementos por separado, así como la interacción entre ellos.

### 5.3.1.- Rx\_Rx

El esquema del Rx\_Rx en GNU Radio, generado con ayuda de GRC, se muestra en la Figura 5.8, la Figura 5.9 y la Figura 5.10. En su etapa inicial resulta idéntico al Receptor Base, explicado en el Apartado 4.2. La primera diferencia se muestra en el uso del bloque Header/Payload Demux AM<sup>12</sup>. Este bloque se ha programado expresamente para el diseño del sistema, con lo que se incluye en el módulo gr-TFMv5. Se basa en el bloque Header/Payload Demux del core de GNU Radio que se empleaba en el Receptor Base, modificándolo para incluir 4 salidas. Para cada cabecera recibida, el Header Payload Demux AM observará el campo BPS (Tabla 5.1) y decidirá por cual de sus salidas envía las muestras de la carga útil para ser demoduladas. Nótese que el campo BPS es inequívoco de la constelación empleada, pues su valor será de 1,2,3,4 y 6 para las BPSK, QPSK, 16-QAM y 64-QAM respectivamente. Además, este diseño permite utilizar la cadena de bloques más adecuada para la demodulación de cada una de ellas. Estas siguen el patrón del Receptor Base, con algunas diferencias. En primer lugar, para las constelaciones QAM se utiliza el igualador lineal de criterio LMS explicado en el Apartado 2.2.4. Esto se debe a que, como ya adelantó, el criterio CMA está diseñado a priori para modulaciones de amplitud constante, lo que no se cumple en el caso de 16-QAM y 64-QAM. Por otro lado, el bloque Constellation Receiver adapta el algoritmo del Costas Loop para su utilización de modulaciones QAM. El escalado digital mediante el bloque Multiply Const a la entrada de los igualadores en 16-QAM y 64-QAM sirve para compensar el escalado en transmisión de estas constelaciones (Apartado 5.2.1). Finalmente, se utiliza un único bloque CRC32 Check para la convergencia directa de las cuatro ramas. Estas no colisionan entre sí porque, gracias a la acción del Header Payload Demux AM, solo una de ellas se encontrará activa en cada momento.

---

<sup>12</sup>El sufijo AM indica Adaptive Modulation



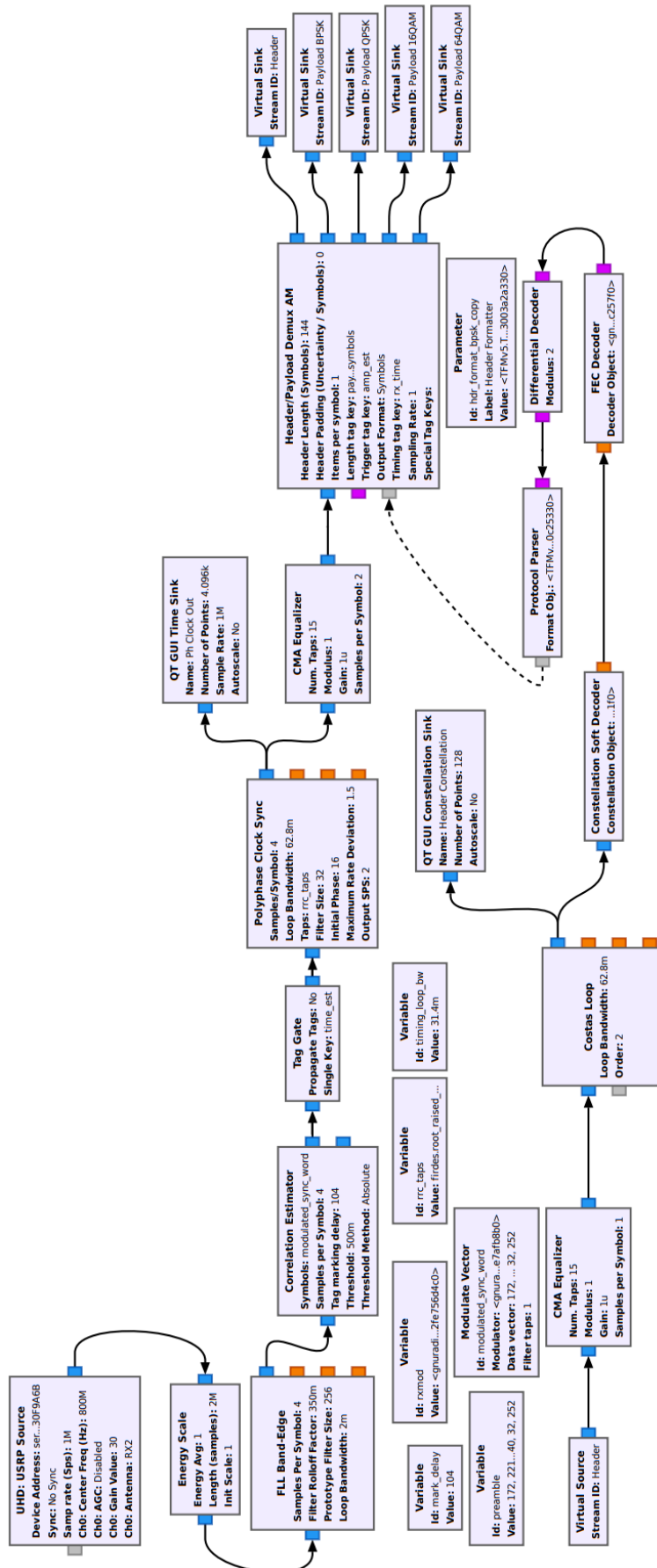


Figura 5.8.- (a) Esquema del Rx\_Rx en GNU Radio (GRC).

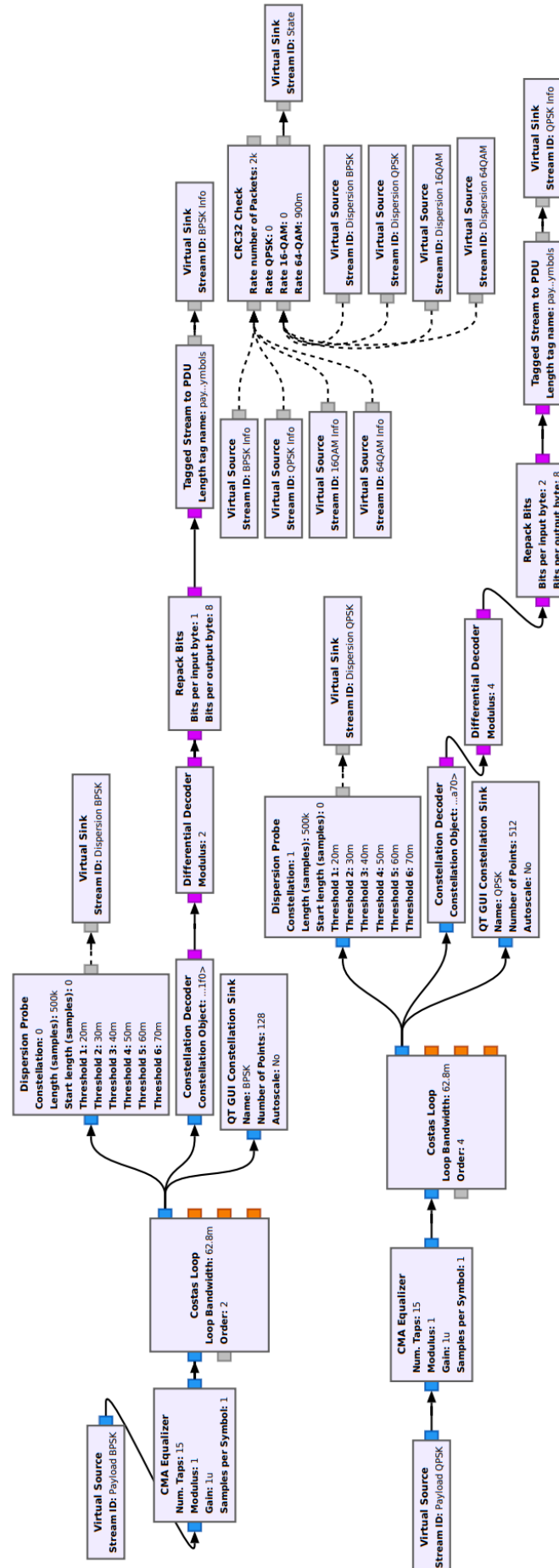


Figura 5.9.- (b) Esquema del Rx\_Rx en GNU Radio (GRC).

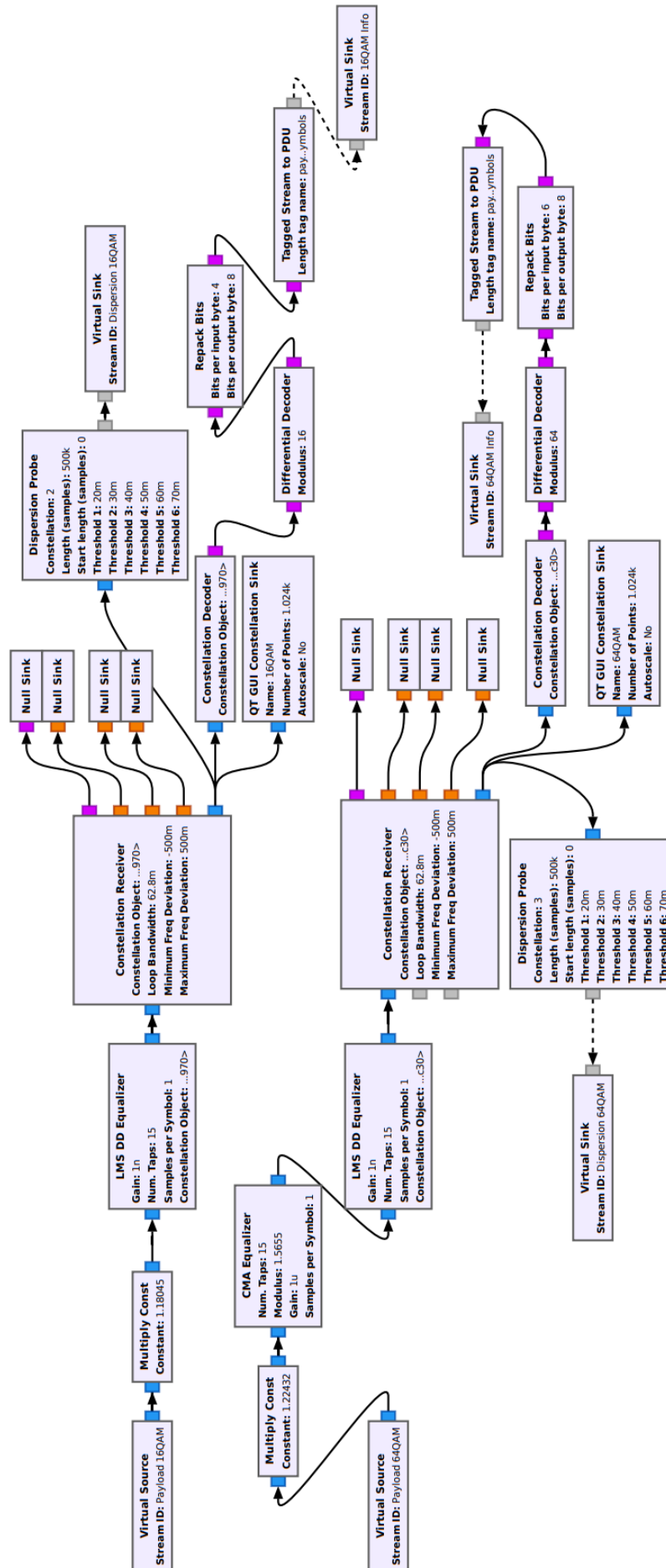
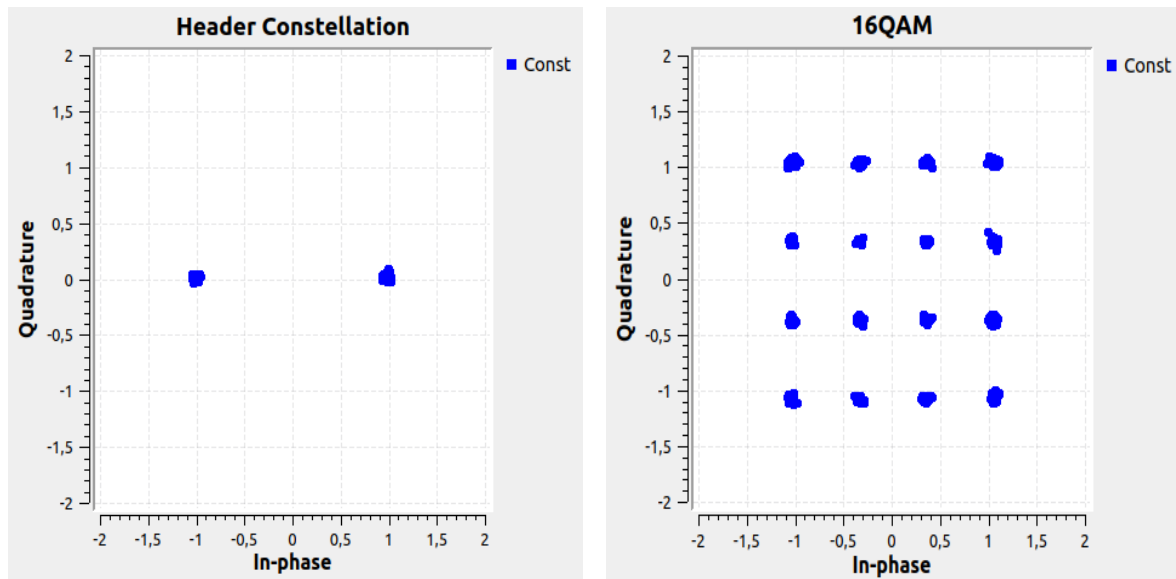


Figura 5.10.- (c) Esquema del Rx\_Rx en GNU Radio (GRC).



(a) Cabecera (BPSK).

(b) Carga útil (16-QAM).

Figura 5.11.- Ejemplo de constelaciones recibidas para la cabecera y carga útil.

La parte destinada a la igualación requiere de una explicación adicional. Nótese que el igualador CMA configurado para BPSK antes del Header Payload Demux AM se mantiene como en el Receptor Base. El motivo es evitar que la identificación del inicio de la cabecera en la muestra adecuada no se vea afectada en escenarios de multicamino, introducido en el Apartado 2.2.3. No obstante, esta igualación, que afecta a las muestras de todo el paquete, distará de ser óptima cuando la carga útil use las constelaciones 16-QAM y 64-QAM. Sin embargo, las pruebas realizadas, mostradas en el Apartado 6.2.3, demuestran que se cumple la función de favorecer la detección de la cabecera en todos los casos y que, además, no se perjudicaba la demodulación aguas abajo para las modulaciones 16-QAM y 64-QAM. Desde el punto de vista teórico este resultado es coherente, pues el hecho de que la primera igualación CMA sea lineal hace que su efecto pueda ser ajustado y/o corregido por otro igualador lineal como el LMS. No obstante, para el caso de 64-QAM se ha comprobado que el incluir un igualador CMA antes del LMS resultaba favorable. Este se configura con  $R_2 = 1.5655$ , de acuerdo con la aplicación de (2.62) a 64-QAM. En la Figura 5.11 se muestra un ejemplo de la constelación recibida para cabecera y carga útil tras la cadena de demodulación para el caso de emplear 16-QAM. En ella se puede observar que ambas constelaciones son idénticas a las mostradas en la Figura 2.4 y la Figura 2.6, salvo por una ligera dispersión de los símbolos. El escenario utilizado para este ejemplo consiste en la conexión de los dos USRP empleados mediante un cable BNC. Más detalles acerca del escenario se recogen en el Apartado 6.2.1.

Por último, el bloque Dispersion Probe sirve para medir la dispersión de las muestras de la constelación de la carga útil sobre las que se realiza la decisión. Este bloque se incluye en el módulo gr-TFMv5, desarrollado durante el presente TFM, y es el encargado de



medir la calidad de la comunicación. Como el Receptor no conoce el estado del canal en cada momento para poder calcular la probabilidad de error de manera exacta, una alternativa plausible es modelar todas las perturbaciones como ruido blanco gaussiano (Apartado 2.2.3), cuya varianza  $\sigma^2$  se estimará de manera empírica según:

$$\tilde{\sigma}^2 = \frac{\sum_{i=0}^{J-1} |r_i - a_i|^2}{J} \quad (5.1)$$

Siendo  $r_i$  la observación o muestra recibida,  $a_i$  el símbolo decidido, y  $J$ , el número de muestras promediadas. A partir de la relación de la probabilidad de error de símbolo,  $P_E$ , frente a  $\Gamma_b$  y las relaciones (2.14) y (2.45) es posible definir los límites correspondientes de varianza para obtener una  $P_b$  determinada, lo que se describirá en el Apartado 6.1.2. El bloque Dispersion Probe realiza el promedio recogido en (5.1) para un número de muestras,  $J$ , definido en la variable Length (samples). Además, se programa con unos valores límite de dispersión, denotados como Threshold o Th, para que, en caso de obtener valores mayores o menores a ellos según la situación, se envía un mensaje al bloque CRC32 Check, que es el encargado de comunicarse con el Rx\_Tx. Este, a su vez, mandará una petición de cambio de modulación al Transmisor. Por otro lado, para el correcto cálculo de la dispersión se debe indicar al bloque la constelación correspondiente, lo que se hace en la variable Constellation. La variable Start length (samples) permite definir un número de muestras a descartar durante en el arranque del bloque.

En la Figura 5.12 se muestran las transiciones entre los distintos estados en función del estado previo y la comparación de  $\tilde{\sigma}^2$  con los umbrales correspondientes en cada caso. En primer lugar, nótese que el cambio entre constelaciones se realiza siempre entre aquellas que tienen un valor de bits por símbolo,  $N$ , más próximo, lo que favorece la estabilidad del sistema. Iniciando siempre en BPSK, el avance del sistema hacia valores de  $N$  mayores se realiza de izquierda a derecha y siempre y cuando el valor de  $\tilde{\sigma}^2$  sea inferior al Threshold indicado. El retroceso a  $N$  inferiores se representa de derecha a izquierda y se da cuando la dispersión supera el umbral correspondiente. Por ejemplo, desde QPSK se puede progresar, en términos de tasa binaria, hacia 16-QAM si  $\tilde{\sigma}^2$  es inferior a  $Th_3$ . Por el contrario, se volverá a BPSK si  $\tilde{\sigma}^2$  supera  $Th_6$ . En caso de no cumplirse ninguna de las condiciones, es decir,  $Th_3 \leq \tilde{\sigma}^2 \leq Th_6$  se mantendrá QPSK.

Por otro lado, el hecho de emplear dos límites distintos para el cambio entre dos constelaciones, por ejemplo, con  $Th_5$  y  $Th_6$  para BPSK y QPSK, asegura un cierta histéresis del sistema que evita situaciones del canal para las que se transicione de manera continua entre dos modulaciones. Además, se debe cumplir

$$Th_1 < Th_2 < Th_3 < Th_4 < Th_5 < Th_6, \quad (5.2)$$



lo que implica que en un punto medio siempre se tenderá a emplear la modulación de menor  $N$ , lo que supone una estrategia conservadora de cara a la robustez del sistema. Por último, para evitar que variaciones bruscas del canal no persistentes en el tiempo afecten al cambio de modulación, este no se realizará hasta que  $\tilde{\sigma}^2$  cumpla la condición correspondiente al cambio durante dos promediados seguidos del bloque Dispersion Probe.

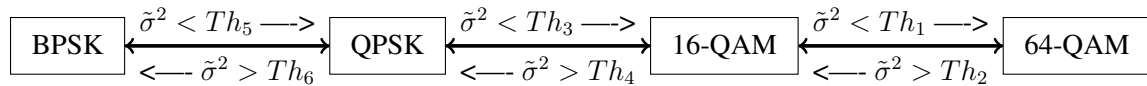


Figura 5.12.- Representación de las condiciones de cambio de modulación en la carga útil.

Para situaciones en las que la dispersión de los símbolos de la constelación aumente mucho, es posible que la estimación de dicha dispersión mediante (5.1) no sea precisa. El motivo es que el emparejamiento entre observaciones,  $r_i$ , y símbolos estimados,  $a_i$ , será incorrecto en un porcentaje no despreciable de casos. Para salvaguardar el sistema frente a esta situación, si la tasa de acierto de paquete, del inglés Packet Success Rate (PAS), disminuye por debajo de unos límites establecidos para cada modulación se realiza una petición de cambio de modulación. Estos límites se fijan en las variables Rate QPSK, Rate 16-QAM y Rate 64-QAM del bloque CRC32 Check. Este bloque recibirá en una entrada los bytes de la información para verificar el código CRC32, y, en otra, los mensajes del Dispersion Probe. Por defecto se mandarían siempre al Rx\_Tx los mensajes de cambio de modulación del Dispersion Probe, a no ser que la PSR baje por debajo de los límites establecidos.

### 5.3.2.- Rx\_Tx

El esquema del Rx\_Tx en GNU Radio, generado con ayuda de GRC, se muestra en la Figura 5.14, la Figura 5.15 y la Figura 5.16. Se trata de un esquema muy similar al del Tx\_Tx con algunas diferencias importantes. En primer lugar, todas las ramas están moduladas en BPSK, que es la única constelación empleada en el Canal de Retorno. La distinción entre flujos se debe en este caso al código incluido en la carga útil de los mismos, que indica la constelación que debe de usar el Transmisor. La estructura del paquete se muestra en la Tabla 5.2. Como se puede observar, en este caso se emplea la cabecera header\_format\_default del Transmisor Base (Apartado 4.1). En cuanto a la carga útil, se añaden 12 bytes de información aleatoria para asegurar la equiprobabilidad de los símbolos a transmitir, que es un supuesto con el que trabajan todas las cadenas receptoras presentadas en este TFM. Por último, se incluyen 2 bytes que indican petición de modulación al Transmisor, campo que será interpretado por el bloque Tx Controller. En la Figura 5.13 se muestra la señal digital a transmitir enviada al USRP cuando los 2 bytes del código de modulación se encuentran con todos sus bits a 1, lo que indica una petición de 64-QAM.



Paquete, 30 bytes, 960 muestras					
C, BPSK, (12), 384 muestras			CU, BPSK, (18), 576 muestras		
CA (8)	LCU (2)	LCU (2)	Random Info (12)	CM (2)	CRC32 (4)

Tabla 5.2.- Estructura de los paquetes generados por el Rx\_Tx. La leyenda de campos es la siguiente: C - Cabecera; CU - Carga Útil; CA - Código de Acceso; LCU - Longitud de la Carga Útil; CM - Código de Modulación.

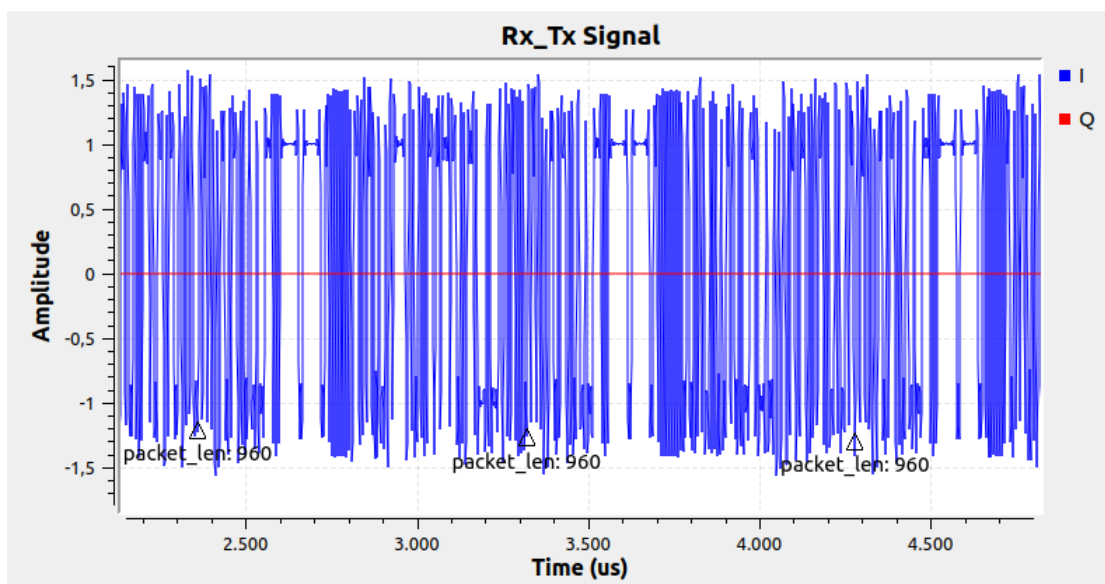


Figura 5.13.- Señal digital a transmitir enviada al USRP para una petición al Transmisor de 64-QAM.

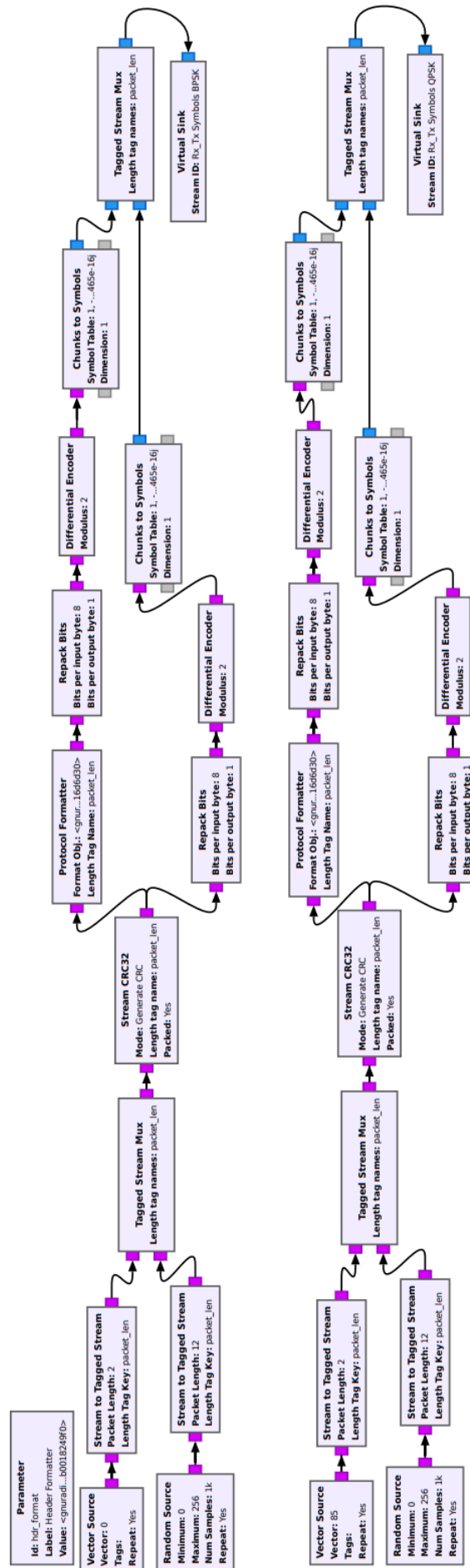


Figura 5.14.- (a) Esquema del Rx\_Tx en GNU Radio (GRC).



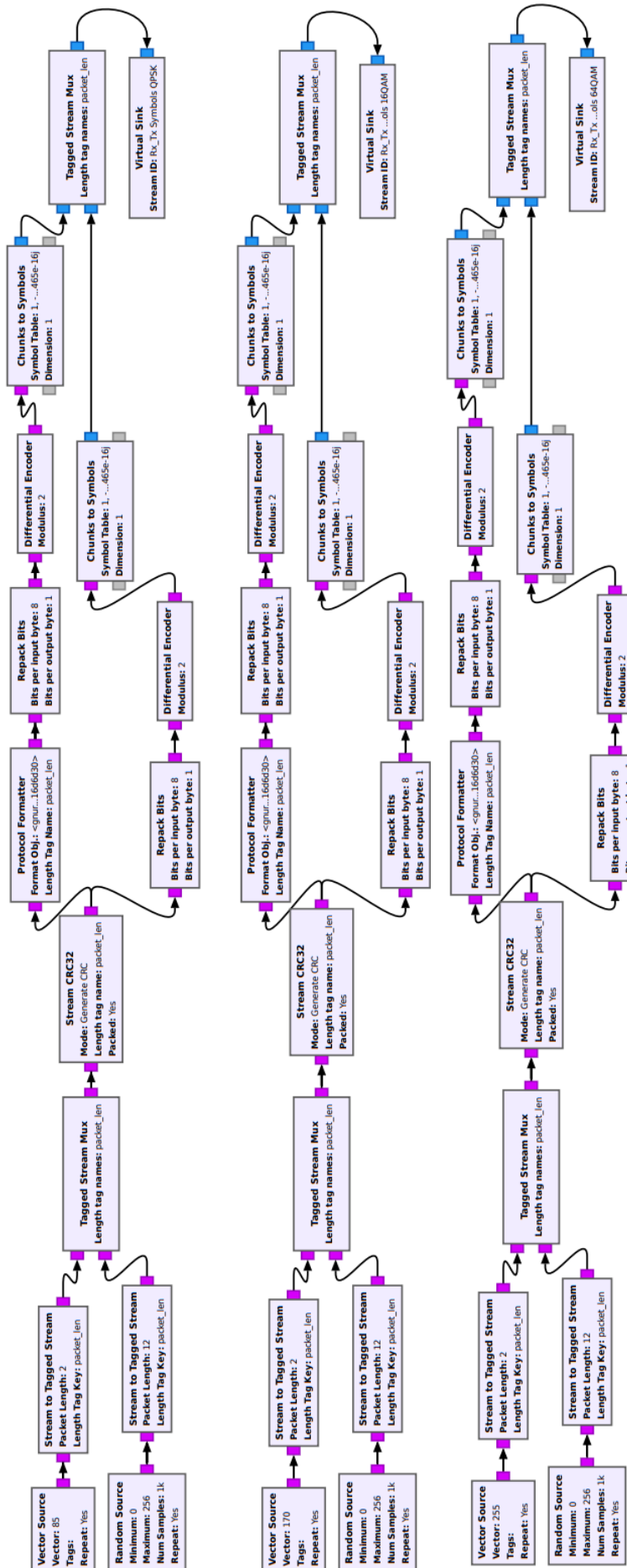


Figura 5.15.- (b) Esquema del Rx\_Tx en GNU Radio (GRC).

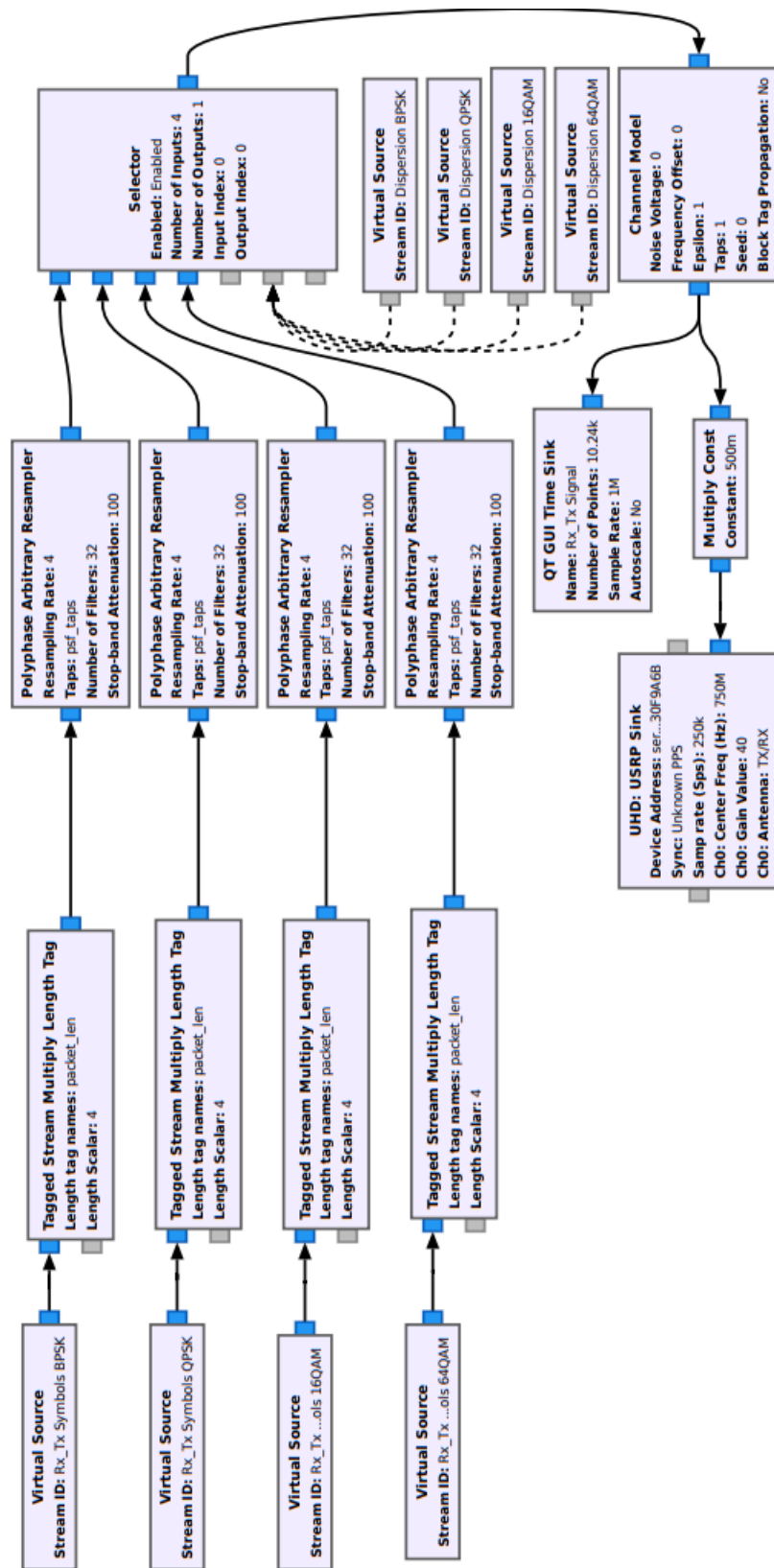


Figura 5.16.- (c) Esquema del Rx\_Tx en GNU Radio (GRC).



## 6. MEDIDAS Y RESULTADOS

En este capítulo se describen las medidas realizadas para el sistema presentado en el Capítulo 5, así como los resultados obtenidos a partir de ellas. A tal fin, se comenzará describiendo aspectos generales a tener en cuenta durante las campañas de medida, para posteriormente explicar la metodología y valores obtenidos en ellas.

### 6.1.- ASPECTOS GENERALES

En este apartado se presentan aspectos generales importantes que han de ser tenidos en cuenta durante las medidas del sistema.

#### 6.1.1.- CONFIGURACIÓN DE LOS USRP

Como se describe en el Apartado 3.2, la implementación física del sistema se ha realizado mediante dos dispositivos USRP modelo B200. Se recuerda, no obstante, que, gracias a la compatibilidad entre los USRP, los esquemas diseñados en GNU Radio presentados hasta ahora funcionarían sin ninguna modificación para cualquier otro modelo de esta gama, eso sí, teniendo siempre en cuenta las prestaciones de cada dispositivo concreto.

Uno de los USRP B200 actúa como la parte hardware del Transmisor mostrado en el esquema de la Figura 5.1, mientras que otro hace lo propio con el Receptor. El hecho de ser un dispositivo full-dúplex permite que el primero implemente simultáneamente el Tx\_Tx y el Tx\_Rx, y, el segundo, el Rx\_Rx y el Rx\_Tx. En la Tabla 6.1 se muestra un resumen de las distintas configuraciones. Además, es importante señalar que en todos los bloques de tipo UHD:USRP Source se han deshabilitado las opciones DC Offset Correction e IQ Imbalance Coercion. Nótese que, como se ha adelantado en el Apartado 5.1, el Canal de Retorno emplea una tasa de muestreo más baja que el Canal Principal, además de una frecuencia de canal distinta.

Subsistema	Esquema	Bloque	Puerto	Ganancia	$f_c$	Tasa de muestreo
Tx_Tx	Figura 5.4	Sink	TX/RX	40 dB	800 MHz	1 MS/s
Tx_Rx	Figura 5.7	Source	RX2	30 dB	750 MHz	250 kS/s
Rx_Rx	Figura 5.8	Source	RX2	30 dB	800 MHz	1 MS/s
Rx_Tx	Figura 5.16	Sink	TX/RX	30 dB	750 MHz	250 kS/s

Tabla 6.1.- Resumen de la configuración de los USRP.

Por comodidad, se ha realizado siempre el control de los USRP y esquemas de GNU Radio desde un mismo PC portátil de sistema operativo Ubuntu 20.04 LTS. Como es obvio,



en un caso real el Transmisor y el Receptor estarían separados físicamente y conectados a dos unidades de procesamiento distintas. No obstante, los resultados obtenidos habrían sido idénticos, pues en ningún caso se ha aprovechado la ejecución del Transmisor y Receptor desde la misma unidad para una comunicación entre ellos a parte de la que realizan a través del canal físico. Por otro lado, cabe señalar que la conexión con el portátil se ha realizado por medio de interfaces USB 2.0, lo que limita enormemente la tasa de muestreo máxima de los USRP [21]. Sin embargo, esto no supone un problema para los valores aquí utilizados.

Por último, y aunque son parámetros configurables desde GNU Radio, para las medidas realizadas se ha empleado siempre un tasa de símbolo de 250 kb, lo que equivale a un período de símbolo,  $T$ , de 4 ms, además de un factor de roll-off,  $\alpha$ , de 0.35. Por tanto, de acuerdo con (2.9) esto implica un ancho de banda de 168.75 kHz.

## 6.1.2.- PROBABILIDAD DE ERROR Y BER

Según lo explicado en el Apartado 5.1, para el funcionamiento del sistema se define una tasa de error de bit en la carga útil denotada por  $BER_{max}$ . Además, como se explicó en el Apartado 5.3.1, la estimación de  $P_b$  en cada momento se realiza en base a la medida de la dispersión de los símbolos recibidos,  $\tilde{\sigma}^2$ . El cálculo se basa en la relación de la probabilidad de error de símbolo,  $P_E$ , con  $\Gamma_b$  (2.15), mostrada en el Apartado 2.2; y en modelar todas las perturbaciones del canal como ruido blanco gaussiano. En la Figura 6.1 se recogen las gráficas resultantes de  $P_b$  frente a  $\Gamma_b$ , asumiendo Codificación Gray y codificación diferencial en todos los casos<sup>13</sup>. Nótese que las gráficas de BPSK y QPSK se encuentran superpuestas.

Sin embargo, estas gráficas no resultan las más intuitivas de cara a la interpretación del funcionamiento del sistema. De acuerdo con el Apartado 4.2 y el Apartado 5.1, los esquemas diseñados en GNU Radio se aseguran de que la energía media de las muestras digitales,  $\mathcal{E}_{av}$  sea constante e igual a 1. Esto implica que, según (2.11) y (2.13), la energía media de bit,  $\mathcal{E}_b$ , será de 0.5, 0.25, 0.125 y 1/12 para las constelaciones BPSK, QPSK, 16-QAM y 64-QAM respectivamente. Se recuerda que los valores calculados sobre las muestras digitales son adimensionales, pues la relación de los mismos con la energía en unidades físicas dependerá de la configuración del USRP. Sin embargo, desde el punto de vista del transmisor, si no se modifica la ganancia, una  $\mathcal{E}_{av}$  constante en las muestras digitales enviadas al USRP implicará una  $\mathcal{E}_{av}$  constante en Julios de la señal transmitida. Desde el punto de vista de la recepción, es interesante también trabajar con los valores digitales, pues la estimación  $\tilde{\sigma}^2$  se realiza sobre estas muestras, lo que se puede entender como una normalización de la medida. Esto implica que, por ejemplo, si se emplea una BPSK y  $\tilde{\sigma}^2 = 0.5$  se tendrá en la estimación que  $\Gamma_b \approx 3$  dB, según (2.14) y (2.45).

El valor de  $\mathcal{E}_b$  está asociado a la constelación empleada, mientras que, para un canal

<sup>13</sup>Para las modulaciones QAM el empleo de codificación diferencial tiene el mismo efecto sobre  $P_b$  que para PSK (2.36).

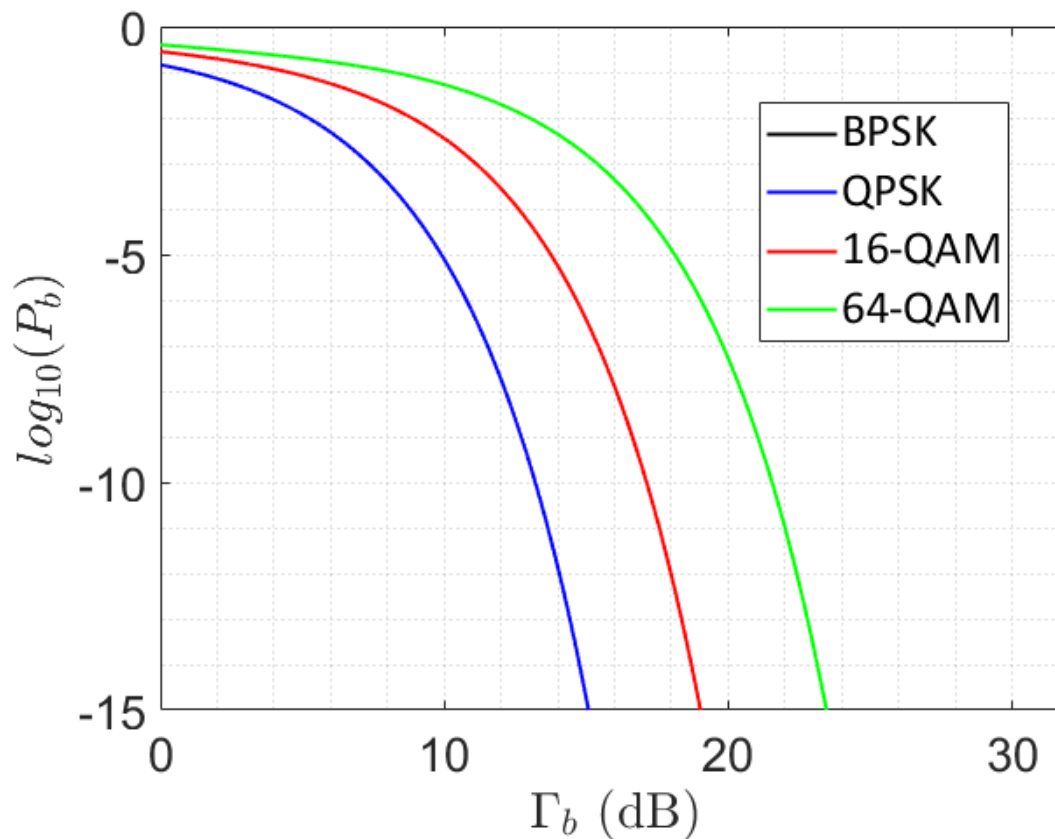


Figura 6.1.- Probabilidad de error de bit frente a  $\Gamma_b$  para las modulaciones empleadas.

dato,  $\tilde{\sigma}^2$  será prácticamente independiente de esta. Teniendo todo lo expuesto hasta ahora en cuenta, se pueden adaptar las gráficas de la Figura 6.1 para obtener  $P_b$  de forma directa a partir de  $\sigma^2$  normalizada para el caso concreto del sistema presentado, lo que se muestra en la Figura 6.2. Conviene recalcar que las gráficas de la Figura 6.1 y Figura 6.2 se corresponden con los valores teóricos, los cuales serían exactos si se estuviese trabajando con un canal perfectamente gaussiano de varianza de ruido conocida. Como esto no es así, y lo que se conoce es una medida de la dispersión de las muestras,  $\tilde{\sigma}^2$ , lo que se obtiene con ello es una estimación de la probabilidad de error, denotada a partir de ahora como  $\tilde{P}_b$ .

Si se quiere verificar que  $\tilde{P}_b$  se corresponde con un valor fiable se debe realizar de algún modo una medida de la BER. Para ello, el Receptor puede comprobar la tasa de paquetes recibidos correctamente, es decir, que su carga útil verifica el código CRC32 para la detección de errores. Esta tarea es realizada por el bloque CRC32 Check, explicado en el Apartado 4.2. Este método cuenta también con la ventaja de que puede ser implementado por el Receptor sin necesidad de información adicional, ni de conocer de antemano los bytes enviados. Sin embargo, dos cosas deben de ser tenidas en cuenta. La primera, que el método será fiable siempre y cuando la probabilidad de error no sea muy elevada, ya que si la cabecera, modulada en BPSK, no se recibe correctamente, se perderá el paquete por completo

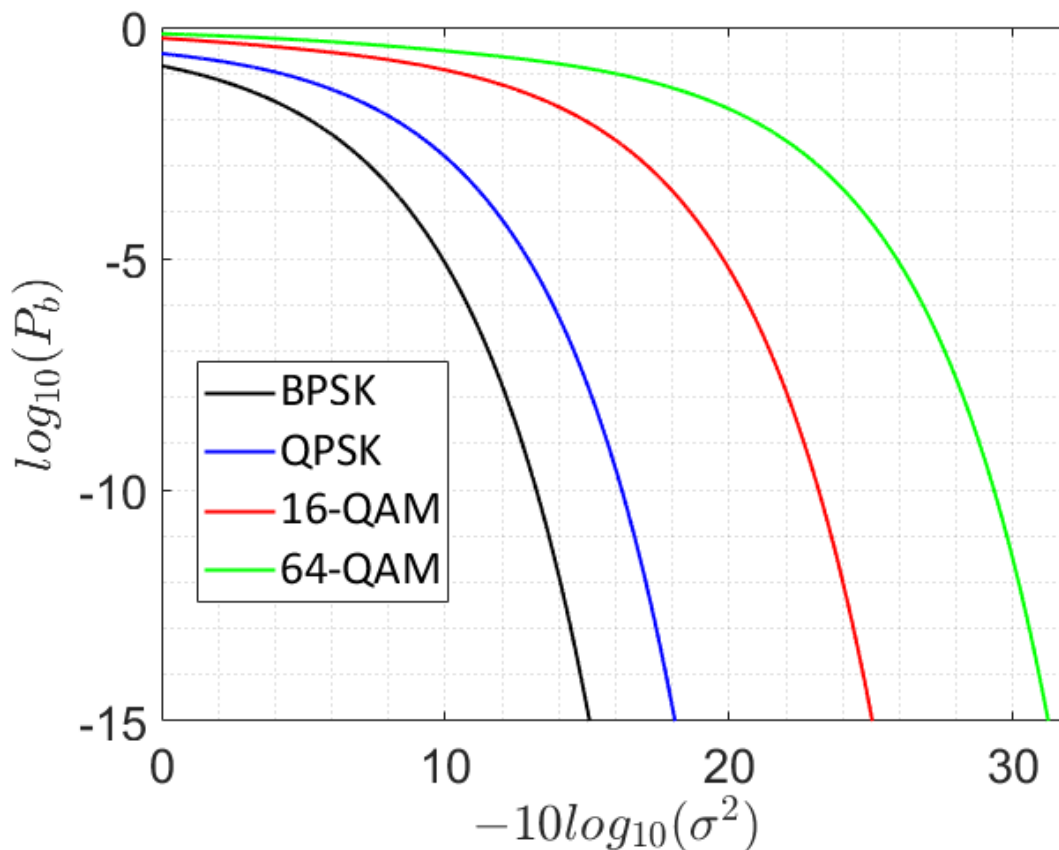


Figura 6.2.- Probabilidad de error de bit frente a  $\sigma^2$  normalizada para las modulaciones empleadas.

y no se llegará a demodular la carga útil. La segunda, que existe la posibilidad de fallo del código, es decir, que un paquete con errores sea considerado correcto. El código CRC32 [13] tiene la capacidad de detectar los siguientes errores en la carga útil:

- Errores simples: todos los paquetes que contengan un único bit erróneo.
- Errores dobles: todos los paquetes que contengan dos bits erróneos.
- Errores impares: todos los paquetes que contengan un número impar de bits erróneos.
- Ráfagas de errores de longitud igual o menor a 32: todos los paquetes que contengan un número cualquiera de errores cuya mínima distancia entre ellos sea igual o menor a 32. Se entiende por distancia el número de posiciones entre los bits erróneos.

Además, en los casos en los que la longitud de la carga útil sea menor a 512 bits, también se detectarán los errores cuádruples. El conjunto de estas condiciones permite despreciar la posibilidad de fallo del código incluso para el peor de los casos, este es, 64-QAM, donde la longitud de la información en la carga útil es máxima (3072 bits, Tabla 5.1).



La medida de la BER por esta vía se realiza de la siguiente forma. Si se mide la tasa de acierto de paquete o PSR utilizando un número de paquetes suficientemente elevado, se obtiene una estimación de la probabilidad de acierto de paquete,  $P_{ap}$ . Dado que la probabilidad de detectar correctamente un paquete se corresponde con la probabilidad de detectar correctamente todos sus bits y suponiendo que cada paquete de datos contiene  $L$  bits, se obtiene la relación

$$P_{ap} = (1 - P_b)^L, \quad (6.1)$$

de donde tomando logaritmos se obtiene:

$$P_b = 1 - 10^{\log_{10}(P_{ap})/L} \quad (6.2)$$

Cabe señalar que (6.1) y (6.2) trabajan con las expresiones teóricas de la probabilidad de error y la probabilidad de acierto de paquete. Sin embargo, la información de la que se dispone es una medida de la PSR, que se relaciona con una medida de la BER a través de las expresiones expuestas.

Como se adelantó en el Apartado 5.3.1, el bloque CRC32 Check utiliza  $\tilde{P}_b$  y la medida de PSR para generar las peticiones de cambio de modulación. El empleo de  $\tilde{P}_b$  es más seguro cuando el límite de la probabilidad de error requerida es muy bajo. En estos casos, de utilizar PSR podría ser demasiado tarde cuando se empezaran a detectar paquetes erróneos tras una degradación del canal, habiendo ya superado la  $BER_{max}$ . Por el contrario, cuando la probabilidad de error es elevada y las perturbaciones del canal alejan su comportamiento de un canal gaussiano,  $\tilde{P}_b$  pierde fiabilidad. Lo que es más, una dispersión baja de los símbolos no asegura una recepción correcta de la información si, por ejemplo, no se ha incluido una codificación diferencial que corrija ambigüedades de fase o las regiones de decisión programadas de los símbolos no son las adecuadas. En base todo a lo explicado, la política será siempre la de basar los cambios de modulación en  $\tilde{P}_b$ , salvo cuando la tasa de acierto de paquete sea menor a los límites fijados en CRC32 Check para cada modulación y durante dos medidas consecutivas. Esta configuración favorece el correcto funcionamiento del sistema para probabilidades de error bajas y, a su vez, asegura que la tasa de errores en la información recibida este realmente por debajo del límite establecido en todo momento.

## 6.2.- CANAL CONTROLADO POR SOFTWARE

En este apartado se describe la campaña de medidas principal del sistema, realizada por medio de un canal controlado digitalmente. Se presentan aquí los resultados obtenidos, así como las conclusiones alcanzadas en base a estos.



## 6.2.1.- INTRODUCCIÓN

Durante la campaña de medidas se utiliza el bloque de GNU Radio Channel Model para reproducir las perturbaciones típicas que afectan a un sistemas de comunicaciones móviles, las cuales se describen en el Apartado 2.3. El bloque se incluye tanto en el Tx\_Tx (Apartado 5.2.1), como en el Tx\_Rx (Apartado 5.3.2), y se considerará siempre que el Canal Principal y el Canal de Retorno son idénticos, con lo que ambas instancias serán programadas con los mismos parámetros. Usando como ejemplo el Tx\_Tx, si se observa la Figura 5.4 se comprueba que la actuación se realiza sobre las muestras digitales justo antes de ser enviadas al USRP. Esto implica que las perturbaciones añadidas formarán parte de la señal eléctrica generada por el dispositivo, la cual se propagará por el canal físico. Tanto para el Canal Principal como para el Canal de Retorno se utilizarán como canal físico sendos cables de tipo BNC con transiciones SMA, las cuales permiten la conexión con los puertos de los USRP. Estos se configuran de acuerdo con la Tabla 6.1. El montaje final se muestra en la Figura 6.3.

En realidad, el canal físico empleado no se compone solo de los cables y transiciones descritos, sino que incluye la cadena transmisora y receptora de los elementos hardware de los USRP. Esto implica que, incluso cuando no se utiliza el Channel Model, existen una serie de perturbaciones que hay que tener en cuenta. Estos son principalmente tres: el retardo de la señal eléctrica, la deriva y el ruido de fase de los osciladores, y el hecho de que el reloj del Transmisor y Receptor no están sincronizados. Todas estas perturbaciones y su solución se comentan en el Apartado 4.2, correspondiente al diseño del Receptor Base.



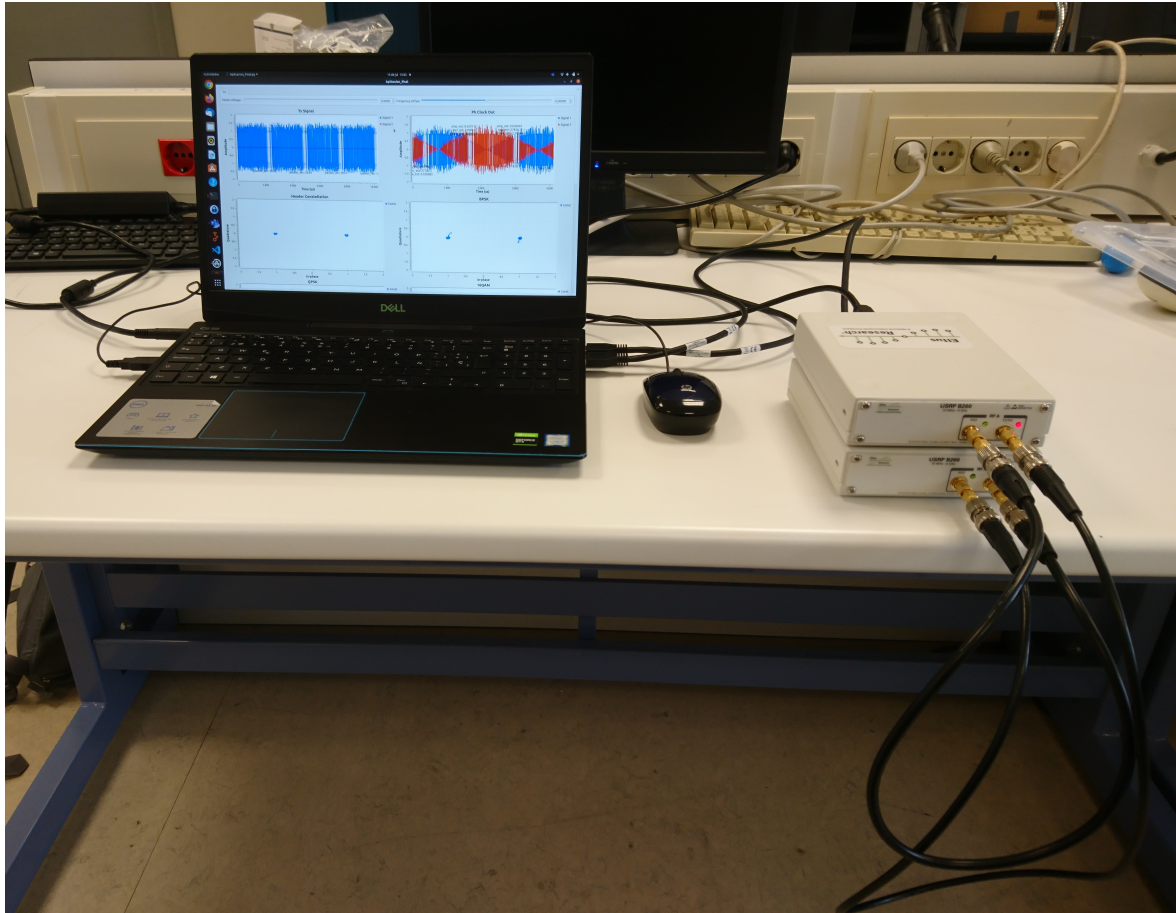


Figura 6.3.- Montaje para las medidas del canal controlado por software.

La primera prueba realizada, antes de utilizar el Channel Model, es la de asegurarse que la condición de potencia media constante descrita en el Apartado 5.1 se cumple. Para esta comprobación se utilizó un analizador de espectros y se midió el valor para las cuatro modulaciones posibles de la carga útil. En la Figura 6.4 y la Figura 6.5 se muestra el resultado para dos de las posibilidades: BPSK y 64-QAM respectivamente.



Figura 6.4.- Medida de la densidad espectral de potencia de la señal transmitida por el Tx\_Tx utilizando BPSK en la carga útil.



Figura 6.5.- Medida de la densidad espectral de potencia de la señal transmitida por el Tx\_Tx utilizando 64-QAM en la carga útil.

## 6.2.2.- CANAL GAUSSIANO

Con el objetivo de comprobar el funcionamiento del sistema para distintos valores de varianza de ruido,  $\sigma^2$ , se emplea el parámetro Noise Voltage del Channel Model. Este permite añadir ruido gaussiano de varianza ajustable sobre la señal digital a transmitir. Según la definición del bloque en GNU Radio, la relación de  $\sigma^2$  con el valor introducido en Noise Voltage, denotado por  $V_n$ , viene dada por:

$$\sigma^2 = \frac{V_n^2}{k} \quad (6.3)$$

, donde  $k$  designa el número de muestras por símbolo, que es de cuatro para el sistema presentado. Como se explicó en el Apartado 6.1, el valor de  $\sigma^2$  se encuentra normalizado respecto de una energía media de símbolo,  $\mathcal{E}_{av}$ , de valor igual a 1.

En este caso las medidas se realizan de forma independiente para cada modulación, sin utilizar el Canal de Retorno para el cambio de la misma. La Figura 6.6 muestra el resultado de las medidas de la BER para las modulaciones analizadas. Todas las medidas se obtuvieron a partir de la comprobación en el receptor de  $10^5$  paquetes consecutivos.

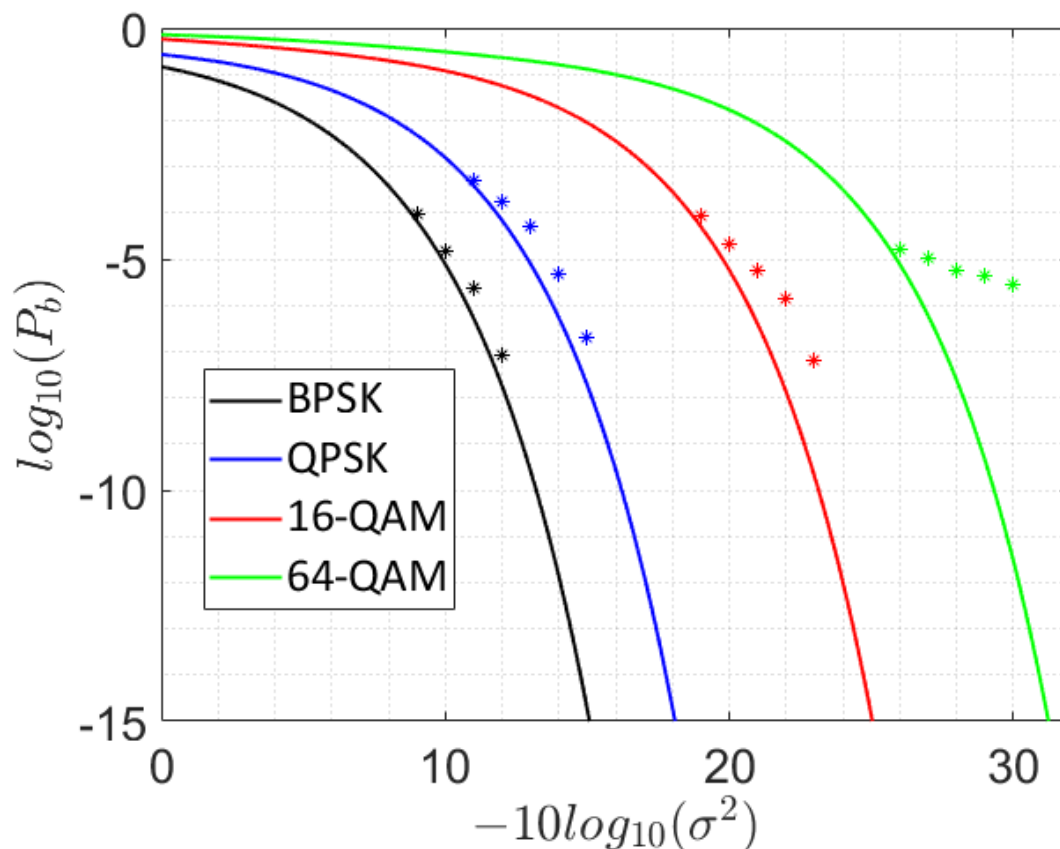


Figura 6.6.-  $P_b$  teórica y BER medida en función de  $\sigma^2$  en canal gaussiano para las distintas modulaciones empleadas en la carga útil. Resultados teóricos y experimentales representados mediante línea continua y asteriscos, respectivamente.

Para las modulaciones BPSK, QPSK y 16-QAM, se observa como las medidas de la BER se ajustan a la  $P_b$  teórica de forma notable, si bien con valores siempre ligeramente superiores. Este resultado es el esperado, pues la dispersión de las muestras en el receptor será siempre ligeramente superior a  $\sigma^2$ , ya que al ruido añadido digitalmente hay que añadir el ruido del canal físico, formado por los USRP y los cables empleados. Lo que es más importante, los bloques digitales adaptativos del receptor añaden también cierta dispersión en las muestras. Nótese que la diferencia entre BER y  $P_b$  es mayor cuanto menor es el valor de  $\sigma^2$ , lo que también resulta coherente, pues en estos casos las contribuciones a la dispersión



total del canal físico y la cadena receptora empiezan a ser del mismo orden que la del ruido digital añadido en transmisión. Este efecto resulta claramente visible para 64-QAM, donde  $\sigma^2$  toma sus valores más bajos, pero el sistema se ve limitado por la dispersión introducida por los bloques digitales del receptor.

## 6.2.3.- CANAL LINEAL

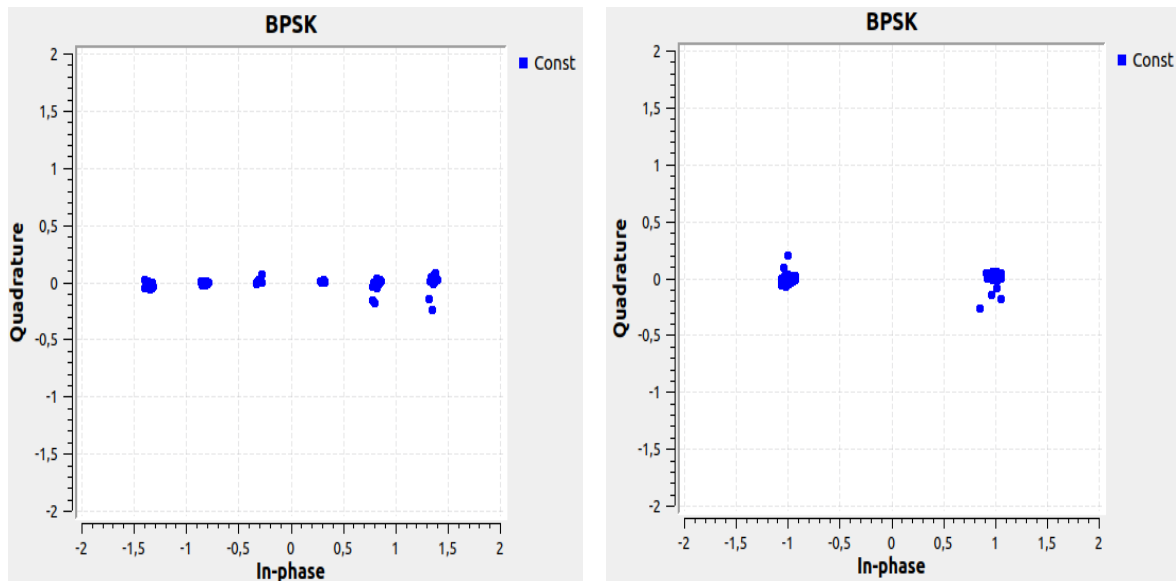
El siguiente paso consiste en comprobar la capacidad de adaptación de las distintas cadenas receptoras frente a un canal lineal con perturbaciones propias de un canal de comunicaciones móviles, ya sea invariante o variante en el tiempo. El bloque Channel Model permite añadir, además de ruido gaussiano, un canal multicamino y un desplazamiento en frecuencia variable propio del efecto Doppler (Apartado 2.3.2).

El multicamino se añade con un filtro FIR digital cuyas componentes se introducen en la variable Taps. Supóngase, por ejemplo, que tomando como referencia (2.43) se define un canal lineal invariante (Apartado 2.3.1) que venga dado por (6.4). Para ello, como se utilizan cuatro muestras por símbolo, se configura la variable Taps con el vector [0.9, 0, 0, 0, 0.3, 0, 0, 0, 0.312]. El hecho de que la suma del módulo de las componentes del canal al cuadrado es igual a 1 permite mantener la potencia transmitida constante.

$$\bar{h}_c(t) = \sum_{n=1}^{N=3} a_n e^{j\phi_n} \delta(t - t_n) \left\{ \begin{array}{l} a_1 = 0.9, \phi_1 = 0 \\ a_2 = 0.3, \phi_2 = 0 \\ a_3 = 0.312, \phi_3 = 0 \\ t_n = \frac{n}{f_s} \end{array} \right. \quad (6.4)$$

Respecto al efecto Doppler, este se simula añadiendo un desplazamiento en frecuencia configurable en la variable Frequency Offset. El valor introducido debe ser normalizado respecto la tasa de símbolo. Por ejemplo, si se quiere introducir un desplazamiento de 1.250 kHz, como la tasa de símbolo empleada es de 250 kb, el valor de Frequency Offset ha de ser de 0.005. De acuerdo con (2.46) esto equivaldría al desplazamiento en frecuencia provocado por el efecto Doppler dada una velocidad de 468.75 m/s con transmisor y receptor alineados y una frecuencia de 800 MHz; o 62.5 m/s para las mismas condiciones y una frecuencia de 6 GHz.

Para comprobar la capacidad de adaptación de las cadenas receptoras, se configuró el Channel Model para inicialmente reproducir el canal lineal invariante mostrado en (6.4). Seguidamente, se inició el sistema y, una vez puesto en funcionamiento, se procedió a modificar en tiempo real el desplazamiento en frecuencia para valores de Frequency Offset entre -0.005 y 0.005; así como la varianza de ruido,  $\sigma^2$ , dentro de los valores considerados razonables de acuerdo con lo expuesto en el Apartado 6.2.2. Dicha modificación en tiempo real es posible gracias al panel de control definido en la variable QT GUI Widget (Figura 5.4). Esta permite que el canal lineal invariante definido por defecto pase a ser un canal variante



(a) Antes de la convergencia.

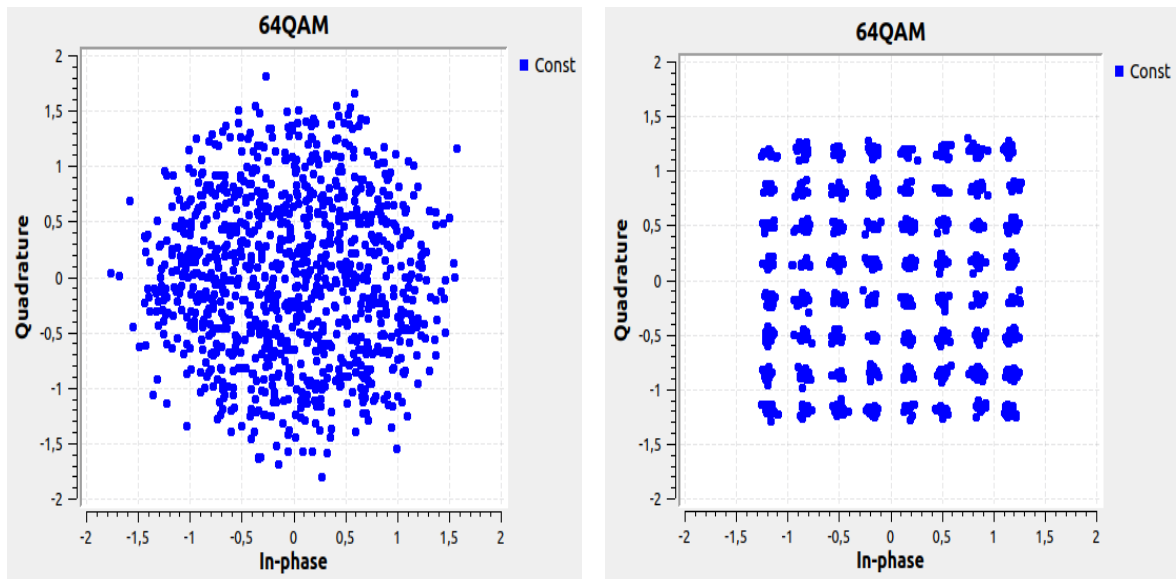
(b) Después de la convergencia.

Figura 6.7.- Constelación de la carga útil para BPSK antes y después de la convergencia de los bloques digitales de la cadena receptora para un canal multicamino.

en el tiempo. Este procedimiento se repitió para las distintas modulaciones de la carga útil, realizando el cambio entre ellas de forma manual y no automática. En todos los casos se observó una correcta convergencia de los bloques encargados de corregir las perturbaciones introducidas, a saber, los igualadores CMA y LMS para la corrección del multicamino; y el bloque FLL Band-Edge y el Costas Loop para el efecto Doppler. Esta convergencia tiene lugar siempre en unos pocos segundos, pudiendo ser acelerada en caso de utilizar una mayor tasa de símbolo. En la Figura 6.7 se observa la comparación de la constelación recibida en la carga útil empleando BPSK antes y después de la convergencia para un valor  $\sigma^2$  igual a cero y el escenario multicamino presentado en (6.4). En la Figura 6.11 se representa lo propio para 64-QAM, pero esta vez con un valor de Frequency Offset de 0.005.

Adicionalmente, el bloque Energy Scale (Apartado 4.2) permite que las modulaciones QAM sean robustas frente a desvanecimientos selectivos en potencia. En la Figura 6.9 se observa la convergencia de la constelación 16-QAM de la carga útil por actuación de este y los demás bloques tras una reducción súbita en tiempo real de la ganancia de transmisión en 3 dB.

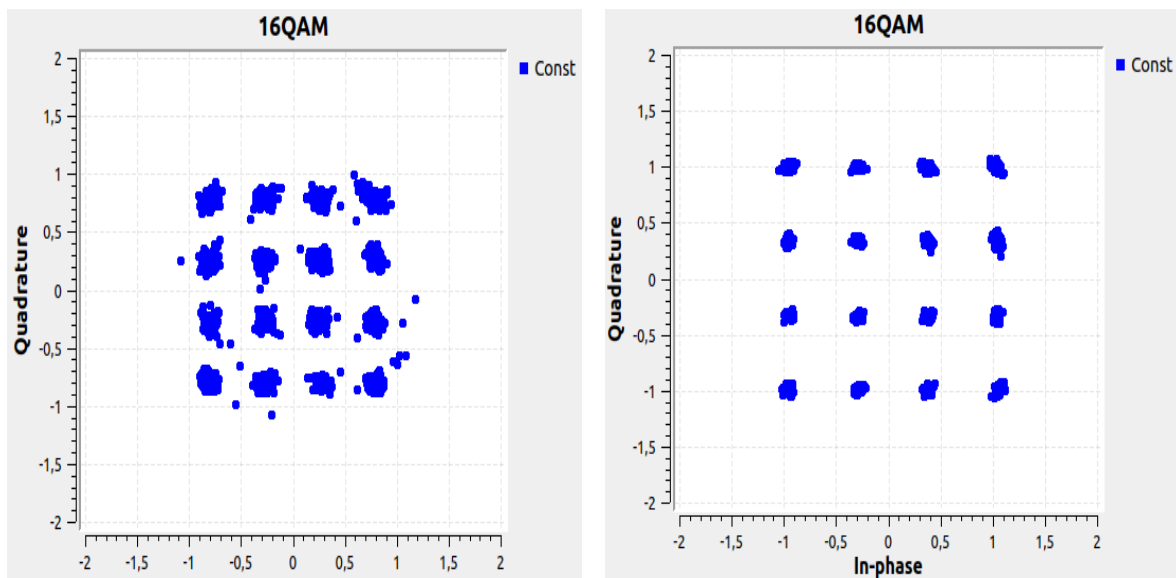
Por último, con el objetivo de cuantificar el efecto de las perturbaciones aquí descritas sobre la BER del sistema, se realizaron medidas para cada una de las modulaciones siguiendo el mismo procedimiento. Este es: En primer lugar, se configuró el escenario multicamino presentado en (2.63); en segundo lugar, se inició el sistema para la modulación de la carga útil bajo medida; seguidamente, se procedió a modificar en tiempo real el desplazamiento en frecuencia de 0 a 0.005 en Frequency Offset; por último, se esperó a la convergencia de la cadena receptora y se tomaron medidas para distintos valores de  $\sigma^2$  utilizando en todas ellas



(a) Antes de la convergencia.

(b) Después de la convergencia.

Figura 6.8.- Constelación de la carga útil para 64-QAM antes y después de la convergencia de los bloques digitales de la cadena receptora para un canal multicamino con efecto Doppler.



(a) Antes de la convergencia.

(b) Después de la convergencia.

Figura 6.9.- Constelación de la carga útil para 16-QAM antes y después de la convergencia de los bloques digitales de la cadena receptora para un desvanecimiento en potencia de 3 dB.

la comprobación de  $10^5$  paquetes consecutivos.

La comparación de los resultados obtenidos frente a los del Apartado 6.2.1 se muestra en la Figura 6.10, la Figura 6.11, la Figura 6.12 y la Figura 6.13 para BPSK, QPSK 16-QAM y 64-QAM, respectivamente. Comparando los resultados experimentales, nótese que en todos los casos el efecto conjunto del multicamino y el desplazamiento en frecuencia provoca una degradación en la BER medida equivalente a aumentar  $\sigma^2$  en 6 dB aproximadamente.

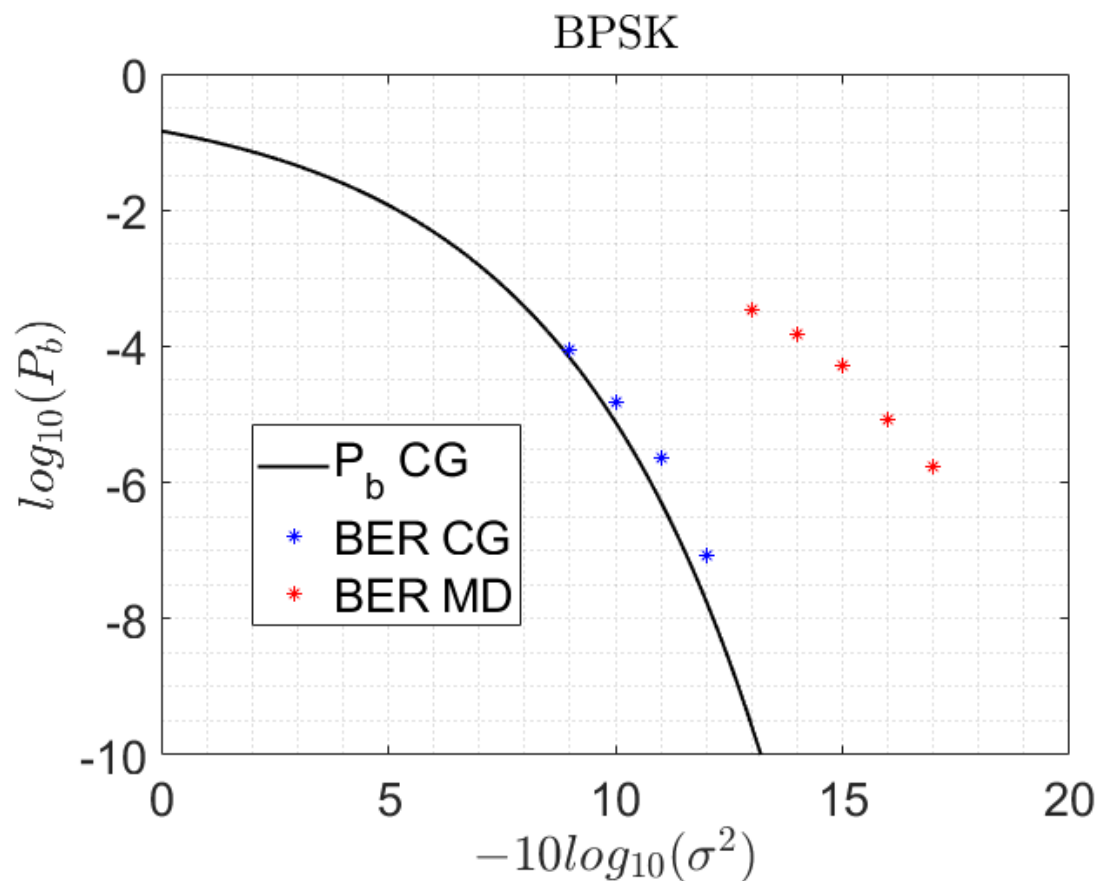


Figura 6.10.-  $P_b$  teórica y BER medida en función de  $\sigma^2$  para BPSK en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD).



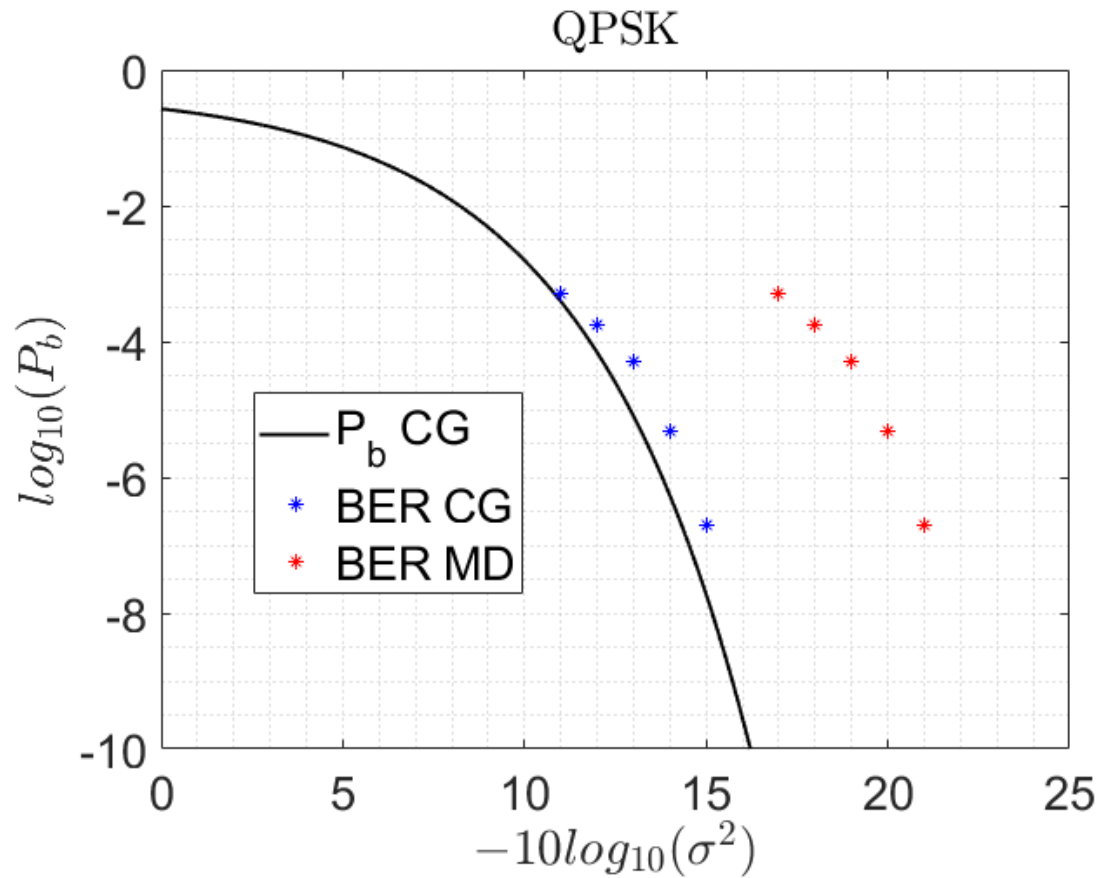


Figura 6.11.-  $P_b$  teórica y BER medida en función de  $\sigma^2$  para QPSK en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD).

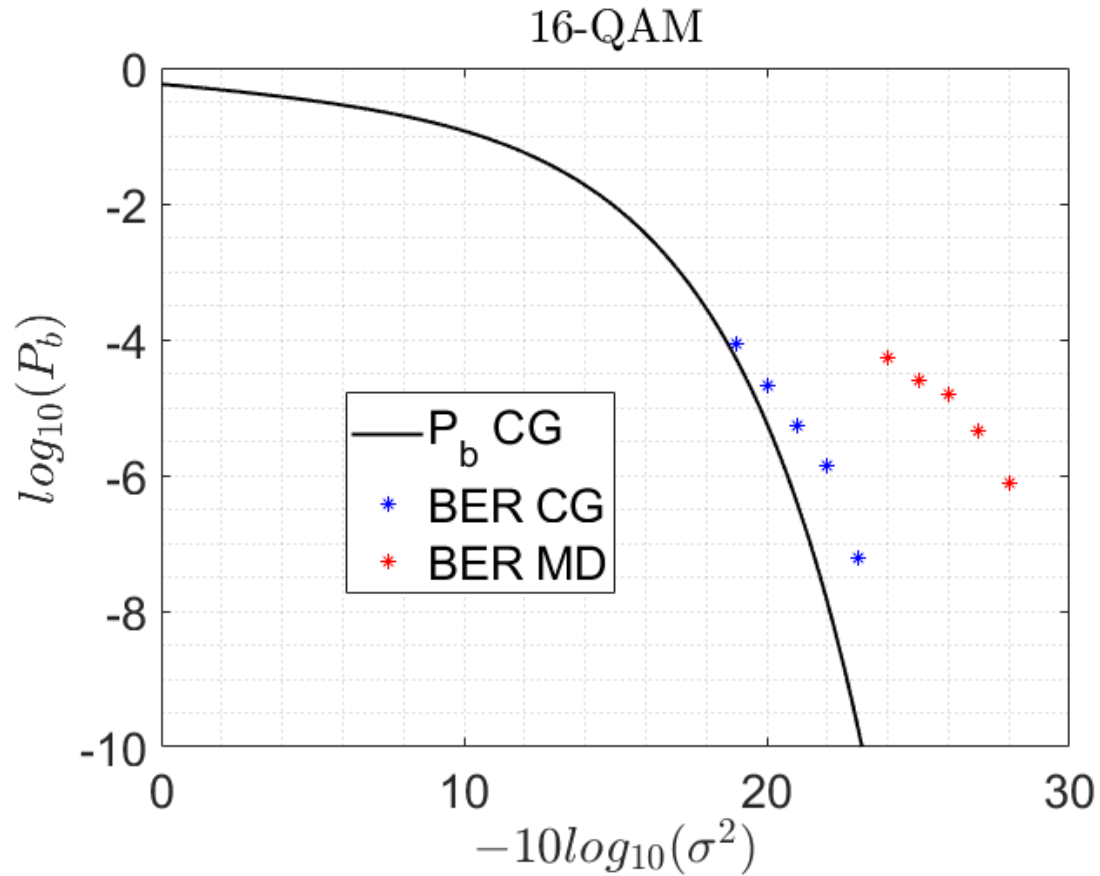


Figura 6.12.-  $P_b$  teórica y BER medida en función de  $\sigma^2$  para 16-QAM en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD).

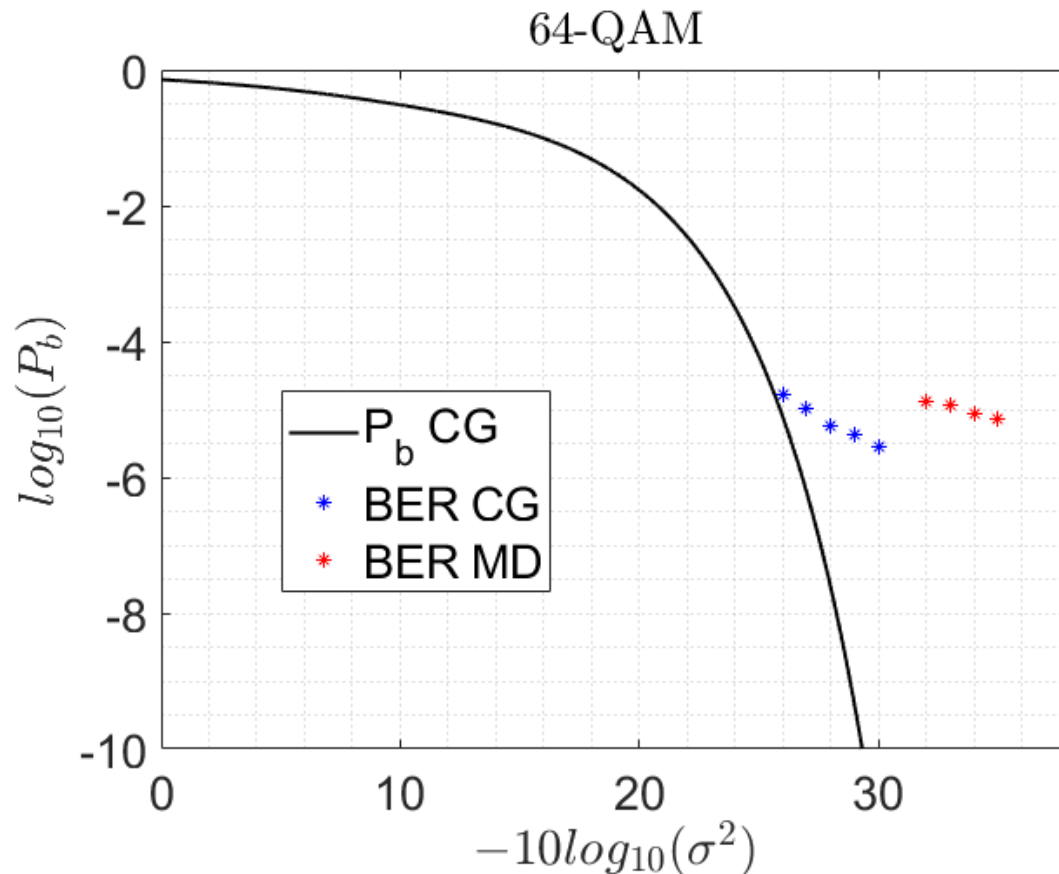


Figura 6.13.-  $P_b$  teórica y BER medida en función de  $\sigma^2$  para 64-QAM en la carga útil. Resultados teóricos en canal gaussiano (línea continua) y resultados experimentales (asteriscos) tanto para canal gaussiano (CG) como para un canal con multicamino y efecto Doppler (MD).

## 6.2.4.- MODULACIÓN ADAPTATIVA

Todas las medidas presentadas hasta ahora se han realizado utilizando las distintas modulaciones de la carga útil de forma aislada, sin implementar el cambio de modulación por medio del Canal de Retorno. Además, en las gráficas mostradas se ha comparado siempre la BER frente a la  $P_b$  teórica en función de  $\sigma^2$ . Sin embargo, como se adelantó en el Apartado 6.1.2, lo interesante desde el punto de vista del cambio de modulación es conocer si la estimación de la probabilidad de error,  $\tilde{P}_b$ , realizada en función de la medida de la dispersión,  $\tilde{\sigma}^2$ , es fiable. Con el objetivo de realizar dicha comprobación, durante la campaña de medidas descrita en el Apartado 6.2.3 se recogió también la media de todas las medidas de  $\tilde{\sigma}^2$  realizadas por el bloque Dispersion Probe durante la comprobación de los paquetes recibidos. El resultado se observa en la Figura 6.14. Nótese que, a pesar de los efectos del multicamino y desplazamiento en frecuencia,  $\tilde{\sigma}^2$  permite una estimación fiable de  $\tilde{P}_b$  en todos los casos, salvo en 64-QAM. Las medidas se corresponden con las mostradas en las gráficas del Apartado 6.2.3, pero, en este caso se representan en función de  $\tilde{\sigma}^2$ , y no del ruido digital

añadido en el transmisor,  $\sigma^2$ . En esta línea, se recuerda que la estimación de  $\tilde{P}_b$  parte de la aproximación de modelar todas las perturbaciones del canal como ruido blanco gaussiano, lo que se comprueba como razonable en base a los resultados obtenidos dentro de los márgenes medidos.

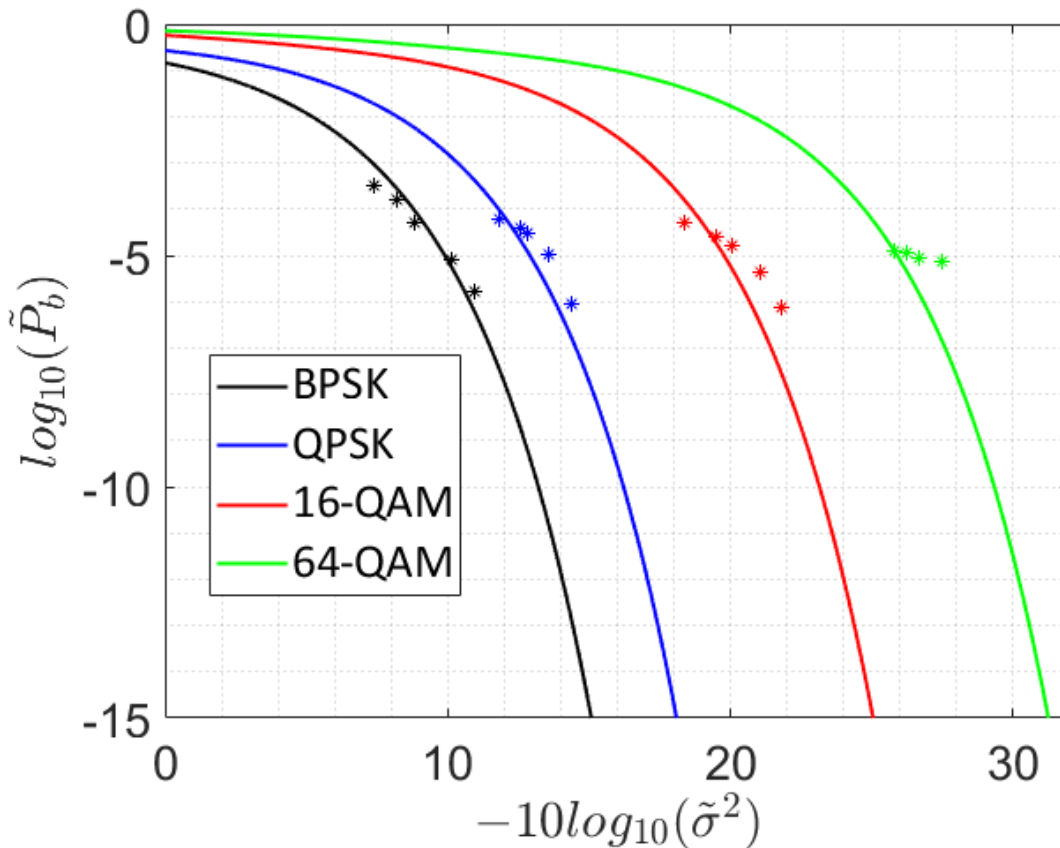


Figura 6.14.-  $\tilde{P}_b$  estimada en función de  $\tilde{\sigma}^2$  para las distintas modulaciones empleadas en la carga útil. Las medidas se corresponden con las del Apartado 6.2.4. Resultados teóricos y experimentales representados mediante línea continua y asteriscos, respectivamente.

En base a toda la información recogida en este capítulo es posible ajustar el sistema para realizar los cambios de modulación asegurándose que no se supera el límite impuesto por  $BER_{max}$ , al tiempo que se maximiza la tasa binaria  $R$  y, por tanto, la eficiencia espectral,  $\nu$  (2.17). Para cuantificar esta mejora se debe tener en cuenta que el período de transmisión de paquete es siempre el mismo en todos los casos (Apartado 5.2.1), y que su longitud en bytes viene indicada en la Tabla 4.1 para los cuatro posibles casos. En base a estas condiciones, la tasa binaria mejora respecto al empleo de BPSK en la carga útil en un factor de 1.78, 3.34 y 4.9 para QPSK, 16-QAM y 64-QAM, respectivamente. La eficiencia espectral se ve también mejorada por el mismo factor.

Si se quiere fijar el límite en la tasa de error binaria como  $BER_{max} = 10^{-4}$ , se ajustarían los umbrales en los bloques Dispersion Probe (Apartado 5.3.1) con los valores mostrados



en la Tabla 6.2. Esto resulta en los cambios de modulación en la carga útil representados en la Figura 6.15. El criterio seguido para el cambio a modulaciones de mayor número de bits por símbolo es el de tener un valor en  $-10\log(\tilde{\sigma}^2)$  al menos 1.5 dB por encima del valor que supondría la  $BER_{max}$  en la modulación de destino. Para el cambio hacia menor  $N$ , el margen empleado es de 0.5 dB en la modulación de partida.

Por otro lado, se configuran los límites inferiores de la tasa de acierto de paquete, PSR, de acuerdo con los valores de la Tabla 6.3. Estos son calculados por medio de (6.1) y para  $BER_{max}$ , e introducidos en el bloque CRC32 Check (Apartado 5.3.1). Como el número de bits transmitidos en la carga útil,  $L$ , es distinto para cada modulación, los valores obtenidos también lo son.

$Th_1$	$2.399 \times 10^{-3}$
$Th_2$	$3.020 \times 10^{-3}$
$Th_3$	$9.772 \times 10^{-3}$
$Th_4$	$12.3039 \times 10^{-3}$
$Th_5$	$46.773 \times 10^{-3}$
$Th_6$	$58.884 \times 10^{-3}$

Tabla 6.2.- Configuración de los umbrales en las instancias de Dispersion Probe para  $BER_{max} = 10^{-4}$ .

Rate QPSK	0.91
Rate 16-QAM	0.82
Rate 64-QAM	0.74

Tabla 6.3.- Configuración de los valores mínimos de PSR en el bloque CRC32 Check para  $BER_{max} = 10^{-4}$ .

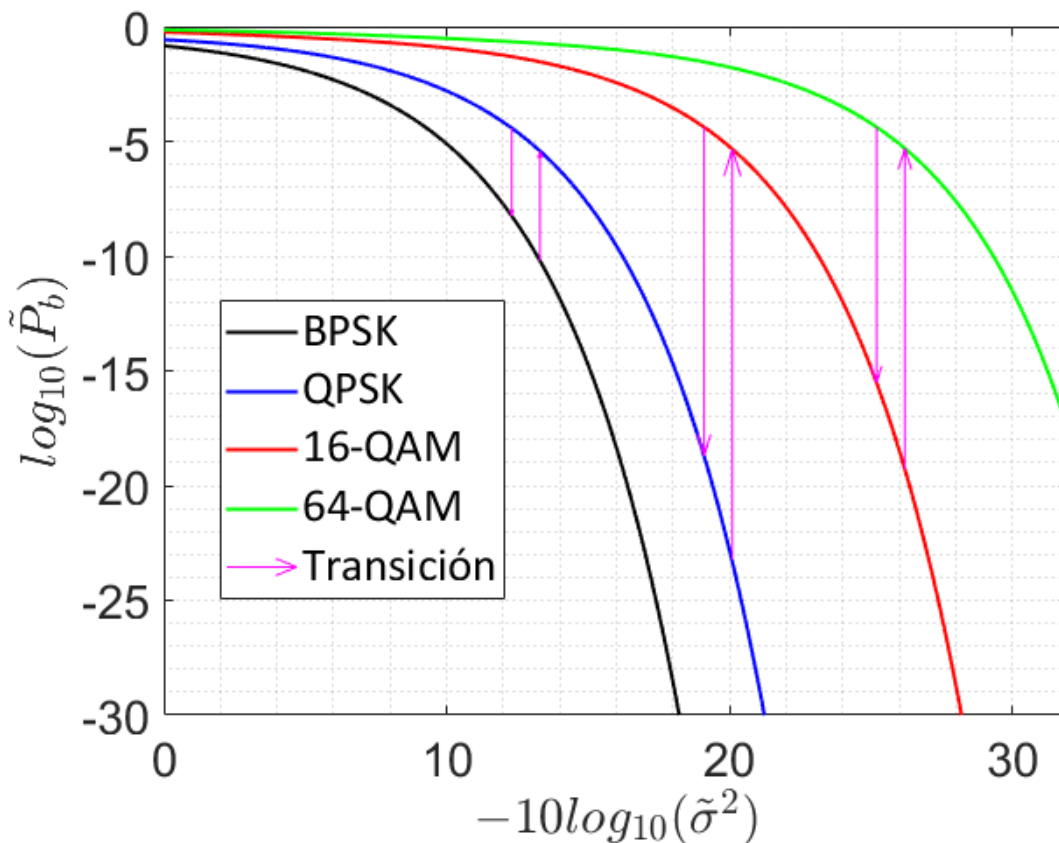


Figura 6.15.-  $\tilde{P}_b$  estimada en función de  $\tilde{\sigma}^2$ . Representación de los cambios entre las distintas modulaciones de la carga útil para  $BER_{max} = 10^{-4}$ .

Por último, se comprueba experimentalmente el funcionamiento del sistema habilitando el cambio de modulación automático y fijando los umbrales presentados en este apartado. Los cambios de modulación se provocan modificando en tiempo real los efectos del canal de ruido y desplazamiento en frecuencia, así como la ganancia del transmisor. En todos los casos se observa una adaptación de la modulación de la carga útil satisfactoria. En la mayoría de ellos, el cambio se produce por comparación de  $\tilde{\sigma}^2$  con los umbrales  $Th$ , y siempre unos pocos segundos después de que se empiece a cumplir la condición correspondiente. Este tiempo de conmutación puede ser acelerado aumentando la tasa de muestreo. En los casos en los que se fuerza una degradación súbita del canal, se comprueba también el cambio por medio de la comprobación de paquetes en CRC32 Check, que cumple con su objetivo de proteger el sistema en estos casos. Además, para este ejemplo en el que el valor de  $BER_{max} = 10^{-4}$ , la estimación de  $\tilde{P}_b$  en función de  $\tilde{\sigma}^2$  sí que resulta ser fiable en 64-QAM. No obstante, como se aprecia en la Figura 6.14, esto no es así para valores de  $BER$  inferiores, por ejemplo,  $BER_{max} = 10^{-6}$ , lo que debe ser tenido en cuenta en estos casos.

## 6.3.- CANAL INALÁMBRICO

En este apartado se describe la comprobación del funcionamiento del sistema en un canal inalámbrico definido por un escenario físico. Para ello se emplean sendos dipolos conectados al Tx\_Tx y el Rx\_Rx del Canal Principal, mientras que se mantiene el cable coaxial para el Canal de Retorno. Esta última característica no supone un problema de cara a comprobar la robustez del sistema, pues el Canal de Retorno es el elemento más fiable del mismo al emplear siempre la modulación BPSK. El montaje resultante se muestra en la Figura 6.16. La configuración de los USRP se corresponde con lo mostrado Tabla 6.1, pero modificando la frecuencia de canal,  $f_c$ , de 800 MHz a 1.06 GHz en el Canal Principal (Tx\_Tx y Rx\_Rx).

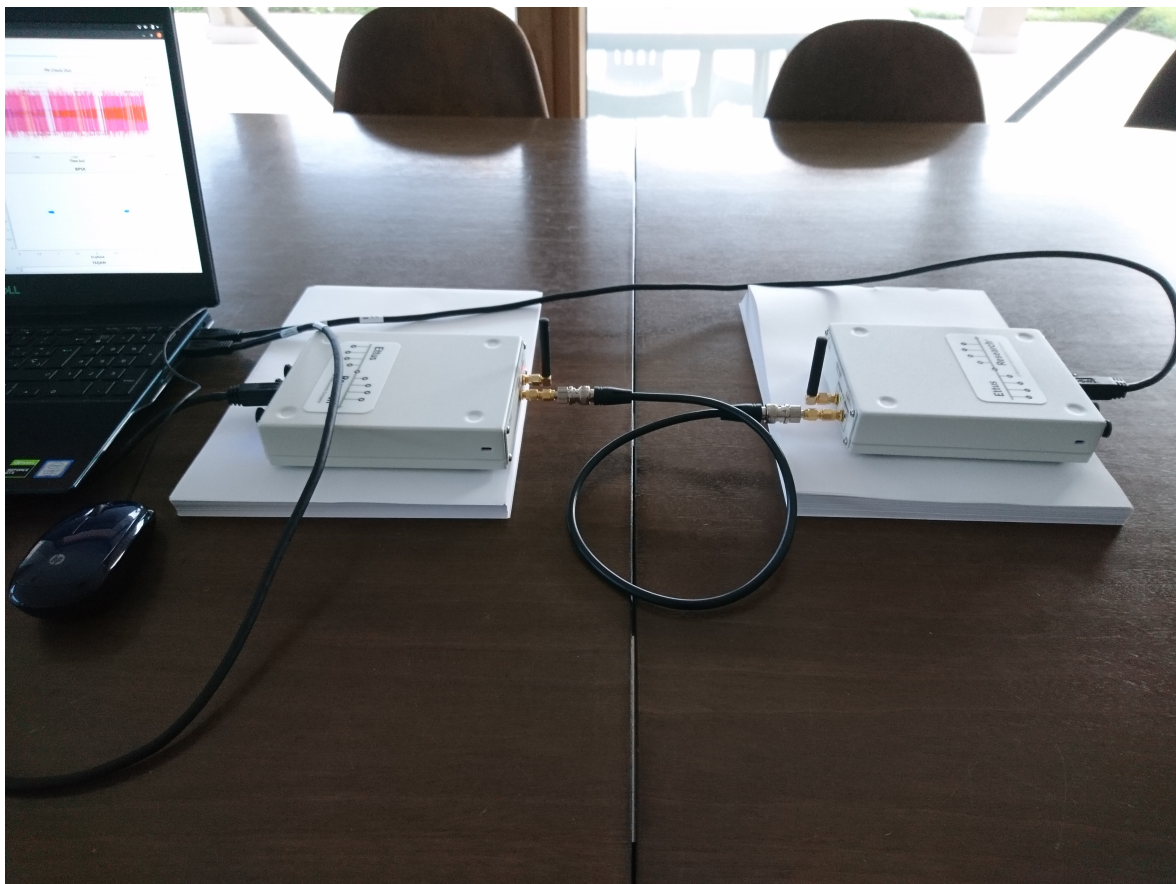
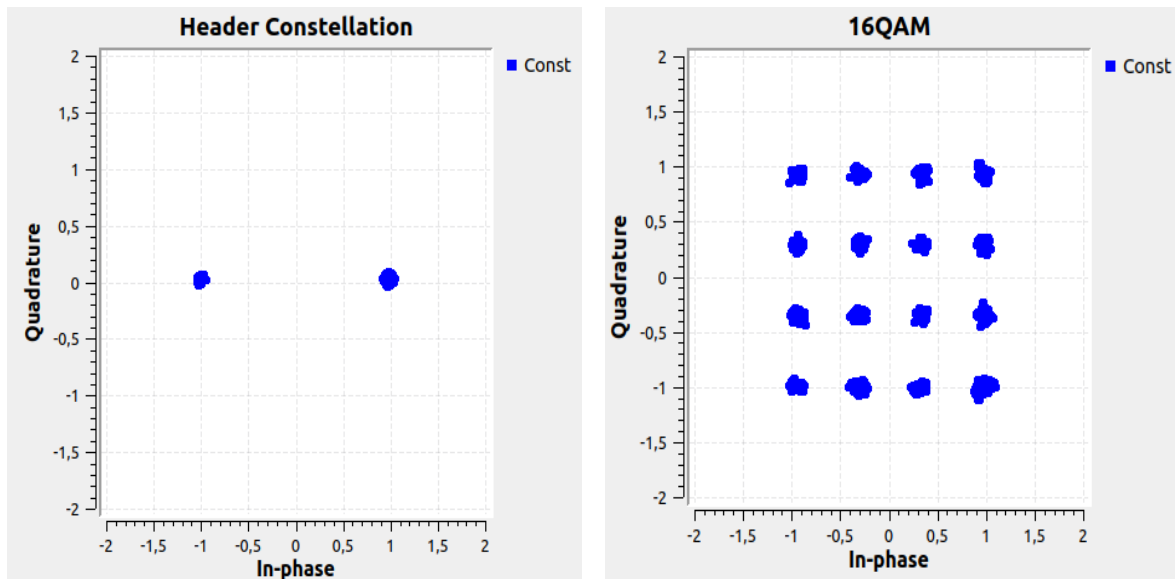


Figura 6.16.- Montaje del sistema para el canal inalámbrico.

Tras el arranque inicial, se observó el cambio de modulación desde BPSK a 16-QAM, sin llegar a 64-QAM. Los umbrales fijados se corresponden con una  $BER_{max} = 10^{-4}$ , según lo expuesto en el Apartado 6.2.4. En la Figura 6.17 se observa la constelación recibida en cabecera y carga útil una vez alcanzada 16-QAM.

Con el objetivo de introducir una variación en temporal en el canal físico, se empleó una plancha metálica para generar un efecto multicamino, como se representa en la Figura 6.18. En la Figura 6.19 se puede observar el efecto de este fenómeno en la constelación reci-



(a) Cabecera (BPSK).

(b) Carga útil (16-QAM).

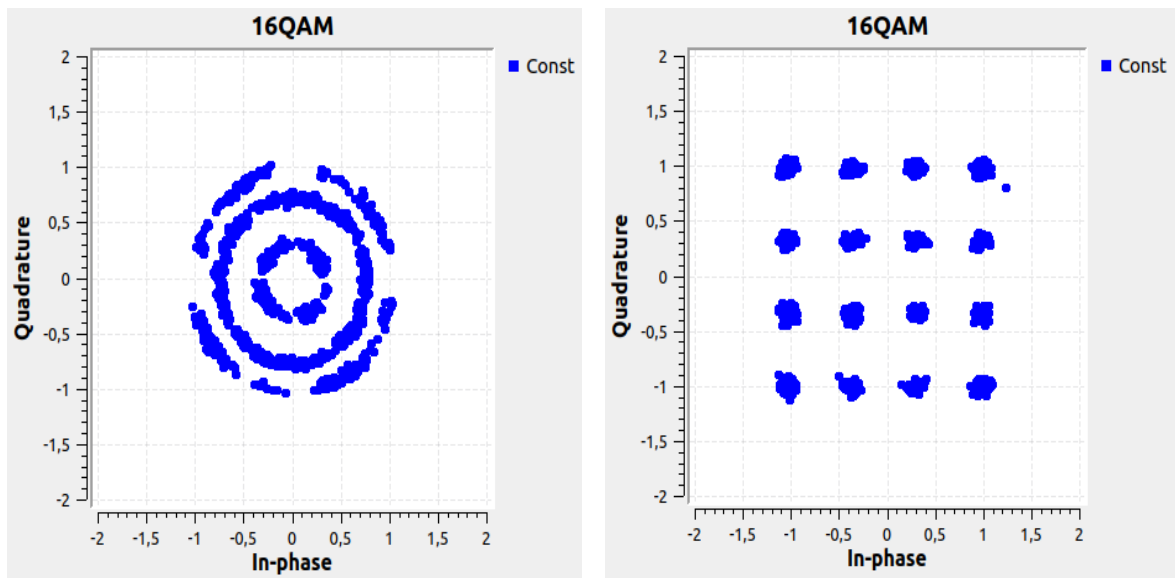
Figura 6.17.- Constelación de la cabecera y la carga útil una vez alcanzada 16-QAM en la segunda.

bida antes y después de la convergencia de los bloques de la cadena receptora. Es importante destacar que, en este caso, no se llegó a producir el cambio a QPSK. Como se explicó en el Apartado 5.3.1, el bloque Dispersión Probe espera a que dos medidas consecutivas cumplan la condición necesaria para realizar la petición de cambio. Para el caso presentado, el sistema converge antes de realizar la segunda medida, razón por la que el cambio no tiene lugar. Lo mismo sucede con el cambio por PSR, del que el bloque CRC32 Check es responsable.





Figura 6.18.- Montaje del sistema para el canal inalámbrico introduciendo un efecto multi-camino.



(a) Antes de la convergencia.

(b) Después de la convergencia.

Figura 6.19.- Constelación de la carga útil (16-QAM) antes y después de la convergencia al modificar el escenario para introducir un canal con efecto multicamino.



# 7. CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

En este capítulo se presentan las conclusiones obtenidas a partir de lo expuesto en la presente memoria, así como las líneas de trabajo futuras que ampliarían este TFM.

## 7.1.- CONCLUSIONES

En el desarrollo de este TFM se ha diseñado un sistema de comunicaciones digitales basado en el paradigma de SDR que utiliza la herramienta GNU Radio en su definición de software y los dispositivos de hardware USRP para su implementación física.

Dicho sistema define un protocolo de comunicación basado en paquetes, en los que la cabecera se encuentra modulada en BPSK y la carga útil puede variar entre BPSK, QPSK, 16-QAM y 64-QAM. El objetivo de la modulación adaptativa es el de aumentar lo máximo posible la tasa binaria manteniendo constantes el ancho de banda y la potencia transmitida, y asegurando siempre una tasa de error binaria máxima,  $BER_{max}$ , siendo esta un valor configurable.

Por otro lado, cada una de las cadenas receptoras correspondientes a cada modulación está diseñada para adaptarse a los tipos de canales y perturbaciones descritos en el Apartado 2.3, que son comunes en un escenario de comunicaciones móviles. Entre ellos destacan un canal variante en el tiempo con ruido, multicamino, efecto Doppler, fallos de sincronización y desvanecimientos en potencia.

Con el fin de verificar el correcto funcionamiento del sistema, se han realizado medidas empleando tanto un canal controlado por software (Apartado 6.2), como un canal inalámbrico en el que se modifica físicamente el escenario de comunicaciones (Apartado 6.3). Los resultados obtenidos verifican el correcto funcionamiento del sistema, a la vez que proporcionan información empírica acerca de su correcta configuración.

Cabe resaltar que todas las cadenas receptoras soportan las perturbaciones descritas de forma notable, y son capaces de converger rápidamente a variaciones temporales. No obstante, el funcionamiento de los bloques digitales adaptativos que las conforman introduce una ligera dispersión que limita el empleo de las modulaciones 16-QAM y 64-QAM para valores de  $BER_{max}$  inferiores a  $10^{-6}$ . Esta limitación es especialmente notoria para 64-QAM, donde además la estimación de la probabilidad de error de bit en función de la dispersión de las muestras recibidas deja de ser fiable. No obstante, un conocimiento de este fenómeno junto a la comprobación en tiempo real de la tasa de acierto de paquete, PSR, permiten que, correctamente configurado, el sistema sea operativo también en estos casos.

Por último, y como resultado de todo el proceso, se ha comprobado la enorme versatilidad y potencia de las herramientas empleadas, a saber GNU Radio y los dispositivos



USRP. En relación a la primera, se han programado bloques digitales de procesado de señal requeridos específicamente para el funcionamiento del sistema desarrollado. Dichos bloques han sido perfectamente integrados en el entorno de GNU Radio, y además resultan en código portable a otra instancia de esta aplicación. Respecto de los USRP, y más concretamente hablando del modelo B200, se ha hecho uso de la capacidad de configurar en tiempo real parámetros como la frecuencia de canal o la ganancia de transmisor y receptor. Además, se ha comprobado el funcionamiento full-dúplex del mismo, al utilizar cada dispositivo para transmitir y recibir simultáneamente. Por otro lado, se ha comprobado la compatibilidad existente entre GNU Radio y los dispositivos USRP.

## 7.2.- LÍNEAS DE TRABAJO FUTURAS

Las líneas de trabajo futuras más destacadas de cara a la ampliación de este TFM se recogen a continuación:

- Utilizar sistema presentado para la implementación de la capa física de una aplicación de intercambio de información concreta, tal y como se adelantó en el Apartado 5.2.1. Algunos ejemplos serían una aplicación de envío de vídeo o de archivos de texto. Esto requeriría de una capa de control adicional, tanto en el transmisor como en el receptor. En el transmisor su función sería la de asegurar que la capa física, en este caso el sistema, transmite la totalidad del flujo de información generado. El objetivo en el receptor pasaría por agrupar los bits demodulados de forma correcta para su traslado de nuevo a la capa de aplicación.
- Añadir codificación de canal que permita la corrección de errores. El presente sistema implementa una codificación de canal basada en el código CRC32 que permite únicamente la detección de errores. Una codificación de canal adicional para la corrección de estos supondría aumentar la redundancia en la información transmitida, para a cambio disminuir la tasa de error de binaria. El balance entre ambos aspectos se realizaría en función de los requisitos de la BER para la aplicación escogida.
- Añadir igualación de canal no ciega. La igualación de canal empleada en el sistema es siempre ciega, es decir, no se conoce la respuesta del canal en cada momento. Suponiendo un escenario de comunicaciones móviles, es posible implementar algoritmos de igualación no ciega realizando una estimación periódica de la respuesta del canal basada en la transmisión de una secuencia conocida. Esto permitiría mejorar el funcionamiento de las cadenas receptoras de todas las modulaciones

Adicionalmente, se puede contemplar el añadir más modulaciones al sistema y extender las medidas realizadas. En relación a esto último, resultaría especialmente interesante



utilizar la capacidad de controlar el canal por medio de software para programar variaciones temporales automáticas que se correspondan con modelos teóricos de canales móviles.



## REFERENCIAS

- [1] Página web de GNU Radio<sup>14</sup>.  
<https://www.gnuradio.org/>
- [2] Página web de Ettus Research.  
<https://www.ettus.com/>
- [3] R. Steele, L. Hanzo, “Mobile Radio Communications” John Wiley & Sons, Second edition, 1999.
- [4] J. M. Hernando Rábanos, “Comunicaciones móviles”, Ed. Centro de Estudios Ramón Areces, 2ª Edición, 2004.
- [5] S. Ahmadi, “LTE-Advanced. A practical Systems Approach to understanding the 3GPP LTE Releases 10 and 11 Radio Access Technologies”, Elsevier, 2014.
- [6] ETSI EN 300 401: "Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".
- [7] ETSI TS 101 756: "Digital Audio Broadcasting (DAB); Registered Tables".
- [8] Página web del estándar DVB.  
<https://www.dvb.org/standards>
- [9] Página web de la iniciativa 5G Infrastructure Public Private Partnership (5G PPP).  
<https://5g-ppp.eu/>
- [10] P. Zhang et al., "Design of Reconfigurable SDR Platform for Antenna Selection Aided MIMO Communication System," in IEEE Access, vol. 7, pp. 169267-169280, 2019, doi: 10.1109/ACCESS.2019.2946720.
- [11] S. Dörner, S. Cammerer, J. Hoydis and S. t. Brink, "Deep Learning Based Communication Over the Air," in IEEE Journal of Selected Topics in Signal Processing, vol. 12, no. 1, pp. 132-143, Feb. 2018, doi: 10.1109/JSTSP.2017.2784180.
- [12] Tingjun Chen, Mahmood Baraani Dastjerdi, Jin Zhou, Harish Krishnaswamy, Gil Zussman, "Wideband Full-Duplex Wireless via Frequency-Domain Equalization: Design and Experimentation", arXiv:1812.01126, 2018.
- [13] John G. Proakis, Masoud Salehi, “Digital Communications”, Mc Graw Hill, Fifth Edition, International Edition, 2008.

<sup>14</sup>Todas las páginas web aquí citadas han sido accedidas por última vez en julio de 2021.



- [14] Antoni Artés Rodríguez, et alter, “Comunicaciones Digitales”, Pearson Educación S.A., 2007.
- [15] Código fuente actualizado del proyecto GNU Radio en Github.  
<https://github.com/gnuradio/gnuradio>
- [16] Manual de GNU Radio y C++ API Reference.  
<https://www.gnuradio.org/doc/doxygen/index.html>
- [17] Dispositivos hardware compatibles con GNU Radio.  
<https://wiki.gnuradio.org/index.php/Hardware>
- [18] GNU Radio Out Of Tree Modules.  
<https://wiki.gnuradio.org/index.php/OutOfTreeModules>
- [19] Código fuente actualizado de UHD desarrollada por Ettus Research.  
<https://github.com/EttusResearch/uhd>
- [20] Manual de los dispositivos USRP y de UHD de Ettus Research.  
<https://files.ettus.com/manual/>
- [21] Ettus Research Knowledge Base, "About USRP Bandwidths and Sampling Rates".  
[https://kb.ettus.com/About\\_USRP\\_Bandwidths\\_and\\_Sampling\\_Rates](https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates)
- [22] USRP B200 de Ettus Research.  
<https://www.ettus.com/all-products/ub200-kit/>
- [23] FPGAs modelo Spartan 6 de Xilinx.  
<https://www.xilinx.com/products/silicon-devices/fpga/spartan-6.html>
- [24] Circuito integrado AD9364 de Analog Devices  
<https://www.analog.com/en/products/ad9364.html>
- [25] Ettus Research Knowledge Base, "Selecting a USRP Device".  
[https://kb.ettus.com/Selecting\\_a\\_USRP\\_Device](https://kb.ettus.com/Selecting_a_USRP_Device)
- [26] Ettus Research Knowledge Base, "UHD".  
<https://kb.ettus.com/UHD>
- [27] F. Harris, E. Venosa, X. Chen and C. Dick, "Band edge filters perform non data-aided carrier and timing synchronization of software defined radio QAM receivers," The 15th International Symposium on Wireless Personal Multimedia Communications, 2012, pp. 271-275



- [28] F. J. Harris and M. Rice, "Multirate digital filters for symbol timing synchronization in software defined radios," in IEEE Journal on Selected Areas in Communications, vol. 19, no. 12, pp. 2346-2357, Dec. 2001, doi: 10.1109/49.974601
- [29] J. Feigin, "Practical Costas loop design: Designing a simple and inexpensive BPSK Costas loop carrier recovery circuit," RF signal processing, pp. 20-36, 2002
- [30] Marcus Müller, "Behind the Veil: A Peek at GNU Radio's Buffer Architecture"  
<https://www.gnuradio.org/blog/2017-01-05-buffers/>