



Universidad de  
Oviedo



**ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

**GRADO EN INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS  
DE TELECOMUNICACIÓN**

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

**IMPLEMENTACIÓN DE UN SISTEMA DE COMUNICACIONES  
DIGITALES BASADO EN CONSTELACIONES IMPROPIAS  
EMPLEANDO SOFTWARE RADIO**

**D. Daniel González Martínez**  
**TUTORES: D. Jesús Alberto López Fernández y D. Álvaro Pendás**  
**Recondo**

**FECHA: Julio 2024**

# Acrónimos

- **ADC:** Analog-to-Digital Converter
- **AWGN:** Additive White Gaussian Noise
- **BER:** Bit Error Rate
- **CPU:** Central Processing Unit
- **CRC32:** Cyclic Redundancy Check 32
- **DAC:** Digital-to-Analog Converter
- **DDC:** Digital Down Converter
- **DUC:** Digital Up Converter
- **FPGA:** Field Programmable Gate Array
- **GPIO:** General-Purpose Input/Output
- **GRC:** GNU Radio Companion
- **ISI:** Inter-Symbol Interference
- **JTAG:** Joint Test Action Group
- **ML:** Maximum Likelihood
- **MMSE:** Minimum Mean Square Error
- **PDU:** Protocol Data Unit
- **OFDM:** Orthogonal Frequency Division Multiplexing
- **QAM:** Quadrature Amplitude Modulation
- **QPSK:** Quadrature Phase Shift Keying
- **RMS:** Root Mean Square
- **SDR:** Software-Defined Radio
- **SER:** Symbol Error Rate
- **SMA:** SubMiniature version A
- **SNR:** Signal-to-Noise Ratio

- **SPR:** Successful Packet Rate
- **TCXO:** Temperature Compensated Crystal Oscillator
- **TFG:** Trabajo de Fin de Grado
- **UHD:** USRP Hardware Driver
- **USRP:** Universal Software Radio Peripheral
- **WLT:** Wavelet Transform

# Notación

- $j$ : Unidad imaginaria
- $\bar{s}(t)$ : Envoltente compleja de la señal  $s(t)$ . Tomando en cuenta  $s(t) = s_I \cos(\omega_c t) - s_Q \sin(\omega_c t)$ , la envoltente compleja se define como  $\bar{s}(t) = s_I + js_Q$
- $\hat{x}$ : Estimación de la muestra  $x$
- $X$ : Variable aleatoria. Se representa con letra mayúscula.
- $E[\cdot]$ : Esperanza de variable aleatoria
- $Q(x)$ : Calcula la probabilidad de que una variable aleatoria normal estándar (con media cero y varianza uno) sea mayor que el valor dado  $x$ . De forma análoga,  $Q^{-1}(x)$  representa la función inversa.

# Índice general

<b>1. Introducción</b>	<b>9</b>
1.1.- Motivación . . . . .	9
1.2.- Planificación . . . . .	11
1.3.- Objetivos . . . . .	12
1.4.- Estructura . . . . .	12
<b>2. Fundamento teórico</b>	<b>14</b>
2.1.- Esquema general de un sistema de comunicaciones digitales . . . . .	14
2.2.- Parámetros característicos de un sistema de comunicaciones digitales . . . . .	18
2.3.- Modulaciones digitales . . . . .	21
2.3.1.- Modulaciones PAM . . . . .	21
2.3.2.- Modulaciones QAM . . . . .	23
2.3.3.- Modulaciones propias e impropias . . . . .	25
<b>3. Herramientas utilizadas</b>	<b>29</b>
3.1.- Matlab . . . . .	29
3.2.- GNU Radio . . . . .	29
3.2.1.- Bloques y comunicación . . . . .	29
3.2.2.- Conexionado . . . . .	30
3.2.3.- Sistemas utilizados . . . . .	32
3.2.3.1.- Sistema auxiliar: QPSK monoportadora . . . . .	32
3.2.3.2.- Sistema auxiliar: OFDM . . . . .	35
3.2.4.- Sistema de comunicaciones digitales en GNU Radio . . . . .	37
3.2.5.- Bloque de cálculo de SER . . . . .	42
3.3.- USRP . . . . .	43
3.3.1.- Introducción . . . . .	44
3.3.2.- Modelo USRP B210 . . . . .	45
<b>4. Medidas de la tasa de error de los diferentes escenarios hardware y software y comparativa con simulaciones</b>	<b>48</b>
4.1.- Escenarios software . . . . .	48
4.2.- Escenarios hardware . . . . .	56
4.2.1.- Comunicación mediante medio guiado . . . . .	56
4.2.2.- Comunicación inalámbrica . . . . .	60
<b>5. Conclusiones y líneas futuras</b>	<b>65</b>

**Referencias**

**67**

# Índice de figuras

1.1. Planificación del estudio mediante un diagrama de Gantt. . . . .	11
2.1. Sistema de comunicaciones digitales básico. . . . .	14
2.2. Representación de una modulación QPSK, mediante cruces azules, frente a una constelación 16-QAM, mediante diamantes rojos, ambas normalizadas con energía igual a la unidad. . . . .	15
2.3. Ilustración del proceso de decisión mediante la representación de la observación y el símbolo estimado a partir de ella en función de la región de decisión sobre la que la observación se encuentra. . . . .	18
2.4. Representación de una constelación 4-PAM, indicando la distancia entre símbolos.	22
2.5. Representación de una constelación propia (de tipo QPSK), mediante círculos azules, frente a su versión impropia (con $\kappa = 0.6$ ), mediante cruces rojas. . . .	26
3.1. Ejemplo de conexión síncrona entre bloques de GNU Radio. . . . .	31
3.2. Ejemplo de conexión asíncrona entre bloques de GNU Radio. . . . .	31
3.3. Tipos de datos soportados por los bloques de GNU Radio. . . . .	32
3.4. Diagrama de bloques de sistema de comunicación monoportadora con modulación QPSK. . . . .	33
3.5. Diagrama de bloques de sistema de comunicación utilizando OFDM. . . . .	35
3.6. Bloque <i>Constellation Object</i> de GNU Radio. . . . .	38
3.7. Diagrama de bloques del sistema final utilizado para el análisis resultados. . . .	41
3.8. Bloque <i>Async CRC32</i> de GNURadio. . . . .	42
3.9. Bloque programado para la medida de la SER en base al código CRC32. . . . .	43
3.10. USRP B210. . . . .	45
3.11. Diagrama de bloques del funcionamiento interno del USRP. . . . .	46
3.12. Esquema de funcionamiento del USRP B210. . . . .	47
4.1. Constelación QPSK estándar que es, a su vez, propia ( $\kappa = 0$ ). . . . .	49
4.2. Constelación QPSK impropia con $\kappa = 0.2$ . . . . .	49
4.3. Constelación QPSK impropia con $\kappa = 0.5$ . . . . .	49
4.4. Constelación QPSK impropia con $\kappa = 0.8$ . . . . .	49
4.5. Constelación QPSK totalmente impropia ( $\kappa = 1$ ) semejante a una 4-PAM rotada.	49
4.6. Tasa de error de símbolo frente a relación señal-ruido de modulaciones QPSK impropias con distintos valores de $\kappa$ en entorno software, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos. . . . .	51
4.7. Constelación 16-QAM estándar que es, a su vez, propia ( $\kappa = 0$ ). . . . .	53
4.8. Constelación 16-QAM impropia con $\kappa = 0.2$ . . . . .	53

4.9. Constelación 16-QAM impropia con $\kappa = 0.5$ . . . . .	53
4.10. Constelación 16-QAM impropia con $\kappa = 0.65$ . . . . .	53
4.11. Constelación 16-QAM impropia con $\kappa = 0.8$ . . . . .	53
4.12. Tasa de error de símbolo frente a relación señal-ruido de modulaciones 16-QAM impropias con distintos valores de $\kappa$ en entorno software, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos. . . . .	54
4.13. Constelación 16-QAM totalmente impropia ( $\kappa = 1$ ) semejante a una 16-PAM rotada. . . . .	55
4.14. Tasa de error de símbolo frente a relación señal-ruido de constelaciones impropias con $\kappa = 1$ en el entorno software de en GNU Radio frente a simulaciones en Matlab de las modulaciones PAM. . . . .	56
4.15. Ilustración del montaje del escenario de comunicación mediante medio guiado.	57
4.16. Bloques de GNU Radio para comunicación con USRPs. . . . .	57
4.17. Tasa de error de símbolo frente a relación señal-ruido de las modulaciones QPSK usando distintos valores de $\kappa$ con USRPs conectados mediante cable SMA, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos. . . . .	59
4.18. Tasa de error de símbolo frente a relación señal-ruido de las modulaciones 16-QAM usando distintos valores de $\kappa$ con USRPs conectados mediante cable SMA, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos. . . . .	60
4.19. Ilustración del montaje del escenario de comunicación inalámbrica. . . . .	61
4.20. Tasa de error de símbolo frente a relación señal-ruido de las modulaciones QPSK usando distintos valores de $\kappa$ con USRPs conectados de forma inalámbrica. . .	62
4.21. Tasa de error de símbolo frente a relación señal-ruido de las modulaciones 16-QAM usando distintos valores de $\kappa$ con USRPs conectados de forma inalámbrica.	63
4.22. Tasa de error de símbolo frente a relación señal-ruido de constelaciones QPSK impropias con $\kappa = 1$ en todos los escenarios descritos. . . . .	64

# Índice de tablas

4.1. Constelaciones impropias QPSK. . . . .	50
---	----

# 1. Introducción

## 1.1.- Motivación

En el ámbito de las comunicaciones digitales, la búsqueda de técnicas que optimicen la eficiencia espectral, la robustez del sistema y la capacidad de adaptación a entornos cambiantes es continua. Si bien las modulaciones convencionales han sido ampliamente utilizadas y han demostrado su eficacia en diversos escenarios, su capacidad para enfrentar desafíos emergentes puede ser limitada, dando lugar a la aparición de nuevas modulaciones con diferentes características conocidas como impropias.

Las modulaciones impropias representan un enfoque innovador que permite una mayor flexibilidad en la representación de la información en el espacio de señal a través de una transformación de la constelación dependiente de un parámetro cuya elección se realiza en función del escenario de comunicaciones presente. Esta característica no convencional permite investigar el potencial de mejorar significativamente las prestaciones de las comunicaciones en redes con limitaciones de interferencias. Es así como, al facilitar una manipulación más flexible de las propiedades estadísticas de la señal, las constelaciones impropias contribuyen a mitigar la interferencia entre usuarios, permitiendo a un usuario secundario transmitir sin degradar significativamente la tasa de transmisión del usuario primario, lo que se traduce en una mejora general del rendimiento del sistema gracias a una gestión más eficiente de la interferencia.

En la última década, la tecnología de Software Defined Radio (SDR) ha revolucionado el diseño y la implementación de sistemas de comunicaciones al permitir que las funciones tradicionalmente realizadas por hardware sean programadas y configuradas por software. Este enfoque no solo ha democratizado el acceso a tecnologías avanzadas de comunicación, sino que también ha facilitado la investigación y desarrollo de nuevas técnicas de modulación y procesamiento de señales, como es el caso de las modulaciones impropias. GNU Radio, un entorno de desarrollo de código abierto ampliamente utilizado en la comunidad SDR, juega un papel crucial al proporcionar una plataforma flexible y poderosa para la implementación de algoritmos de procesamiento de señales. La combinación de SDR y GNU Radio no solo permite la exploración de técnicas de modulación no convencionales, como las modulaciones impropias, sino que también facilita la experimentación y evaluación en entornos controlados y reales. Esta integración abre nuevas posibilidades para optimizar el rendimiento del sistema, explorando cómo estas técnicas pueden mejorar la eficiencia espectral, la capacidad de adaptación dinámica y la robustez frente a condiciones variables del canal. Investigar cómo las modulaciones impropias pueden ser implementadas y optimizadas dentro del contexto de SDR y GNU Radio es fundamental para avanzar en el conocimiento y la aplicación práctica de estas técnicas en el campo de las comunicaciones digitales modernas.

El propósito principal de este Trabajo de Fin de Grado (TFG) es estudiar y eva-

luar las modulaciones impropias en el contexto de las comunicaciones digitales. Se busca comprender cómo estas técnicas pueden ser implementadas prácticamente para optimizar el rendimiento del sistema, obteniendo como indicador de rendimiento o factor de mérito su probabilidad de error asociada en la recepción de información. Además, se pretende explorar cómo las modulaciones impropias pueden integrarse de manera efectiva en entornos de SDR, apoyándose en una herramienta de software libre y código libre denominada GNU Radio y aprovechando las capacidades de adaptación y reconfiguración que este paradigma ofrece.

## 1.2.- Planificación

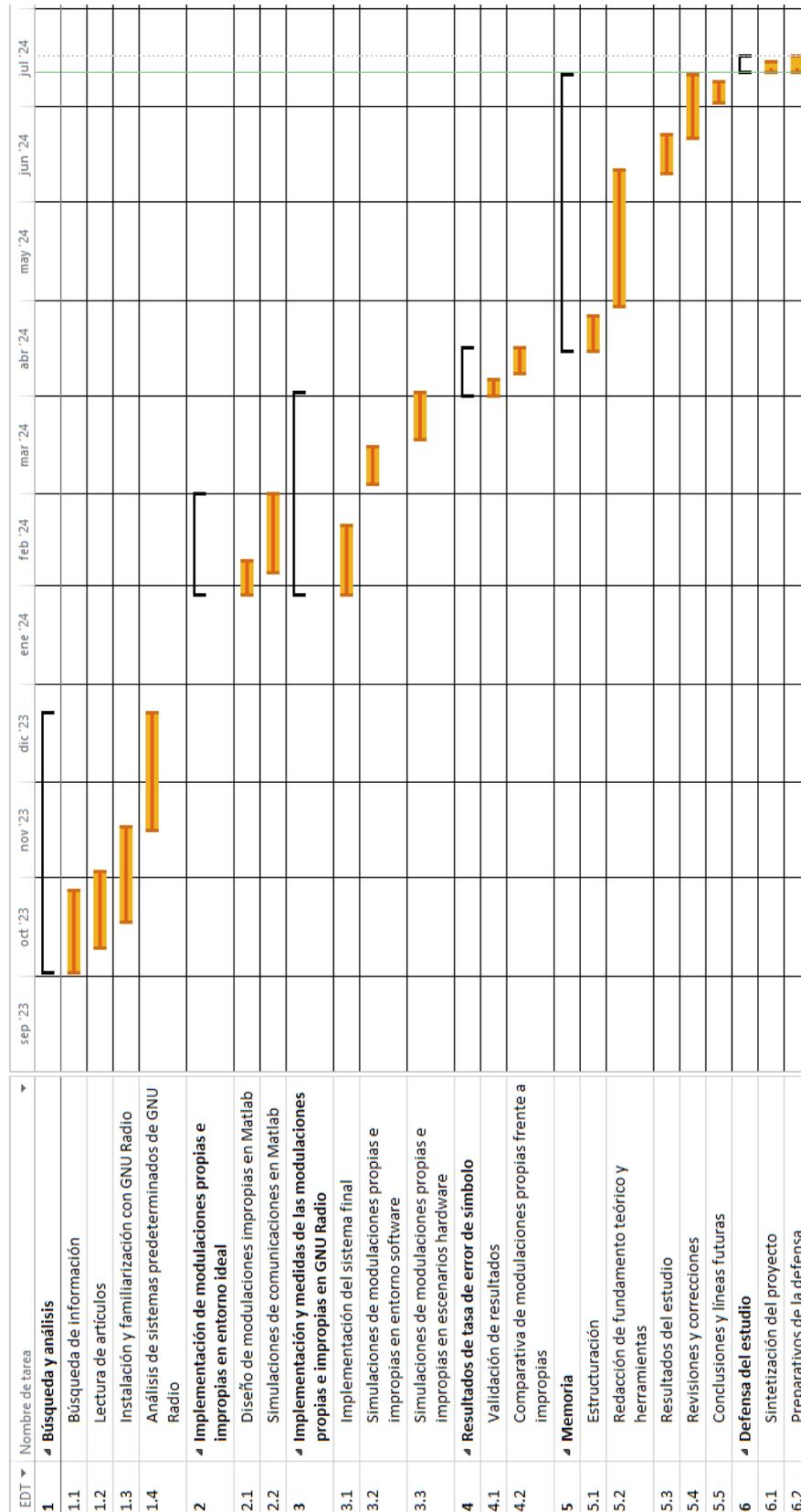


Figura 1.1.- Planificación del estudio mediante un diagrama de Gantt.

### 1.3.- Objetivos

El objetivo principal de este TFG es investigar y evaluar las modulaciones impropias en el contexto de las comunicaciones digitales, con el fin de optimizar el rendimiento de los sistemas de comunicación. Estas modulaciones ofrecen unas características innovadoras en la codificación de señales, permitiendo mejorar las condiciones de las modulaciones tradicionales.

Para lograr este objetivo, se comenzará con un análisis teórico de las modulaciones impropias. Este análisis permitirá entender las bases matemáticas y los beneficios potenciales de estas modulaciones en comparación con las modulaciones tradicionales. Se estudiarán las propiedades estadísticas de las constelaciones impropias y se evaluará su impacto en la probabilidad de error de símbolo.

Una vez comprendidos los fundamentos teóricos, se procederá a la implementación práctica de estas modulaciones utilizando la plataforma de SDR y la herramienta de software GNU Radio. Los subsistemas se desarrollarán y validarán inicialmente mediante simulaciones de software, asegurando que las modulaciones impropias funcionan correctamente en un entorno controlado. Posteriormente, se integrarán estos subsistemas en un sistema completo de transmisión y recepción, utilizando dispositivos USRP (Universal Software Radio Peripheral) para la implementación física.

El siguiente paso será evaluar el rendimiento del sistema implementado en diferentes condiciones operativas. Se realizarán pruebas en diversos escenarios cambiando el canal de transmisión, tanto en condiciones controladas como en entornos más realistas. Estas pruebas permitirán medir parámetros clave centrándose en la probabilidad de error de símbolo.

Finalmente, se elaborarán conclusiones basadas en el análisis de los resultados experimentales y teóricos. Estas conclusiones incluirán una valoración de la viabilidad y efectividad de las modulaciones impropias en sistemas de comunicación digitales basados en SDR. Se identificarán posibles mejoras y se propondrán líneas de investigación futura que puedan contribuir al avance en este campo.

### 1.4.- Estructura

La presente memoria se divide en cinco capítulos principales. Tras un primer capítulo de introducción donde se declara el objetivo y la motivación del estudio, el segundo capítulo presenta los fundamentos teóricos esenciales de las comunicaciones digitales sobre los que se basa el trabajo. El tercer capítulo describe las herramientas principales utilizadas, incluyendo Matlab y GNU Radio para la parte de software, y los dispositivos USRP para la parte de hardware.

El cuarto capítulo está dedicado al diseño y análisis de resultados del sistema final,

incluyendo tanto la implementación de software como de hardware. Por último, el quinto capítulo se reserva para las conclusiones y las líneas de trabajo futuras.

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

## 2. Fundamento teórico

En este capítulo, se establecen los fundamentos esenciales de las comunicaciones digitales, sentando así las bases teóricas necesarias para abordar el diseño del sistema en desarrollo. De manera concisa y esquemática, se exponen los conceptos clave que servirán como cimiento para el desarrollo del sistema.

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

Con la finalidad de familiarizarse con el funcionamiento de un sistema de comunicaciones digitales, se comenzará con una explicación del esquema general mostrado en la Figura 2.1. En aras de la claridad, adaptaremos este esquema para un sistema de una sola portadora y un canal gaussiano.

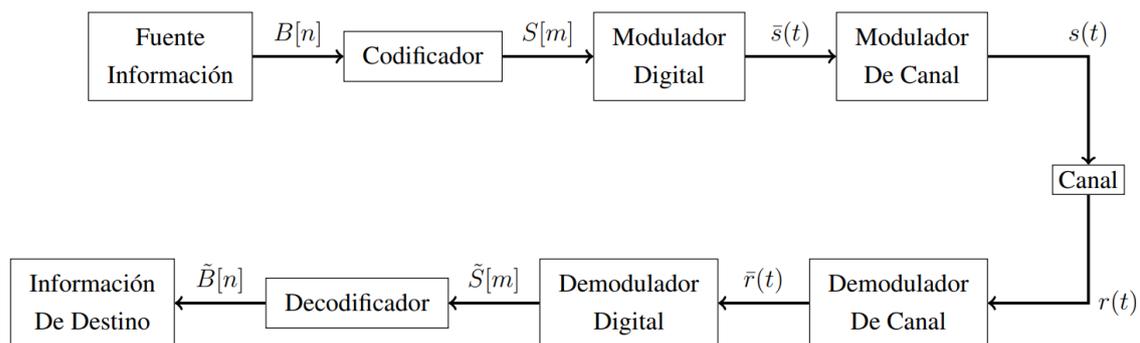


Figura 2.1.- Sistema de comunicaciones digitales básico.

En primer lugar, se parte de la *Fuente de Información binaria*, la cual será la encargada de generar la señal  $B[n]$ , definida por los bits de información generados por el transmisor en la comunicación. Esta señal se someterá a diferentes procesos durante la comunicación para ser transmitida por un medio y, posteriormente, recuperada en su totalidad por el receptor, de manera que la secuencia de bits recibidos por el receptor sea equivalente a la original. Tras la fuente de información, se encuentra el bloque codificador, cuyo principal propósito será el de proteger al información frente a errores de transmisión (codificador de canal) y escuchas no deseadas (encriptador), así como reducir la redundancia de la fuente (*Codificador de fuente*), aunque, en aspectos más específicos que no se abarcarán en este proyecto, podría tener distintas utilidades. El propósito del Modulador Digital radica en la transformación de la secuencia de bits generada por el codificador de canal en una señal transmisible por el canal de comunicaciones. Esta señal se formará mediante agrupación de  $N$  bits, dependien-

tes de la constelación utilizada, donde cada agrupación se corresponderá con un símbolo de  $S[m]$ .

El *Modulador Digital* del transmisor se puede desglosar, desde una perspectiva teórica, en dos bloques: el *Modulador Discreto* y el *Modulador de Señal*. La función del primero radica en asignar un vector de coordenadas a un conjunto de  $N$  bits. Dicho vector da lugar a la representación en el espacio de la señal del símbolo y dependerá de la modulación empleada. Cada símbolo se asigna a una coordenada, y el número de símbolos  $M$  es igual a  $2^N$ . La Figura 2.3 ilustra un esquema de las dos modulaciones que se emplearán en este proyecto (QPSK y 16-QAM), con  $N = 2$  y  $N = 4$  respectivamente. Es relevante destacar que el espacio de señal asociado a estas modulaciones tiene dos dimensiones, lo que es independiente del número de bits por símbolo, por lo que sus constelaciones asociadas se representan sobre un plano.

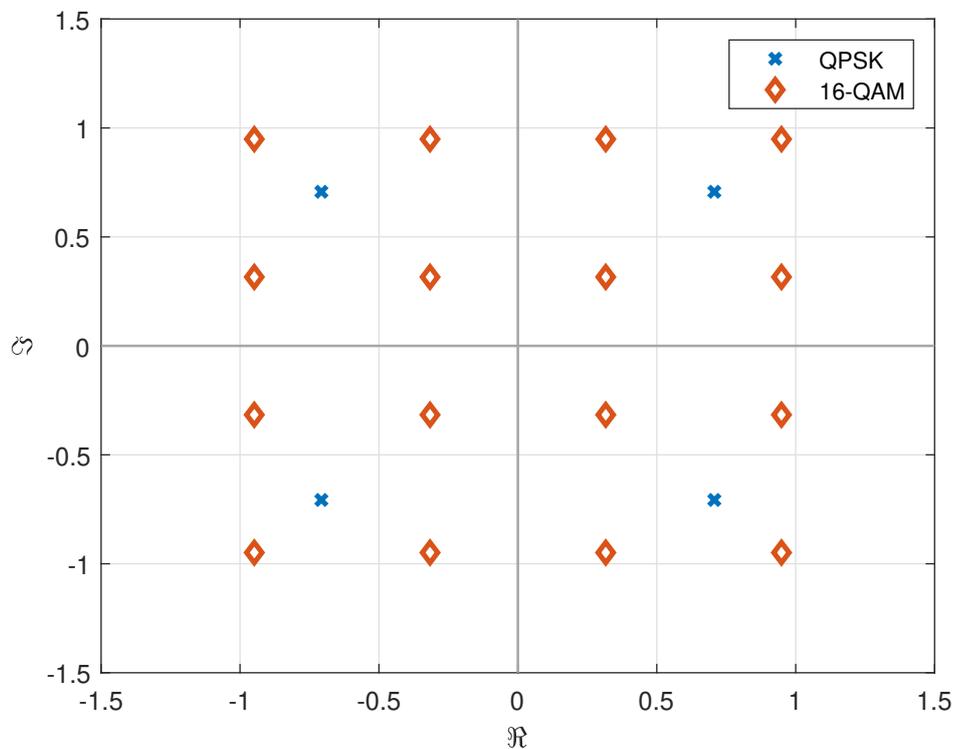


Figura 2.2.- Representación de una modulación QPSK, mediante cruces azules, frente a una constelación 16-QAM, mediante diamantes rojos, ambas normalizadas con energía igual a la unidad.

Por consiguiente, se requieren dos números reales, o uno complejo, para expresar las coordenadas de cada símbolo. En estos casos, es común referirse a la parte real como la componente en fase (I), mientras que la parte imaginaria se denota como la componente en cuadratura (Q), formando una modulación I-Q. A la salida de *Modulador Discreto* se situará

la secuencia discreta y compleja denotada  $A[m]$ , la cual se introducirá en el *Modulador de Señal*, encargado de generar la señal continua junto la función base:

$$\bar{s}(t) = \sum_{m=-\infty}^{m=\infty} A[m]\phi(t - mT) \quad (2.1)$$

La señal  $\bar{s}(t)$  contiene las componentes I y Q de la señal de información a transmitir y se suele notar como envolvente compleja o equivalente paso-bajo de la señal transmitida  $s(t)$ . Luego de haberse ejecutado la modulación digital, es de interés trasladar la señal compleja a la frecuencia deseada, siendo el *Modulador de Canal* el encargado de realizar esta función. 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 a partir de las componentes I-Q según la siguiente expresión:

$$s(t) = \sum_{m=-\infty}^{\infty} A_I[m]\phi(t - mT) \cos(\omega_c t + \theta) - \sum_{m=-\infty}^{\infty} A_Q[m]\phi(t - mT) \sin(\omega_c t + \theta) \quad (2.2)$$

Una vez es generada la señal a la frecuencia de la portadora, se propagará por el canal de transmisión, iniciando posteriormente el proceso de recepción. El bloque inicial será el Demodulador de Canal, encargado de desplazar la señal a banda base. En este procedimiento se considerará el canal más simple posible, siendo este un gaussiano en el que no se sufre ningún tipo de distorsión y la única perturbación añadida es el ruido gaussiano, de ahí el nombre de canal gaussiano. La única perturbación considerada en este canal es la debida al ruido térmico que se produce, a su vez, como resultado de la agitación térmica de los electrones de los materiales conductor, resultando en variaciones aleatorias en el voltaje a lo largo del canal. Estas fluctuaciones aleatorias, se muestran como variables aleatorias gaussianas de media cero, las cuales se superponen a la señal transmitida. Por consiguiente, obtenemos  $r(t)$  a la salida del canal, denotada de la siguiente manera:

$$r(t) = s(t) + n(t), \quad (2.3)$$

siendo  $n(t)$  la componente de señal de salida del canal debida al ruido. Hay que destacar el uso de un filtro paso banda, previo al *Demodulador de Canal*, cuyo objetivo es eliminar en la medida de lo posible las perturbaciones existentes fuera del ancho de banda de la señal. Posteriormente, la señal se introduce en el *Demodulador de Señal*, el cual se encarga de procesar la forma de onda a su entrada para generar la secuencia de símbolos deseada,  $\tilde{S}[m]$ . Para obtener la información binaria original, el *Demodulador* analizará los fallos producidos durante la transmisión y producirá la secuencia binaria correspondiente  $\tilde{B}[m]$ . El proceso del *Demodulador Digital* se divide en dos etapas, similares a las del Modulador Digital. En primer lugar, el *Demodulador de Señal* convierte la señal recibida en coordenadas expresadas

en el mismo espacio de señal donde se define la modulación. Para lograr esto, se emplea un filtro adaptado a cuya salida se toman muestras en los instantes que son múltiplos del período de símbolo,  $mT$ . Este proceso es esencialmente un producto escalar entre la señal recibida y la señal base desplazada temporalmente. Así, las muestras de salida del filtro adaptado del receptor en los instantes  $mT$  consisten en la secuencia de símbolos transmitidos,  $A[m]$ , sumados a una secuencia de muestras de ruido  $n[m]$  que puede ser modelada como una secuencia de variables aleatorias gaussianas independientes e idénticamente distribuidas. Para obtener la secuencia a la salida, denominada  $R[m]$ , se realiza la convolución entre la señal recibida y la señal base desplazada en el tiempo. La segunda etapa del *Demodulador Digital*, el *Decisor*, toma la secuencia  $R[m]$  resultante y la convierte en una secuencia de símbolos, implicando tomar decisiones basadas en las muestras recibidas para determinar qué símbolo se transmitió originalmente. Para explicar este proceso, se parte de una variable aleatoria  $X$  que toma valores de un alfabeto discreto  $X_i = \{x_1, x_2, \dots, x_M\}$ . Considerando que estas muestras recorren el canal de comunicación y se expondrán al ruido provocado, en el proceso de demodulación se introduce  $Y = HX + N$  mostrándose  $H$  como la influencia de perturbación del canal sobre la muestra transmitida y  $N$  como el ruido aditivo. En este punto, el *Decisor* es el encargado de asignar el valor adecuado compensando dichas perturbaciones y estimar el símbolo transmitido como aquel más próximo a la muestra recibida, ver muestra en rojo en Figura 2.3 que se estima como símbolo con  $x_{10}$ . A tal fin, se emplea el operador argumento del mínimo,  $\arg \min$ , sobre el conjunto de símbolos de la constelación:

$$\hat{x} = \arg_{x_i} \min |y - x_i h|, \quad (2.4)$$

siendo  $y$  la observación a la salida del muestreador y  $h$  la realización del canal que suponemos conocida.

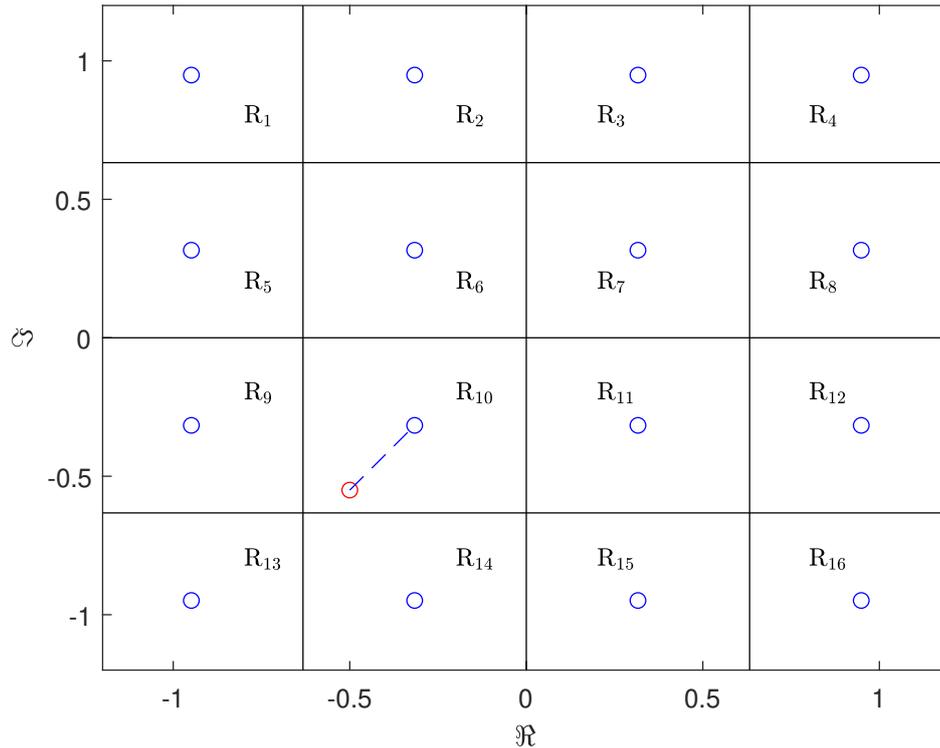


Figura 2.3.- Ilustración del proceso de decisión mediante la representación de la observación y el símbolo estimado a partir de ella en función de la región de decisión sobre la que la observación se encuentra.

Las regiones de decisión (también denominadas regiones de Voronoi), se definen como  $R_i$  con  $i = 1, 2, \dots, M$  y se conforman en base a la distancia mínima y a la probabilidad de cada símbolo, en este caso equiprobables, para establecer las áreas que determinan a qué variable se asigna cada muestra. Posteriormente, cada símbolo se desglosa en los bits correspondientes apoyándose en la constelación, obteniendo así la información de destino  $\tilde{B}[n]$ .

## 2.2.- Parámetros característicos de un sistema de comunicaciones digitales

A continuación, se muestra la obtención de algunos de los parámetros más característicos de los sistemas de modulación digital:

- $N$ : Número de bits por símbolo.
- $\xi_{av}$ : Energía media de la constelación obtenida promediando la energía de los símbolos representados en el espacio de señal. Se considera que cada símbolo tiene una probabilidad a priori  $P_i$  y energía  $\xi(a_i)$ , considerando  $a_i$  como las coordenadas complejas del símbolo en el espacio. La energía de cada símbolo se puede expresar como la norma

al cuadrado de su representación en el espacio de señal, resultando:

$$\xi_{av} = \sum_{i=1}^M P_i |a_i|^2 \text{ (J)} \quad (2.5)$$

- $\xi_0$ : Energía media de las señales transmitidas. Si la constelación representa la envolvente compleja de las señales realmente transmitidas (señales complejas), los símbolos transmitidos tendrán, en realidad, la mitad de la energía obtenida a partir de la constelación. Sin embargo, si la constelación representa las señales transmitidas (señales reales), entonces la energía de las mismas es la de su representación en espacio de señal. Así:

$$\xi_0 = \left\{ \begin{array}{l} \xi_{av} \text{ para señales reales} \\ \frac{1}{2}\xi_{av} \text{ para señales complejas} \end{array} \right\} \text{ (J)} \quad (2.6)$$

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

$$P_0 = \frac{\xi_0}{T} \text{ (W)} \quad (2.7)$$

- $\xi_b$ : Energía media por bit transmitido:

$$\xi_b = \frac{\xi_0}{N} \text{ (J)} \quad (2.8)$$

- $\gamma_b$ : Relación señal a ruido por bit. Representa la energía media por bit transmitido normalizada:

$$\gamma_b = \frac{\xi_b}{N_0} \quad (2.9)$$

donde  $N_0$  representa la densidad espectral de potencia del ruido (W/Hz), por lo que  $\gamma_b$  es adimensional ya que  $\text{W/Hz} = \text{W s} = \text{J}$ . Es común expresar  $\gamma_b$  en dB, en cuyo caso se denota como  $\Gamma_b$ :

$$\Gamma_b = 10 \log_{10}(\gamma_b) \text{ (dB)} \quad (2.10)$$

- $\beta$ : Ganancia geométrica o factor de mérito de la constelación. Define la eficacia que tiene la constelación para separar señales aprovechando su energía, siendo un parámetro de calidad. La distancia mínima entre símbolos se representa por  $d$ .

$$\beta = \frac{d^2}{\xi_{av}} \quad (2.11)$$

- $P_E$ : Probabilidad de error:

$$P_E = \sum P_i P(s \neq s_i). \quad (2.12)$$

Este parámetro admite expresiones cerradas como las que se obtendrán para las modulaciones PAM y QAM en los apartados en términos de la función Q, siempre que

las fronteras de decisión sean paralelas a los ejes de representación de la constelación. Cuando esto no sea así, resulta muy útil obtener una medida aproximada de la  $P_E$  a través de las cotas que, en general, se pueden expresar como:

$$Q\left(\frac{d}{2\sigma}\right) \leq P_E \leq KQ\left(\frac{d}{2\sigma}\right) \quad (2.13)$$

con  $K$  número promedio de señales vecinas a una dada.

- $P_b$ : Probabilidad de error de bit. Se define como el número de bits incorrectamente recibidos en relación con el número total de bits transmitidos y, en general, se puede obtener unas cotas de la misma según:

$$\frac{1}{N}P_E \leq P_b \leq P_E \quad (2.14)$$

Además, el uso de codificación Gray para las constelaciones establece una relación entre la probabilidad de error de símbolo  $P_E$  y la probabilidad de error de bit  $P_b$  teniendo en cuenta que símbolos adyacentes solamente se diferencian en un bit. De este modo, recogemos la siguiente limitación:

$$P_b \approx \frac{P_E}{N} \quad (2.15)$$

- $\nu$ : Eficiencia espectral. Se define como el cociente entre la tasa de transmisión obtenida y el ancho de banda ocupado:

$$\nu = \frac{R}{W} \left( \frac{\text{bits}}{\text{s} \cdot \text{Hz}} \right) \quad (2.16)$$

- $\rho$ : Eficiencia en potencia. Se define como el inverso de la energía por bit normalizada para obtener una probabilidad de error dada que se usa como referencia para comparar modulaciones entre sí:

$$\rho = \frac{1}{\gamma_b} \quad (2.17)$$

Se puede demostrar que  $\gamma_b$  se encuentra acotada:

$$\frac{2}{\beta N} [Q^{-1}(P_E)]^2 \leq \gamma_b \leq \frac{2}{\beta N} \left[ Q^{-1}\left(\frac{P_E}{k}\right) \right]^2 \quad (2.18)$$

definiendo  $Q(x) = 1 - F_X(x)$  y  $Q^{-1}(\cdot)$  representa su función inversa. Además,  $k$  representa el número promedio de vecinos de cada símbolo, siendo  $P_E$  su probabilidad de error.

Además, considerando valores bajos de la probabilidad de error y una tendencia asintótica de la función  $Q^{-1}$ , se puede determinar:

$$P_E \downarrow \downarrow \left\{ \begin{array}{l} \gamma_b \approx \gamma_{b,inf} \\ g_{1,2} \approx \frac{\beta_1 N_1}{\beta_2 N_2} \end{array} \right\} \quad (2.19)$$

Al trabajar con un canal libre de interferencias, gracias al Criterio de Nyquist se puede recoger la eficiencia espectral máxima:

$$\nu_{\max} = \frac{R}{W_{\min}} = \frac{R}{2\frac{L}{T}} = 2\frac{N}{L} \quad (2.20)$$

## 2.3.- Modulaciones digitales

El proceso de modulación digital implica la modificación de alguna propiedad de una señal portadora analógica, como la amplitud, frecuencia o fase, para representar la información digital. A pesar de que existen varios tipos de modulaciones digitales, cada una con sus propias características y aplicaciones, se centrará el estudio en el análisis de las modulaciones de amplitud en cuadratura (QAM, del inglés Quadrature Amplitude Modulation), también haciendo alusión a las modulaciones de pulsos en amplitud (PAM, del inglés Pulse Amplitude Modulation).

Como bien se ha descrito previamente, se utilizarán modulaciones en dos dimensiones  $L = 2$ . Con ello, podemos ubicar las coordenadas de cada símbolo con dos características, su componente en fase denominada I, también considerada su parte real, y su componente en cuadratura Q, considerada su componente en fase.

### 2.3.1.- Modulaciones PAM

Las modulaciones de pulsos en amplitud se caracterizan por transmitir información mediante variaciones en la amplitud de una serie de pulsos. A diferencia de las modulaciones QAM, que modulan tanto la amplitud como la fase de una portadora, las modulaciones PAM se centran únicamente en la amplitud de los pulsos transmitidos. Dependiendo del convenio empleado, los pulsos pueden tener sólo polaridad positiva o positiva y negativa. En este trabajo se considera que los pulsos podrán tener ambas polaridades. En una modulación M-PAM, el número de niveles de amplitud es  $M$ , donde  $M = 2^N$ , siendo  $N$  el número de bits por símbolo.

Desde una perspectiva matemática, la señal transmitida en una modulación M-PAM puede expresarse como una secuencia de pulsos cuya amplitud toma valores discretos pertenecientes a un conjunto de  $M$  niveles (en este trabajo, esos niveles pueden ser tanto positivos como negativos). Cada símbolo transmitido puede representarse como:

$$s(t) = \sum_{n=-\infty}^{\infty} a_n p(t - nT), \quad (2.21)$$

donde  $a_n$  es la amplitud del  $n$ -ésimo pulso, que puede tomar uno de los  $M$  valores posibles,  $p(t)$  es la forma de onda del pulso (por ejemplo, un pulso rectangular), y  $T$  es el periodo de

símbolo.

Para definir una modulación M-PAM, es importante establecer los parámetros de energía media y ganancia geométrica de la constelación, desembocando en las probabilidades de error de símbolo y bit. La energía media de una señal M-PAM puede derivar en función de la distancia entre los símbolos, denotada como  $d$ . Considerando que los símbolos  $a_i$  son equidistantes y simétricos alrededor del origen, Figura 2.4, se puede determinar  $a_i = a_1 + id = -(M - 2i - 1)\frac{d}{2}$ , con  $i = 1, 2, \dots, M$ . En este punto ya es posible definir la energía media de acuerdo a la Ecuación (2.5) como sigue:

$$\xi_{av} = \frac{1}{M} \sum_{i=1}^M \left| -(M - 2i - 1)\frac{d}{2} \right|^2 = \frac{d^2}{12}(M^2 - 1) \quad (2.22)$$

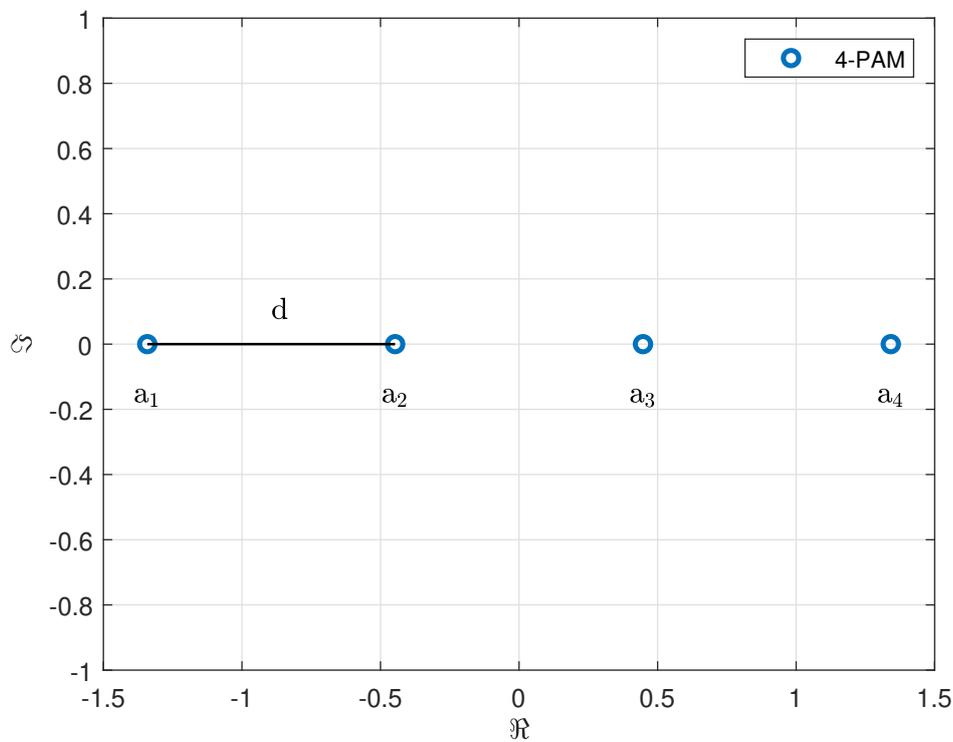


Figura 2.4.- Representación de una constelación 4-PAM, indicando la distancia entre símbolos.

A continuación, la ganancia geométrica de una constelación M-PAM, Ecuación (2.11), se deduce de la siguiente forma:

$$\beta_{M-PAM} = \frac{d^2}{\frac{d^2}{12}(M^2 - 1)} = \frac{12}{(M^2 - 1)} \quad (2.23)$$

Además, la probabilidad de error de símbolo en una modulación M-PAM se puede aproximar mediante la cota superior de la relación señal a ruido por bit, Ecuación (2.18).

De esta forma, la probabilidad de error quedaría en función de  $k$ , siendo esta una constante que indica el promedio de símbolos vecinos de la constelación. Este concepto se refiere a aquellos símbolos que están ubicados a distancias mínimas entre sí en el espacio de señal. Así:

$$P_{E,M-PAM} = 2 \left(1 - \frac{1}{M}\right) Q \left( \sqrt{\frac{6\gamma_b N}{M^2 - 1}} \right) \quad (2.24)$$

donde  $Q(x)$  es la función Q,  $\gamma_b$  es la relación señal a ruido por bit, y  $N$  es el número de bits por símbolo. Utilizando codificación Gray, probabilidad de error de bit se puede aproximar según la Ecuación (2.15).

La eficiencia en potencia para una modulación M-PAM, considerando bajas probabilidades de error, se puede expresar como:

$$\rho_{M-PAM} \approx \frac{6N}{(M^2 - 1)[Q^{-1}(P_E)]^2} \quad (2.25)$$

Finalmente, la eficiencia espectral máxima de una modulación M-PAM es:

$$\nu_{\max,M-PAM} = 2N \quad (2.26)$$

Mientras que la modulación QAM, descrita a continuación, aprovecha tanto la fase como la amplitud para transmitir información, PAM se limita a la amplitud, lo que hace que PAM sea más simple de implementar. Sin embargo, PAM puede ser útil en aplicaciones donde la simplicidad del receptor es crucial, o en sistemas donde la fase de la señal puede no ser estable. Además, serán de interés para el análisis de resultados.

### 2.3.2.- Modulaciones QAM

Las modulaciones de amplitud en cuadratura se distinguen por transportar información tanto en la fase como en la amplitud de la señal transmitida. En estas modulaciones, los símbolos se distribuyen sobre los nodos de una rejilla cuadrada, con su centro coincidiendo en el origen de coordenadas. Siguiendo esta premisa, si  $N$  es un número par, la constelación QAM correspondiente tendrá un número de señales igual a una potencia de cuatro ( $M = 4^N$ ), dando lugar a la forma típica de este tipo de constelación, un cuadrado. En este estudio se valorará el comportamiento de las constelaciones compuestas por 4 y 16 símbolos, es decir, 4-QAM o QPSK y 16-QAM.

Desde una perspectiva matemática, es posible expresar la envolvente compleja de la señal transmitida utilizando una modulación M-QAM según la Ecuación 2.2. La secuencia a la salida del Modulador de Señal  $A[m]$ , se define conforme la ecuación  $A[m] = A_i[m] + jA_Q[m]$ . Cada una de sus componentes es una secuencia de variables aleatorias independientes  $\{a_i\}_{i=1,2,\dots,M}$  representando  $a_i$  una de las dos coordenadas que pueden tomar

los posibles símbolos. Las coordenadas de una M-QAM se escogen con el objetivo de simplificar la detección, alojando las señales en los nodos de una rejilla cuadrada. Sin embargo, lo ideal, desde el punto de vista energético, sería que las señales se dispusieran en una rejilla hexagonal y que la envolvente de la constelación fuera una circunferencia, no un rectángulo. Debido a la aparición del Demodulador de Canal, la envolvente compleja se desplaza a la frecuencia central de la portadora  $f_c$  con una fase aleatoria  $\theta$ , obligando a incorporar un sistema de recuperación de fase en el receptor.

Tal y como se ha procedido anteriormente en las modulaciones PAM, se puede definir la energía media de la constelación como sigue:

$$\xi_{av} = \frac{d^2}{6}(M - 1) \quad (2.27)$$

Así mismo, la obtención del parámetro de ganancia geométrica para esta modulación también es directa:

$$\beta_{M-QAM} = \frac{6}{M - 1} \quad (2.28)$$

Por otra parte, teniendo en cuenta que una M-QAM cuadrada se compone de apilar  $\sqrt{M}$  constelaciones  $\sqrt{M}$ -PAM, es posible obtener la expresión de la probabilidad de error de símbolo de M-QAM a partir de la de  $\sqrt{M}$ -PAM de la siguiente forma:

$$P_{E,M-QAM} = 1 - P_{A,M-QAM} = 1 - P_{A,\sqrt{M}-PAM}^2 = 1 - [1 - P_{E,\sqrt{M}-PAM}]^2 \quad (2.29)$$

Teniendo en cuenta que se ha definido previamente la probabilidad de error de una modulación PAM:

$$P_{E,M-QAM} = 1 - \left[ 1 - \frac{2\sqrt{M} - 2}{\sqrt{M}} Q \left( \sqrt{\frac{3\gamma_b N}{M - 1}} \right) \right]^2 \quad (2.30)$$

con Codificación Grey  $P_b \approx \frac{P_E}{N}$

Considerando valores bajos de la probabilidad de error, se obtiene la eficiencia en potencia:

$$\rho_{M-QAM} \approx \frac{3N}{(M - 1)[Q^{-1}(P_E)]^2} \quad (2.31)$$

Finalmente, es directo obtener la máxima eficiencia espectral:

$$\nu_{\max,M-QAM} = N \quad (2.32)$$

La modulación PAM puede considerarse un caso especial de la modulación QAM en la que no existe componente en cuadratura. En QAM, los símbolos se distribuyen en un plano complejo, mientras que en PAM, los símbolos se distribuyen únicamente en una

dimensión real (eje de amplitud). En particular, una modulación 2-PAM es equivalente a una modulación BPSK (Binary Phase Shift Keying), y una modulación 4-PAM puede verse como una proyección de una constelación 16-QAM sobre uno de sus ejes.

### 2.3.3.- Modulaciones propias e impropias

Junto con la anterior clasificación de las modulaciones estándar en PAM y QAM, que son algunas de las que se han utilizado tradicionalmente, existe otra clasificación en función de ciertas propiedades estadísticas de la constelación asociada. Así, es posible clasificar las modulaciones o, mejor dicho, sus constelaciones asociadas como propias e impropias. Por ejemplo, la modulación M-PAM es impropia (totalmente impropia, de hecho), mientras que M-QAM es propia. Aunque las constelaciones propias tienen, en general, una mayor eficiencia en potencia para el mismo número de señales que las impropias, recientemente, se ha estado estudiando el papel que pueden desempeñar las impropias en escenarios donde la interferencia juega un papel importante en las perturbaciones a las que se ve sometida la comunicación. Esta sección revisa el concepto de modulaciones impropias, su fundamentación teórica y cómo generarla a partir de una constelación propia (QAM) [1]. Para comprender las modulaciones impropias, es esencial primero definir su creación y utilidad.

Toda constelación comprende un conjunto de símbolos complejos. Una señal compleja se considera propia (o circularmente simétrica) si sus componentes real e imaginaria son estadísticamente independientes y tienen varianzas iguales. Esto implica que la señal es invariante bajo rotaciones en el plano complejo. En términos matemáticos, una señal es propia cuando cumple que  $E[V^2] = 0$ , donde  $E[\cdot]$  denota el valor esperado. En contraste, una señal impropia es aquella en la que existe una correlación entre las componentes real e imaginaria, lo que se traduce en una asimetría estadística. Esta impropiedad se cuantifica mediante el coeficiente de circularidad:

$$\kappa = \frac{|E\{X^2\}|}{E\{|X^2|\}} = \frac{|\tilde{\sigma}_X^2|}{\sigma_X^2} \quad (2.33)$$

Para señales propias,  $\kappa = 0$ , mientras que para señales impropias,  $0 < \kappa \leq 1$ , Figura 2.5. Esta asimetría puede ser explotada para mejorar el rendimiento en ciertos sistemas de comunicación. Así, por ejemplo, es posible comprobar que cualquier constelación M-QAM cuadrada ofrece  $\kappa = 0$  (constelación propia) y que M-PAM lleva asociada  $\tilde{\sigma}_X^2 = \sigma_X^2$ , por lo que  $\kappa = 1$  es una constelación totalmente impropia.

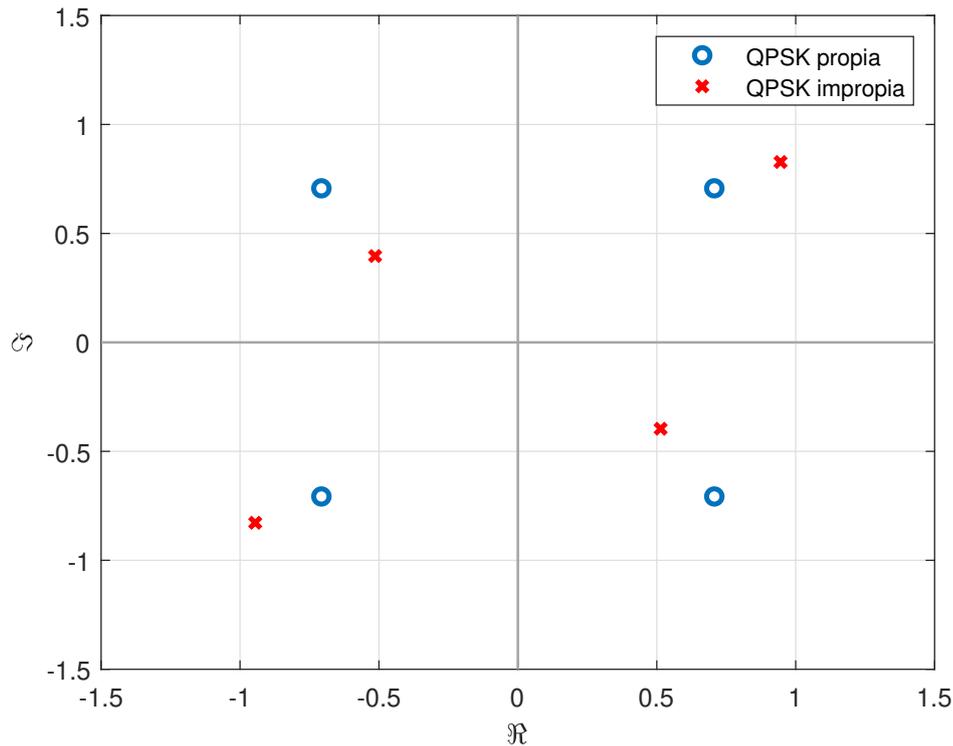


Figura 2.5.- Representación de una constelación propia (de tipo QPSK), mediante círculos azules, frente a su versión impropia (con  $\kappa = 0.6$ ), mediante cruces rojas.

La señalización mediante constelaciones impropias ofrece ventajas en situaciones donde la interferencia y la eficiencia espectral son destacables. En canales de comunicación con alta interferencia, las constelaciones propias pueden no ser óptimas debido a su simetría inherente. Las constelaciones impropias, al permitir una mayor flexibilidad en la manipulación de las propiedades estadísticas de la señal, pueden ayudar a mitigar la interferencia. Por ejemplo, en sistemas de radio, el uso de señales impropias permite al usuario secundario transmitir sin degradar significativamente la tasa de transmisión del usuario primario. Esta capacidad para gestionar la interferencia de manera más eficiente se traduce en una mejora general del rendimiento del sistema.

El diseño de las constelaciones impropias implica la disposición óptima de puntos en el plano complejo para maximizar el rendimiento del sistema de comunicación. Una de las técnicas más efectivas para lograr esto es el empaquetamiento hexagonal. Este método se basa en la disposición de puntos en una red hexagonal, que es conocida por su eficiencia en el uso del espacio. Dicho empaquetamiento es óptimo en términos de densidad, traduciéndose en una minimización de la energía media de la constelación dado su número de señales y la distancia mínima entre símbolos de la misma. Esta minimización es de especial importancia para reducir la tasa de error de símbolo, provocando una disminución en la probabilidad de errores de decodificación causados por la proximidad entre puntos de la constelación.

De forma visual, el proceso comienza encontrando una elipse rotada que interseca la red hexagonal, conteniendo exactamente  $M$  puntos de la red. Este enfoque inicial proporciona una aproximación cercana al coeficiente de circularidad deseado [2] [3].

No obstante lo anterior, existe una forma sencilla de obtener una constelación impropia con un coeficiente de circularidad deseado a partir de una constelación propia denominada Transformación Lineal Amplia (WLT). La WLT es una técnica matemática que permite transformar una señal propia en una señal impropia.

Con el objetivo de analizar dicha técnica, este estudio centra el diseño de constelaciones impropias para minimizar la SER mediante el uso de la WLT debido a su menor complejidad de implementación y, por lo tanto, mayor practicidad. A continuación, se mostrarán los procesos oportunos para la creación de dicha constelación óptima a partir de modulaciones propias.

Es destacable el uso del criterio de decisión de máxima verosimilitud (ML), donde se asume que cada símbolo dentro del conjunto de señales tiene una probabilidad igual de ser transmitido, es decir,  $1/M$ . Aunque esta suposición no maximiza la información mutua, simplifica significativamente el enfoque en la selección de puntos de la señal y es una práctica común en la mayoría de las implementaciones reales.

Partiendo de una modulación M-QAM con energía unitaria denotada  $X = X_I + jX_Q$ , se puede determinar la constelación impropia:

$$V = V_I + jV_Q = h_1X + h_2X^* , \quad (2.34)$$

donde  $h_1 = \sqrt{0.5(1 + \alpha)}$  y  $h_2 = \sqrt{0.5(1 - \alpha)}e^{j\phi}$ . Además, se define  $\alpha = \sqrt{(1 - k^2)}$  junto con una fase óptima  $\phi^* \in [0, \pi/2]$ , escogida para maximizar la distancia mínima entre puntos de la constelación. Considerando  $0 < \kappa \leq 0.5$ , la fase óptima para cualquier número de símbolos se obtiene como  $\phi^* = \pi/2$ . Sin embargo, una vez superado este umbral, se debe calcular el valor:

$$\phi^* = \arccos \left( \frac{-1 + 2\sqrt{5\kappa^2 - 1}}{5\kappa} \right) \in [0, \pi/2] \quad (2.35)$$

Por un lado, si la constelación elegida es QPSK, es decir,  $M = 4$ , se puede calcular el ángulo de fase óptimo para valores del coeficiente de circularidad  $0.5 \leq \kappa \leq 1$ . En cambio, para valores de  $M > 4$ , se podrá determinar el el valor de  $\phi^*$  siempre y cuando  $0.5 \leq \kappa \leq 2/\sqrt{5}$ . En el caso de que se necesite  $\kappa > 2/\sqrt{5}$ , se recurre a un algoritmo iterativo, no abarcado en este estudio, que analiza todas las distancias posibles entre pares de puntos de la constelación.

Finalmente, la WLT se puede describir de forma matricial como:

$$\begin{bmatrix} V_I \\ V_Q \end{bmatrix} = \begin{bmatrix} h_1 + |h_2|\cos(\phi) & |h_2|\sin(\phi) \\ |h_2|\sin(\phi) & h_1 - |h_2|\cos(\phi) \end{bmatrix} \begin{bmatrix} X_I \\ X_Q \end{bmatrix} \quad (2.36)$$

En los capítulos posteriores, se realizará una comparación entre constelaciones propias e impropias con el objetivo de evaluar su comportamiento en términos de probabilidad de error. La comparación se llevará a cabo mediante simulaciones y análisis teóricos, que permitirán identificar las ventajas y desventajas de cada tipo de constelación en diferentes escenarios de comunicación. Esta evaluación es fundamental para entender cómo estas constelaciones pueden influir en el rendimiento de los sistemas de comunicaciones y en la calidad del servicio ofrecido a los usuarios.

## 3. Herramientas utilizadas

En esta sección se explicarán las herramientas utilizadas para posteriormente entender la implicación que han tenido en el desarrollo del estudio. Fundamentalmente el TFG se ha apoyado en: Matlab, GNURadio y USRP. Las dos primeras herramientas confeccionan la parte software del proceso, mientras que los dispositivos USRP se corresponden con el comportamiento en el mundo real, hardware.

### 3.1.- Matlab

MATLAB, acrónimo de Matrix Laboratory, es un entorno de programación y un lenguaje de alto nivel ampliamente utilizado para el análisis numérico, la visualización de datos, la simulación y el desarrollo de algoritmos [4].

Es conocido por ser un lenguaje fácil de usar y aprender. Su sintaxis es intuitiva y permite a los usuarios escribir programas y scripts de manera rápida y eficiente. A diferencia de otros lenguajes de programación, está orientado a la manipulación de matrices y vectores, lo que lo hace especialmente útil para operaciones matemáticas. Además, ofrece potentes herramientas de visualización que permiten a los usuarios crear gráficos y representaciones visuales de datos de alta calidad. Otra ventaja destacable es la capacidad de visualización gráfica de los datos tanto en 2D como en 3D, siendo fundamental para el análisis y la comunicación de resultados.

### 3.2.- GNU Radio

GNU Radio es una poderosa herramienta de código abierto que permite a los usuarios construir y experimentar con sistemas de radio definidos por software [5]. Como proyecto open source, su código fuente está disponible en GitHub, permitiendo a los desarrolladores acceder, modificar y mejorar el software de manera colaborativa. Esta naturaleza abierta facilita una actualización dinámica y constante, ya que los miembros de la comunidad global de desarrolladores pueden contribuir con nuevas funcionalidades, correcciones de errores y mejoras de rendimiento de manera continua, asegurando que GNU Radio evolucione y se adapte a las necesidades cambiantes de los usuarios.

#### 3.2.1.- Bloques y comunicación

Una de las características fundamentales de GNU Radio son los bloques, que son las unidades básicas de procesamiento en cualquier flujo de datos dentro del entorno. Los

bloques en GNU Radio se pueden considerar como pequeños programas o funciones que realizan tareas específicas, tales como la modulación, demodulación, filtrado, conversión de frecuencias, y mucho más. Estos se conectan entre sí para formar un diagrama de flujo que representa un sistema de radio completo. Cada bloque en GNU Radio tiene una entrada y una salida, lo que permite que los datos fluyan de un bloque a otro de manera secuencial. Adicionalmente, la programación de los mismos está basada en Python y C++, pudiendo ser programados en cualquiera de los dos lenguajes.

Una característica importante de GNU Radio es su integración con otras herramientas y entornos. Por ejemplo, GNU Radio Companion es una interfaz gráfica que permite a los usuarios construir diagramas de flujo de manera visual, arrastrando y soltando bloques en un lienzo y conectándolos mediante líneas. Esto no solo simplifica el proceso de diseño y configuración, sino que también permite a los usuarios visualizar el flujo de datos y depurar problemas más fácilmente. Además, existe la opción de interactuar con hardware SDR, como los dispositivos USRP, lo que permite a los usuarios transmitir y recibir señales de radio.

Los bloques en GNU Radio están diseñados para ser modulares y reutilizables, lo que significa que un elemento creado para una aplicación específica puede ser reutilizado en diferentes proyectos con poco o ningún cambio. Esto es posible gracias a la arquitectura bien definida y las interfaces estándar que se proporcionan para la creación de bloques. Para ello, ofrece una amplia biblioteca de librerías predefinidas que cubren una variedad de funciones comunes en el procesamiento de señales, lo que facilita la construcción de sistemas complejos sin tener que programar un bloque completamente nuevo. Atendiendo a distintos criterios, pueden categorizarse. Por ejemplo, una clasificación relevante para las entradas y salidas es la que distingue entre síncronas y asíncronas. Las entradas o salidas síncronas operan según una tasa de muestreo constante, es decir, demandan y producen datos a intervalos regulares. En cambio, las entradas asíncronas no dependen de un intervalo fijo para recibir información, y las salidas asíncronas no generan datos en momentos predecibles. De esta manera, se deduce que no es posible conectar directamente una salida síncrona con una entrada asíncrona, y viceversa. Adicionalmente, hay que considerar la compatibilidad de los tipos de datos manejados; por ejemplo, no se puede conectar una salida que emite valores enteros (*int*) con una entrada que espera recibir valores de punto flotante (*float*).

A pesar de parecer una tarea trivial, se ha de configurar complementariamente los parámetros de cada bloque, pudiendo conllevar dificultades. Apoyándose en la herramienta gráfica GRC, esta tarea se hace de forma visual e intuitiva, incrementando la eficiencia del usuario.

### 3.2.2.- Conexionado

La conexión entre estos bloques se realiza en el entorno GRC, consiguiendo que los flujos de datos se visualicen en dicho entorno en la herramienta gráfica, donde cada bloque se

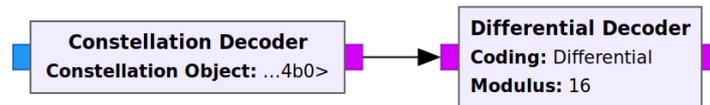


Figura 3.1.- Ejemplo de conexión síncrona entre bloques de GNU Radio.

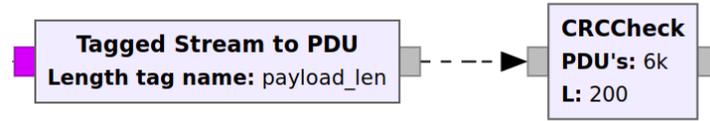


Figura 3.2.- Ejemplo de conexión asíncrona entre bloques de GNU Radio.

caracteriza con puertos de entrada y salida que permiten el flujo de datos.

Las conexiones en GNU Radio se dividen en dos categorías principales: conexiones de datos síncronas y conexiones asíncronas. Las síncronas, Figura 3.1, son las más comunes y permiten el flujo continuo de datos entre bloques. Estas se representan por líneas sólidas en GRC y son ideales para procesos en tiempo real, ya que los datos fluyen de manera continua y sin interrupciones. Un tipo particular de conexión síncrona incluye las etiquetas de flujo, stream tags. Estas etiquetas son metadatos asociados a muestras específicas dentro del flujo de datos y permiten añadir información adicional como complemento. Los bloques pueden generar, modificar o interpretar estas etiquetas según sea necesario.

Por otro lado, las conexiones asíncronas, Figura 3.2, permiten la comunicación entre bloques de manera no determinista, siendo útiles para señales de control o para la comunicación con aplicaciones externas. Se representan con líneas discontinuas. A diferencia de las conexiones síncronas, los mensajes en conexiones asíncronas no están necesariamente alineados temporalmente con las muestras de datos, lo que introduce cierta variabilidad en el procesamiento. Para este tipo de enlaces, es importante considerar la latencia y el tiempo de respuesta del sistema. La asincronía puede introducir variabilidad en el procesamiento, por lo que debe ser manejada cuidadosamente, especialmente en aplicaciones sensibles al tiempo.

A la hora de diseñar las conexiones entre bloques, es esencial considerar la compatibilidad de los tipos de datos. Las salidas de un bloque deben ser compatibles con las entradas del siguiente bloque. GNU Radio proporciona diversos tipos de datos, como flotantes, enteros y complejos, presentados en la Figura 3.3. Además, es importante gestionar la tasa de muestreo de manera coherente entre bloques conectados directamente. También, comprender cómo se propagan y utilizan las etiquetas de flujo puede optimizar la gestión de metadatos en el sistema. Los bloques deben estar configurados para manipular estas etiquetas adecuadamente.

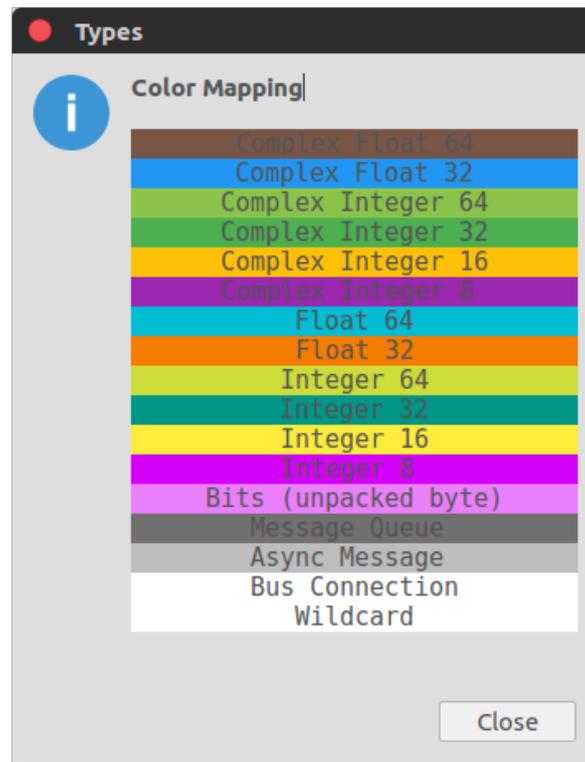


Figura 3.3.- Tipos de datos soportados por los bloques de GNU Radio.

### 3.2.3.- Sistemas utilizados

Para familiarizarse con la herramienta gráfica y los bloques de GNU Radio, se comenzará analizando distintos sistemas propuestos [6]. El primero de ellos simula una comunicación monoportadora utilizando una modulación QPSK, mientras que el segundo presenta una mayor complejidad utilizando multiplexación por división de frecuencias ortogonales. Estos esquemas se detallarán con el objetivo de comprender la utilidad de GNU Radio en comunicaciones, sirviendo como un proceso de introducción y aprendizaje de la herramienta. Para finalizar la sección, se expone una explicación detallada sobre el sistema final, objetivo último de este TFG.

#### 3.2.3.1.- Sistema auxiliar: QPSK monoportadora

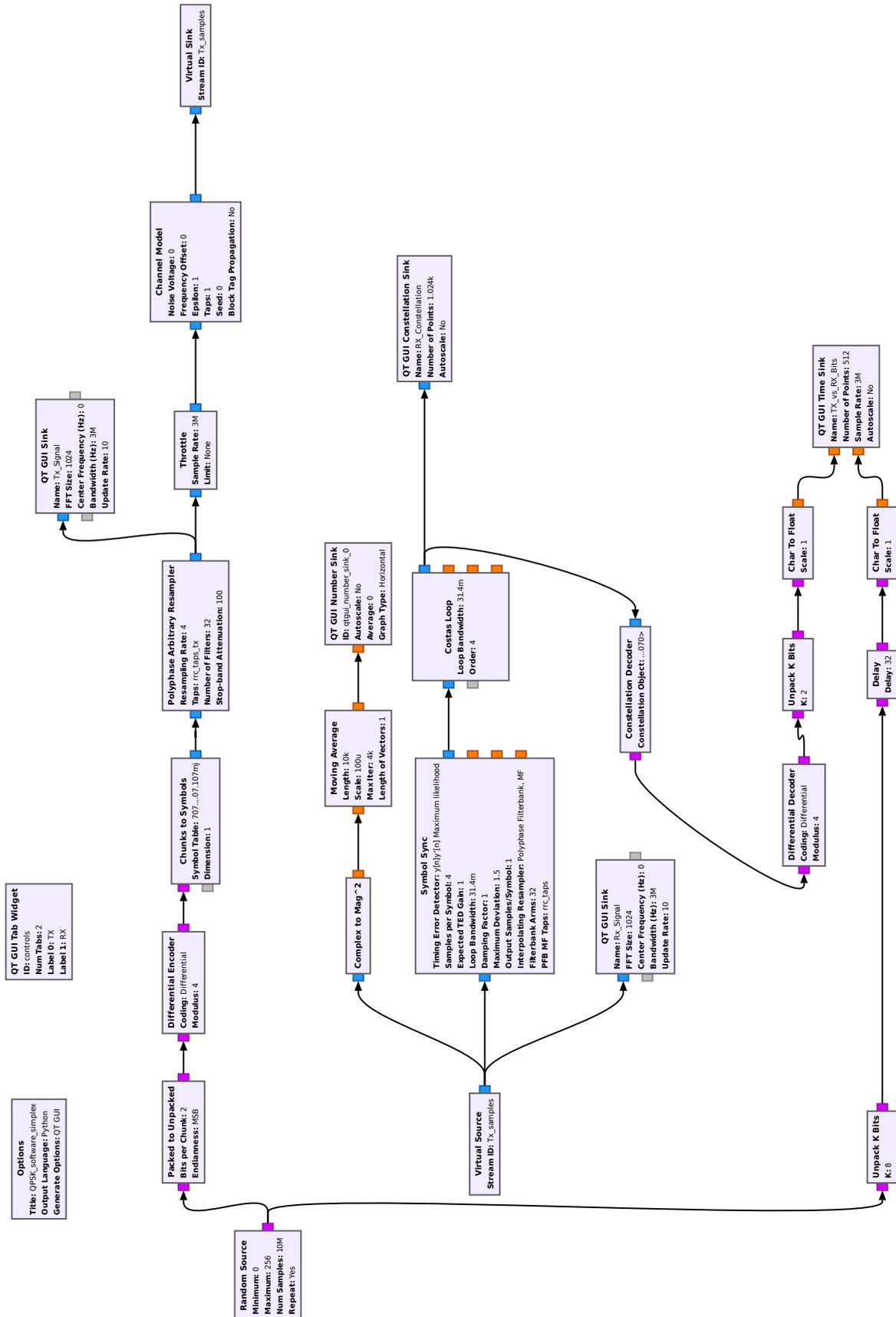


Figura 3.4.- Diagrama de bloques de sistema de comunicación monoportadora con modulación QPSK.

La modulación QPSK es una técnica que permite la transmisión eficiente de datos mediante variaciones de fase de una señal portadora. En el contexto de GNU Radio, el diseño de un sistema QPSK implica una serie de pasos que abarca desde la generación de datos hasta su transmisión y recepción, permitiendo comprender la utilidad de los bloques que lo componen. Debido a la posterior explicación en profundidad de sistemas más complejos, se ha decidido no entrar en detalle en el propósito de cada bloque en esta subsección. Aún así, este esquema ha sido de gran utilidad para comprender y familiarizarse con el uso de la herramienta gráfica de GNU Radio y sus características.

El proceso comienza con la generación de datos, donde se crean secuencias de bits que simulan la información que se desea transmitir. Estos datos pasan por una serie de transformaciones y codificaciones necesarias para prepararlos para la modulación. En GNU Radio, este paso se facilita mediante el uso de un bloque de fuente aleatoria que genera una secuencia continua de datos. A continuación, la modulación QPSK convierte estos datos digitales en una señal analógica modulada, adecuada para la transmisión. En este propósito intervienen los bloques situados en la fila superior de la figura donde, la mayoría, describen su utilidad con el propio nombre. Se puede destacar la aparición del bloque *Throttle*, el cual recoge las muestras de su entrada para emitirlas a una tasa indicada, teniendo así controlada la velocidad de transmisión de las tramas.

Para simular el canal de comunicación, GNU Radio permite introducir diversos efectos como ruido y desplazamientos de frecuencia, replicando las condiciones reales que afectan la señal transmitida. Esta capacidad es importante para probar y ajustar el sistema en un entorno controlado, asegurando que pueda manejar las interferencias y distorsiones comunes en las comunicaciones.

En la recepción, el sistema debe recuperar los datos originales a partir de la señal modulada recibida. Esto implica la demodulación, donde se invierten las transformaciones aplicadas durante la modulación y transmisión. La sincronización de fase es un aspecto crítico en esta etapa, ya que garantiza que los datos recuperados sean precisos y coherentes con los datos transmitidos.



OFDM es una técnica de modulación que descompone una señal de alta tasa de datos en varias señales de menor tasa, las cuales se transmiten simultáneamente sobre diferentes frecuencias portadoras ortogonales. Esta técnica mejora significativamente la resistencia al desvanecimiento y a la interferencia de multitrayecto, problemas comunes en las comunicaciones inalámbricas. En combinación con 16-QAM, donde cada portadora puede transmitir uno de 16 símbolos diferentes, se logra una mayor eficiencia espectral, permitiendo transmitir más información en el mismo ancho de banda.

El sistema comienza con la generación de datos utilizando un bloque de fuente aleatoria. El bloque *Random Source* en GRC es esencial para simular los datos que se transmitirán. Este bloque genera una secuencia de números aleatorios dentro de un rango especificado. En este caso, los parámetros importantes incluyen un mínimo de 0, un máximo de 255, un número de muestras de 1000 y la opción de repetición activada. Esta configuración asegura una fuente constante de datos que representan los bits de información que serán modulados y transmitidos.

A continuación, se reempaquetan los bits para adaptarlos al formato requerido por el modulador. Esto se logra utilizando un bloque que reorganiza los bits de entrada para crear secuencias de bits adecuadas para la modulación 16-QAM. Específicamente, se configura para convertir bytes de 8 bits en palabras de 2 bits. El bloque *Chunks to Symbols* tiene como objetivo realizar la modulación 16-QAM. Este toma las palabras de bits y las convierte en símbolos que representan puntos en la constelación 16-QAM. La tabla de símbolos se configura para mapear cada combinación de bits a un punto específico en la constelación, que es una representación de los diferentes niveles de amplitud y fase posibles.

Una vez que los datos están modulados en símbolos, se utilizan bloques OFDM para preparar la señal para la transmisión. El bloque *OFDM Carrier Allocator* asigna los símbolos modulados a las subportadoras OFDM. Los parámetros importantes aquí incluyen la longitud de la FFT, los portadores ocupados, las portadoras piloto y los símbolos piloto. Estas portadoras y símbolos piloto ayudan en la sincronización y la estimación del canal en el receptor. Seguidamente, se añade un prefijo cíclico a cada símbolo OFDM, el cual es una copia de la última parte del símbolo y se coloca al principio del símbolo para combatir la interferencia entre símbolos, ISI. Los parámetros clave incluyen la longitud de la FFT y la longitud del prefijo cíclico.

Para simular el entorno de transmisión, se utiliza el bloque *Channel Model*, capaz de introducir ruido y otros efectos del canal en la señal. Los parámetros configurables incluyen el voltaje de ruido, el desfase de frecuencia, el factor de escala y los coeficientes de los taps del canal. Inicialmente, el voltaje de ruido se puede configurar a cero para observar la señal en un canal ideal, pero es ajustable para simular diferentes niveles de ruido e interferencia.

En la etapa de recepción, la señal OFDM debe ser sincronizada, igualada y demodulada para recuperar los datos originales. El bloque *OFDM Synchronization* es responsable de

sincronizar la señal recibida con la señal transmitida, utilizando los portadores y símbolos piloto, al igual que sucedía en bloques anteriores. Tras la sincronización, el bloque *OFDM Frame Equalizer* compensa las distorsiones introducidas por el canal, utilizando los símbolos piloto para estimar y corregir las variaciones en la amplitud y fase de las subportadoras. Entre los parámetros destacables de este bloque, se encuentra la longitud de la FFT. El bloque *OFDM Serializer* convierte las señales de las subportadoras de vuelta a una secuencia de símbolos. Este bloque es de especial importancia para reconstruir los datos en su forma original antes de la demodulación gracias al uso previo de la FFT y las portadoras ocupadas.

Finalmente, el bloque *Constellation Decoder* demodula los símbolos 16-QAM recibidos para recuperar los bits de datos originales. Este bloque utiliza un objeto de constelación predefinido para mapear los puntos de la constelación de vuelta a combinaciones de bits. La correcta configuración de este bloque es esencial para una recuperación precisa de los datos.

Para analizar y verificar el funcionamiento del sistema, se utilizan varios bloques de visualización en GRC. El bloque *QT GUI Constellation Sink* permite observar la constelación de símbolos 16-QAM, lo cual es útil para verificar que la demodulación se está realizando correctamente. Cualquier dispersión o desviación en los puntos de la constelación puede indicar problemas de ruido o sincronización. El bloque *QT GUI Time Sink* muestra las señales en el dominio del tiempo, proporcionando una vista detallada de la forma de onda de la señal transmitida y recibida. Esto es útil para observar la integridad de la señal y detectar posibles problemas de ISI o de sincronización. El bloque *QT GUI Frequency Sink* muestra el espectro de la señal, lo que es fundamental para analizar la ocupación del espectro y la presencia de ruido. Este análisis espectral ayuda a identificar y mitigar problemas de interferencia y a optimizar el uso del ancho de banda disponible.

A pesar de que esta técnica no se utilice en el sistema de simulación final, la implementación de un sistema OFDM con 16-QAM proporciona una comprensión profunda de los principios de transmisión y recepción, así como de los efectos del canal de comunicación. A través de la configuración y ajuste de los bloques en GRC, se ha adquirido la capacidad de visualizar y analizar cada etapa del proceso de transmisión y recepción siendo muy útil para entender cómo se comportan los sistemas de comunicaciones en diferentes condiciones de canal y su implementación en GNU Radio.

### 3.2.4.- Sistema de comunicaciones digitales en GNU Radio

En esta sección se realizará una analogía detallada entre el diagrama de bloques que representa el sistema final, Figura 3.7, y un sistema de comunicaciones digitales básico, expresado en Figura 2.1. Este análisis tiene como objetivo descomponer y explicar el funcionamiento de cada componente del sistema, comparando sus funciones con las etapas típicas de un sistema de comunicaciones digitales.

La Figura 3.7 representa un sistema de procesamiento de señales implementado me-

diante bloques funcionales interconectados, los cuales cumplen funciones específicas que pueden ser relacionadas con las etapas de transmisión, modulación, procesamiento de señal, y recepción en un sistema de comunicaciones digitales. Cada bloque será detallado a continuación, describiendo su propósito y su análogo en un sistema de comunicaciones.

El bloque representado en la Figura 3.6 es fundamental en un sistema donde se implementan modulaciones impropias, ya que define la estructura de la constelación utilizada en el sistema de comunicación. Específicamente, este bloque describe una *Constellation Type* variable, lo que indica su flexibilidad para adaptarse a diferentes esquemas de modulación, esencial en modulaciones impropias donde las constelaciones no son fijas ni estándar. La característica *Symbol Map* proporciona el mapeo de símbolos, vital para la correcta interpretación y decodificación de los datos transmitidos. Por otra parte, los *Constellation Points* determinan las posiciones exactas de los símbolos en el espacio de constelación, siendo esta característica la que hay que ajustar para poder generar las constelaciones impropias deseadas (por ejemplo, una constelación QPSK impropia con  $\kappa = 0.8$  debería de indicarse en este campo como  $[1.063 + 0.752j, 0.512 - 0.201j, -0.512 + 0.201j, -1.063 - 0.752j]$ ). Además, la "Soft Decisions Precision" de 8 bits permite una mayor resolución en la toma de decisiones suaves, mejorando la capacidad del sistema para manejar ruidos e interferencias, aspectos críticos en modulaciones impropias. Las demás características de este bloque no son críticas para esta implementación.

Constellation Object	
<b>ID:</b>	const
<b>Constellation Type:</b>	Variable Constellation
<b>Symbol Map:</b>	[0,1,2,3,4,5,6...
<b>Constellation Points:</b>	pcon...
<b>Rotational Symmetry:</b>	1
<b>Dimensionality:</b>	1
<b>Normalization Type:</b>	None
<b>Soft Decisions Precision:</b>	8

Figura 3.6.- Bloque *Constellation Object* de GNU Radio.

Habiendo definido la constelación, el proceso de comunicación inicia con la generación de datos digitales, representada por los bloques de *Random Source* y *Signal Source*. En un sistema de comunicaciones, estos bloques son equivalentes a la fuente de información que se desea transmitir. La fuente aleatoria simula la generación de bits de datos, mientras que la fuente de señal puede representar una señal continua. El siguiente paso es el procesamiento previo de la señal, donde intervienen bloques como *Polyphase Arbitrary Resampler*, *Throttle*, y *Multiply*. Estos bloques son responsables de preparar la señal para su transmisión. En un sistema de comunicaciones digitales, este proceso se podría conocer como acondicionamiento de la señal, pudiendo formar parte de la modulación. El *Resampler* ajusta la tasa de

muestreo de la señal, equivalente a adaptar la señal a las características del canal de comunicación, siendo el *Throttle* el encargado de controlar la velocidad de procesamiento, similar a regular la tasa de transmisión de bits.

La modulación digital se representa por bloques como *Chunks to Symbols* y *Constellation Modulator*. En un sistema real, serán los encargados de convertir un grupo de bits en puntos de la constelación, representados por un número complejo. En este estudio solamente se utilizarán modulaciones de tipo QAM, pero podrían variar dependiendo de la configuración específica e intereses del usuario. Posteriormente, la transmisión de la señal modulada se realiza a través de un medio o canal, representado en el diagrama por la interconexión entre bloques de procesamiento de señales. En un sistema de comunicaciones, este elemento puede introducir ruido y distorsiones, los cuales pueden ser simulados mediante bloques de adición de ruido y filtros. De esta forma se asegura que la señal transmitida se asemeje a una transmisión real afectada por las imperfecciones del canal.

Una vez la señal es transmitida por el canal, se encuentran dos bloques adicionales que juegan un papel fundamental en la sincronización y recepción de la información. El primero de ellos, *Symbol Sync*, es responsable de la sincronización de los símbolos, lo cual es esencial para la correcta demodulación de la señal. En un sistema de comunicaciones digitales, la sincronización de símbolos asegura que el receptor pueda identificar correctamente el inicio y el final de cada símbolo transmitido, minimizando los errores de interpretación de la señal recibida. Este bloque ajusta las muestras de la señal recibida para alinearlas con los tiempos de los símbolos transmitidos. Además, se apoya en el bloque inmediatamente posterior, *Constellation Receiver*, para realizar la demodulación de la constelación de la señal, comparando las señales recibidas con los puntos de la constelación predefinida para determinar los símbolos transmitidos. Este proceso es fundamental para la correcta decodificación de los datos transmitidos en el *Constellation Decoder*.

Se finaliza el sistema con la línea de bloques inferior, encargada del procesamiento final de la señal para asegurar que los datos originales sean recuperados de manera precisa. El *Differential Decoder* es responsable de decodificar la señal que ha sido modulada diferencialmente, lo que implica convertir los cambios entre símbolos sucesivos de vuelta a su forma original, bits. A continuación, se toman los símbolos recibidos y se descomponen en bits individuales gracias al uso del *Unpack Bits*, preparando los datos para la correlación de códigos de acceso. Los bloques finales se encargan de comparar los bits recibidos con un código de acceso conocido para detectar el comienzo de un paquete de datos y asegurarse de que se ha recibido la información correcta. Una vez identificado el paquete, se reagrupan los bits en tramas más grandes adecuadas para su procesamiento posterior. Finalmente, el *Tagged Stream to PDU* convierte el flujo continuo de datos etiquetados en Unidades de Datos de Protocolo, PDU, que son estructuras de datos que pueden ser fácilmente manejadas por los niveles superiores del sistema de comunicaciones. Esta cadena de bloques asegura

que la información transmitida sea recibida, decodificada y formateada correctamente para su análisis posterior.

Se ha podido observar como la flexibilidad de GNU Radio como plataforma de SDR proporcionan un entorno ideal para la investigación y el desarrollo en comunicaciones digitales. Además de los bloques estándar para modulación y demodulación, el software admite la integración de algoritmos personalizados, tratándose en la siguiente sección, y la adaptación rápida a nuevos estándares y requisitos de aplicación. Esto facilita la experimentación con nuevas técnicas de procesamiento de señales y la evaluación comparativa de diferentes enfoques de diseño, resultando de utilidad y efectivo.

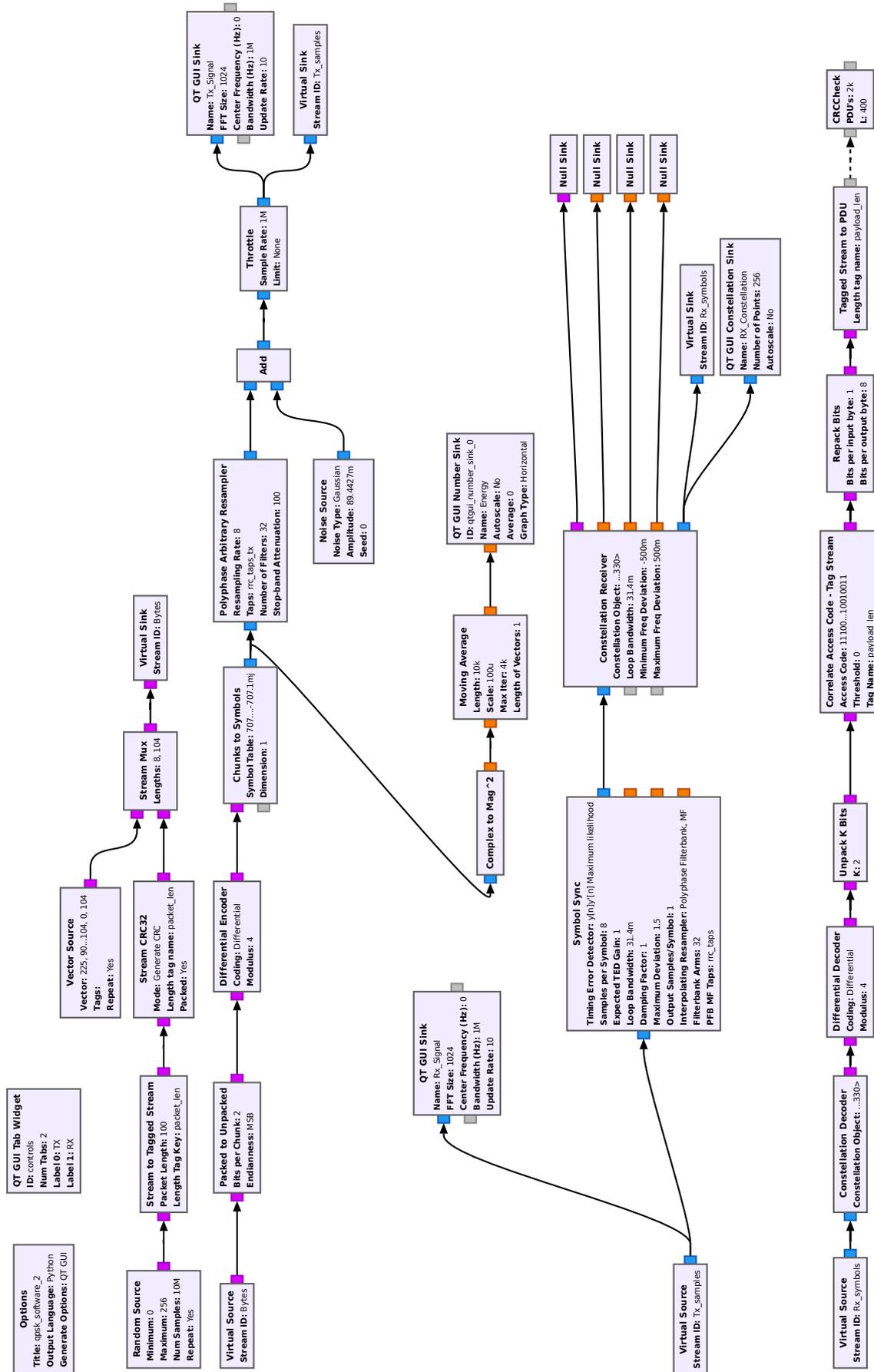


Figura 3.7.- Diagrama de bloques del sistema final utilizado para el análisis resultados.

### 3.2.5.- Bloque de cálculo de SER

Durante el desarrollo del TFG, además de utilizarse distintos bloques y sistemas ubicados en la librería por defecto de GNU Radio, también se decidió programar un bloque personal capaz de cumplir los requerimientos necesarios. Este bloque tendría como propósito detectar las tramas erróneas recibidas durante la comunicación, así como calcular la probabilidad de error (SER) de símbolo según la modulación escogida.

Para su desarrollo, se escogió como punto de partida un bloque estándar proporcionado por la librería de GNU Radio denominado Async CRC32, cuyo propósito es recoger el resultado de la comprobación de redundancia cíclica de una secuencia de datos (trama o PDU) de forma asíncrona, permitiendo la verificación de integridad en sistemas de comunicación y almacenamiento. El CRC32 es una técnica común para detectar errores en los datos, generando un valor de 32 bits que cambia con cualquier modificación en las tramas de entrada. En el proceso de transmisión, este código se añade como cabecera a cada paquete de datos con el objetivo de que, al final del proceso de recepción, el bloque asíncrono programado para validar el código CRC32 de cada paquete. En base a esta comprobación, el bloque produce estimaciones periódicas de la SER, permitiendo verificar la integridad de los datos recibidos.

El enfoque asíncrono de este bloque ofrece flexibilidad al manejar datos que no llegan de forma continua y reduce la latencia al no requerir bloques de datos de tamaño fijo para comenzar el cálculo. Esto es particularmente beneficioso en sistemas de comunicación en tiempo real y en aplicaciones donde la recepción de datos es impredecible. Además, al poder manejar datos en ráfagas, el bloque se adapta a una variedad de escenarios.

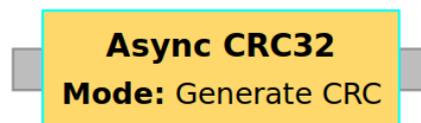


Figura 3.8.- Bloque *Async CRC32* de GNURadio.

Analizando su composición, se decidió realizar un bloque similar que siempre funcionase como comprobador y no generador, es decir, que siempre se enocontrarse en una situación de recepción. De esta manera, se sabría cuando se reciben las tramas con errores y se podría calcular de forma aproximada la probabilidad de error de la comunicación. Com-

plementariamente, se introdujeron 3 parámetros adicionales. El primer parámetro,  $PDU's$ , se corresponde con el número de tramas que se usan para hacer una estimación de la SER. Si se escoge un valor bajo, puede ser que no se represente adecuadamente la variabilidad de todo el conjunto de datos. Esto podría resultar en un sesgo, donde las características de la muestra final no reflejan las del conjunto de datos completo, provocando un cálculo de probabilidad de error irreal. Además, ha de tenerse en cuenta la longitud de las tramas y la modulación utilizada para poder considerar los errores provocados en la comunicación. Con este objetivo, se introduce el segundo parámetro  $L$ , el cual, considerando que se utiliza Codificación Grey, podemos calcular:

$$L = \frac{\text{longitud de la trama (bytes)} \times 8}{N} \quad (3.1)$$

Siendo  $N$  el número de bits utilizado por cada símbolo de la modulación. Observando la Figura 3.7, el bloque *Stream to Tagged Stream* indica la longitud en bytes de cada trama, pudiendo hacer un cálculo directo según la modulación realizada.

Tras analizar  $L$  tramas el bloque producirá la estimación de la SER como:

$$SER = \log_{10}\left(1 - \frac{10^{\log_{10}(SPR)}}{L}\right) \quad (3.2)$$

El término  $SPR$  se refiere a la tasa de acierto de paquete, la cual se obtiene como la relación de tramas obtenidas correctamente frente a tramas totales. Por ejemplo, si se reciben correctamente 980 de 1000 paquete, la  $SPR$  es 0.98.  $L$  se ha definido previamente en Ecuación 3.1. Este valor se representará por pantalla proporcionando una fácil interacción con el usuario.

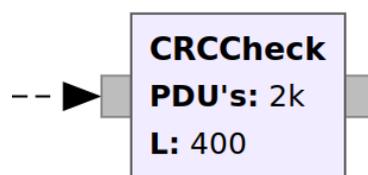


Figura 3.9.- Bloque programado para la medida de la SER en base al código CRC32.

### 3.3.- USRP

En esta sección se introducirá esta gama de dispositivos, utilizados para la implementación hardware del escenario en cuestión. Tras una introducción se desarrollará brevemente un modelo específico: USRP B210. Complementariamente, estas secciones se referirán a la compatibilidad que tienen con la herramienta software GNU Radio.

#### 3.3.1.- Introducción

Los USRP, fabricados por Ettus Research [7], representan una gama de dispositivos de radio definidos por software que integran todos los elementos de hardware necesarios para implementar aplicaciones en esta tecnología. Estos dispositivos no solo incluyen los componentes físicos del front-end y back-end para la transmisión y recepción, sino que también permiten la modificación de parámetros destacables, como la ganancia en potencia o frecuencia, dentro de un rango específico. Esta capacidad de configuración es esencial para aprovechar al máximo la versatilidad y capacidad de adaptación del paradigma SDR. La unidad central encargada de ajustar estos parámetros de forma coordinada y en tiempo real es una FPGA, la cual no solo se encarga de la configuración de los elementos de hardware, sino también de una parte del procesamiento de las muestras digitales.

Además, los USRP permiten la comunicación con una unidad de procesamiento externa, como la CPU de un ordenador personal, mediante una interfaz definida a nivel de software mediante la librería UHD, programada en los lenguajes C y C++. Esta característica, la cual se utilizará durante las simulaciones en un canal físico, aumenta la accesibilidad y versatilidad del sistema, en consonancia con el paradigma SDR.

La amplia gama de modelos de USRP, con características y precios diversos, facilita la optimización de recursos para adaptarse a las necesidades específicas de cada aplicación. La fiabilidad de los dispositivos USRP, su amplia comunidad de usuarios y la compatibilidad entre modelos, garantizada por la interoperabilidad basada en UHD, hacen de ellos una elección confiable y versátil para el desarrollo de sistemas de comunicaciones adaptativas sobre dispositivos SDR.



Figura 3.10.- USRP B210.

### 3.3.2.- Modelo USRP B210

En el desarrollo de este TFG se emplean dos dispositivos modelo USRP B210 [8], Figura 3.10. Esta sección tiene como objetivo explicar El diagrama de bloques del modelo B210 se muestra en la Figura 3.11.

En primer lugar, el bloque general de interfaz del USRP B210 se implementa mediante una conexión USB 3.0, que también proporciona alimentación al dispositivo. Aunque es posible utilizar una conexión USB 2.0, las prestaciones en términos de la tasa de muestreo máxima se ven afectadas negativamente. Los bloques responsables del control de transmisión y recepción están gestionados por un circuito integrado diseñado para la conexión USB, conocido como USB 3.0 Controller. La FPGA, que actúa como unidad de control, implementa los bloques DDC y DUC. Los bloques cuya función son de diezmado e interpolación se representan de manera independiente. Por otra parte, los convertidores ADC y DAC, junto con los elementos de radiofrecuencia como filtros, mezcladores, amplificadores, PLLs y VCOs, se implementan mediante un chip de Analog Devices. El elemento restante es un amplificador de ganancia fija al final de la etapa de transmisión.

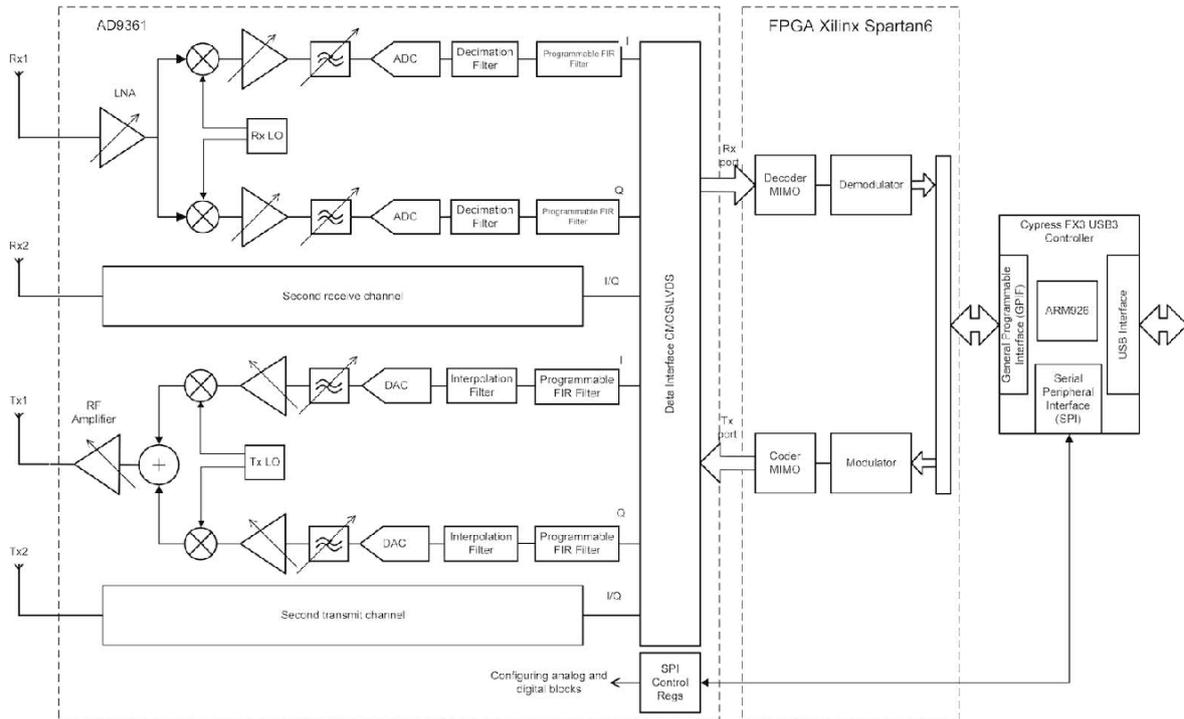


Figura 3.11.- Diagrama de bloques del funcionamiento interno del USRP.

El USRP B210 es una solución avanzada en el campo de SDR caracterizada por su interfaz FPGA-chip que proporcionan flexibilidad para el procesamiento de señales. Las líneas I/Q facilitan el intercambio de muestras digitales, permitiendo la manipulación señales complejas. La interfaz CTRL conecta la FPGA con el chip, permitiendo ajustar dinámicamente los parámetros de configuración del hardware según las necesidades específicas de la aplicación en tiempo real. Por otro lado, la generación de la señal de reloj del chip se realiza mediante un oscilador TCXO, cuya estabilidad y precisión son gestionadas por la FPGA a través de un DAC. Esta señal no solo sincroniza internamente las operaciones del dispositivo, sino que también se utiliza como referencia común para garantizar la coherencia y la sincronización con otros sistemas externos. Complementando estas funciones, las interfaces REF IN y GPIO permiten la sincronización y la conexión con dispositivos externos, mientras que la interfaz JTAG facilita la programación y depuración de la FPGA de manera efectiva conforme al estándar JTAG.

En cuanto a sus especificaciones técnicas, el USRP B210 destaca con una tasa máxima de muestreo de 61.44 MS/s tanto para su convertidor analógico-digital, ADC, como para el convertidor digital-analógico, DAC, ofreciendo una resolución de 12 bits. Su diseño full-duplex permite que los dos transmisores y receptores operen simultáneamente, lo cual es ideal para aplicaciones que requieren la transmisión y recepción de señales en tiempo real sin compromisos de rendimiento. En términos de rendimiento RF, el USRP B210 garantiza una potencia de salida máxima de 20 dBm y una potencia de entrada de hasta -15 dBm, ase-

gurando una compatibilidad con sistemas externos adaptados a  $50 \Omega$ . La ganancia disponible del transmisor puede alcanzar hasta 76 dB, mientras que la del receptor es de 89.8 dB, lo que permite ajustar la sensibilidad y el rango dinámico para adaptarse a diversas condiciones de señal y entorno.

Destacando en la recepción de señales, el dispositivo mantiene una figura de ruido por debajo de los 8 dB. Además, su punto de intercepción de tercer orden a la entrada se encuentra en -20 dBm. En aplicaciones que requieren precisión en la fase de la señal, el USRP B210 ofrece un bajo nivel de ruido de fase, con valores RMS de 1 grado medidos a 3.5 GHz y 1.5 grados a 6 GHz, lo cual es crucial para aplicaciones que dependen de la coherencia y sincronización precisa de las señales.

Finalmente, la Figura 3.12 presenta de manera clara las conexiones realizadas internamente en el USRP además de sus componentes internos de forma resumida. Si bien es verdad que se refiere a un esquema genérico, el modelo B210 solamente cambiaría en la conexión Ethernet, realizándose mediante USB como se ha explicado previamente.

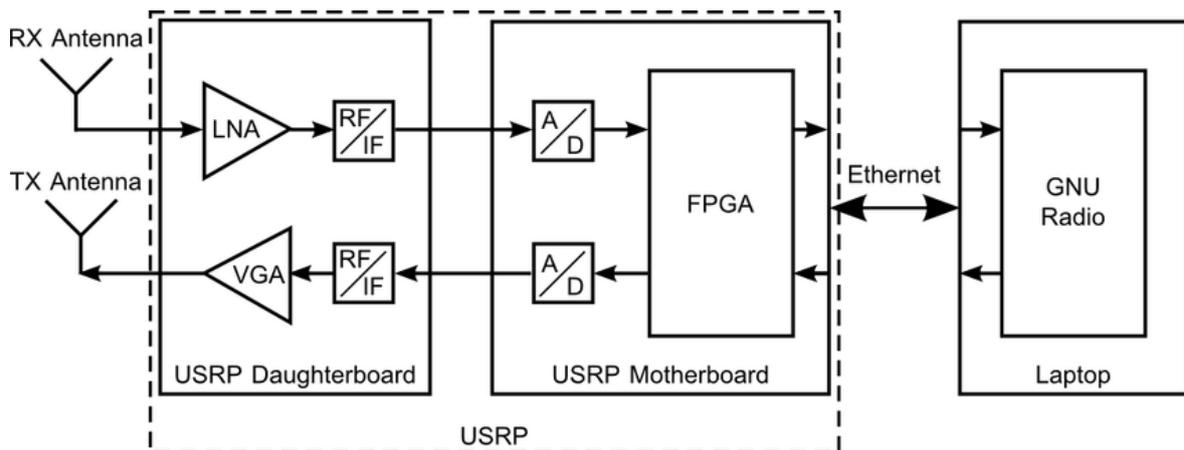


Figura 3.12.- Esquema de funcionamiento del USRP B210.

## 4. Medidas de la tasa de error de los diferentes escenarios hardware y software y comparativa con simulaciones

El propósito de este capítulo es determinar el diseño del sistema de transmisión y analizar el comportamiento de las modulaciones impropias bajo diferentes valores del factor de circularidad, tanto en simulaciones de software como en implementaciones reales utilizando hardware específico como los dispositivos USRP B210. Se examinan constelaciones QPSK y 16-QAM, comparando las probabilidades de error de símbolo provenientes de simulaciones de Matlab con resultados prácticos obtenidos mediante GNU Radio. Este enfoque permite evaluar cómo el coeficiente de circularidad afecta la robustez de la comunicación digital. Además, se discuten las limitaciones y la influencia del entorno físico en las implementaciones reales, proporcionando las ventajas y desventajas de las modulaciones impropias en diferentes contextos de aplicación. En cada sección se aportarán las gráficas necesarias que serán contrastadas con las recogidas durante la práctica de este TFG.

### 4.1.- Escenarios software

Para conocer el comportamiento de las modulaciones impropias frente a las tradicionales como QAM, en primera instancia se han generado distintas simulaciones en Matlab. Para cada valor de  $\kappa$ , se presenta una constelación de símbolos que define las posiciones posibles de los símbolos modulados. Luego, se transmiten  $10^6$  símbolos seleccionados aleatoriamente de esta constelación. Luego, se simula el proceso de decisión mediante un mapeado a los símbolos más cercanos en la constelación original tras añadir ruido gaussiano, cuyo nivel se ajusta de forma según la SNR correspondiente suponiendo que la potencia de la señal esta normalizada,  $\sigma^2 = 10^{\frac{\text{SNR}(\text{dB})}{10}}$ . La SER resultante se calcula comparando los símbolos recibidos con los símbolos transmitidos y determinando la proporción de errores, generando así las curvas de SER pertinentes en función de la SNR y para cada valor de  $\kappa$  analizado.

Posteriormente, se ha procedido a diseñar el sistema que se muestra en la Figura 3.7 en GNU Radio, en el cual se realiza una comunicación capaz de determinar los errores cometidos durante la transmisión de datos permitiendo analizar las diferencias entre ambos escenarios.

Con la finalidad de recoger los resultados adecuados en el sistema de GNU Radio, se ha definido inicialmente el factor de circularidad de interés, pudiendo obtener la constelación impropia determinada por dicho valor. En este caso, partiendo de una constelación QPSK, se utilizarán los diseños señalados en la Tabla 4.1 y representado en las Figuras 4.1-4.5.

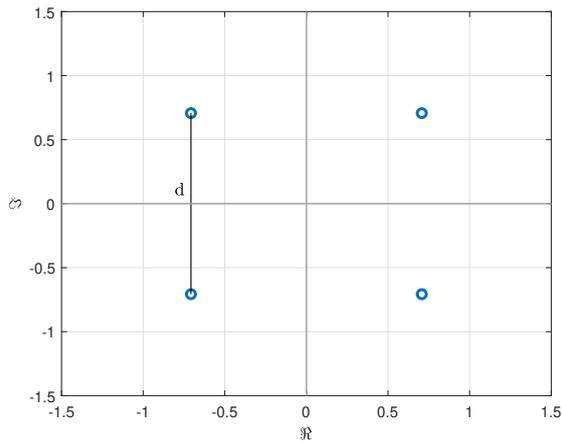


Figura 4.1.- Constelación QPSK estándar que es, a su vez, propia ( $\kappa = 0$ ).

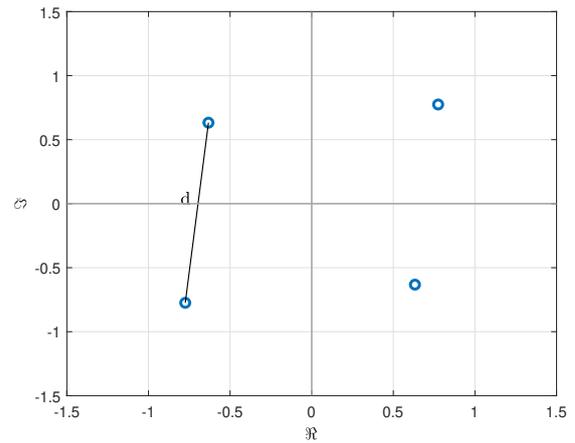


Figura 4.2.- Constelación QPSK impropia con  $\kappa = 0.2$ .

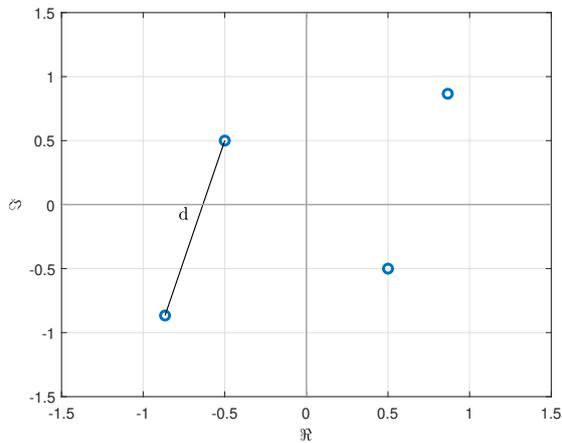


Figura 4.3.- Constelación QPSK impropia con  $\kappa = 0.5$ .

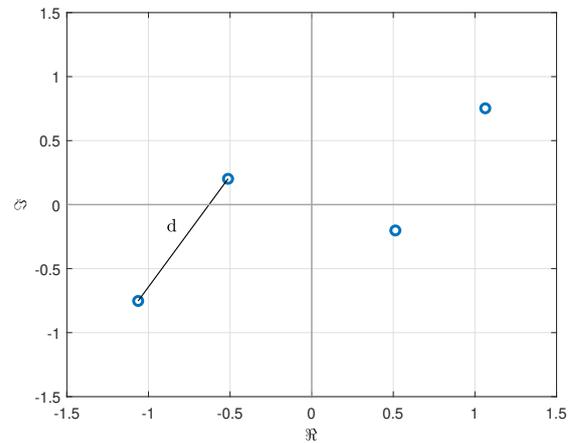


Figura 4.4.- Constelación QPSK impropia con  $\kappa = 0.8$ .

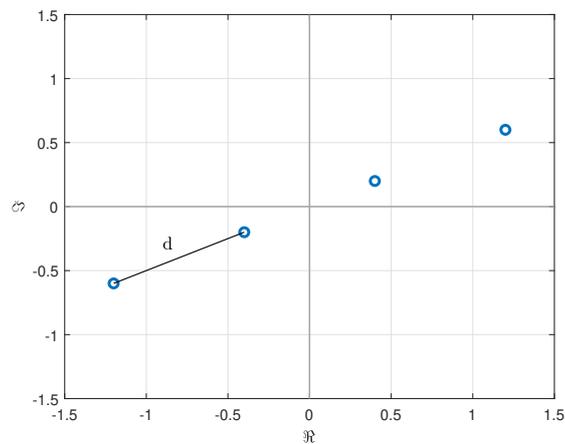


Figura 4.5.- Constelación QPSK totalmente impropia ( $\kappa = 1$ ) semejante a una 4-PAM rotada.

$\kappa$	$V$			
	00	01	10	11
0	$0.707 + 0.707j$	$0.707 - 0.707j$	$-0.707 + 0.707j$	$-0.707 - 0.707j$
0.2	$0.774 + 0.774j$	$0.632 - 0.632j$	$-0.632 + 0.632j$	$-0.774 - 0.774j$
0.5	$0.866 + 0.866j$	$0.500 - 0.500j$	$-0.500 + 0.500j$	$-0.866 - 0.866j$
0.8	$1.063 + 0.752j$	$0.512 - 0.201j$	$-0.512 + 0.201j$	$-1.063 - 0.752j$
1	$1.200 + 0.600j$	$0.400 - 0.200j$	$-0.400 + 0.200j$	$-1.200 - 0.600j$

Tabla 4.1.- Constelaciones impropias QPSK.

Es importante tener en cuenta que la elección de la secuencia de símbolos para GNU Radio no es libre. Eso se debe a la existencia de un bloque que compensa los desfases o rotaciones producidas por la actuación del canal. Si este elemento no asigna los símbolos de manera correcta, esta rotación no se puede compensar adecuadamente, haciendo que las tramas se consideren erróneas o, incluso, perdidas. Para poder realizar una asignación correcta, se han escogido como punto de partida las constelaciones predeterminadas de la librería *digital* de GNU Radio. Habiendo obtenido las constelaciones, se procederá a recoger la probabilidad de error de símbolo tanto de forma simulada con Matlab como en GNU Radio, a nivel de software, obteniendo así las curvas representadas en la Figura 4.6.

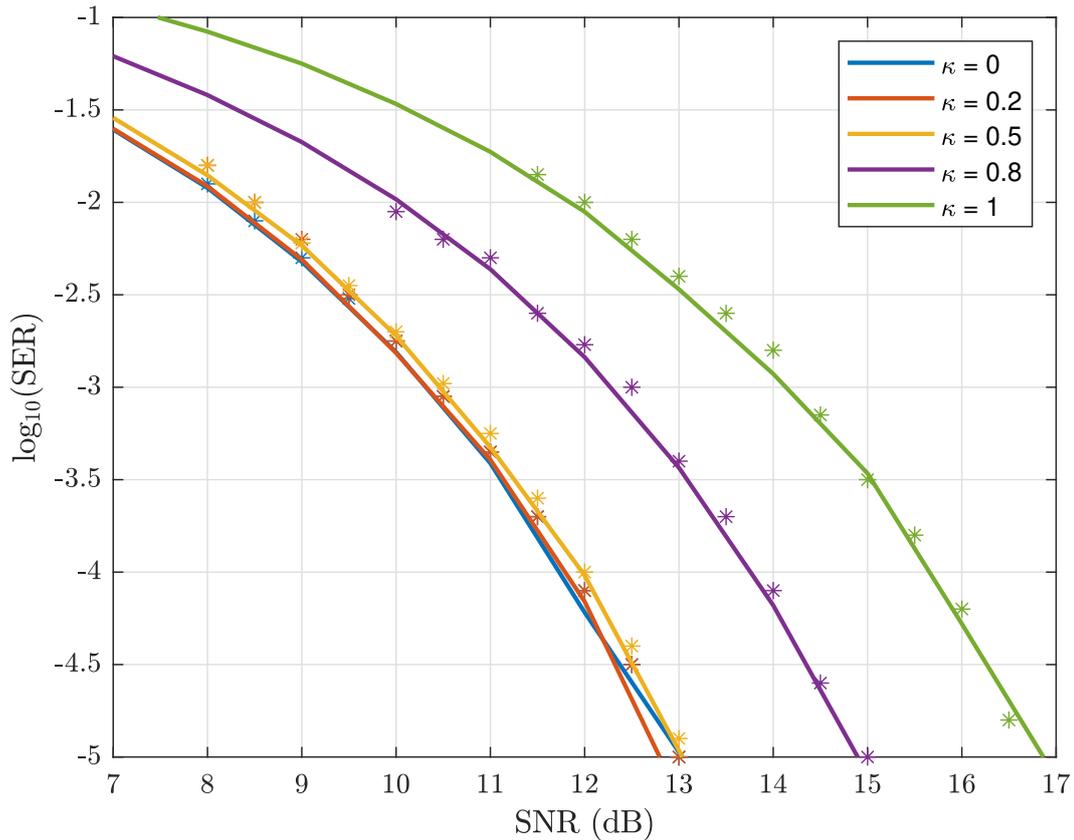


Figura 4.6.- Tasa de error de símbolo frente a relación señal-ruido de modulaciones QPSK impropias con distintos valores de  $\kappa$  en entorno software, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos.

Si bien cabía esperar que se reflejasen discrepancias con las simulaciones, la precisión de los resultados parece bastante buena, salvando pequeñas diferencias y que parece que los asteriscos sobreestiman ligeramente la  $P_E$ . Este fenómeno puede venir dado por distintos factores, entre ellos, la precisión de la herramienta GNU Radio, provocando pequeñas discrepancias entre las curvas. Adicionalmente, la influencia de los valores escogidos en el bloque CRCCheck, Apartado 3.2.5, es de gran relevancia debido al rango de validez que se podría considerar de los resultados. Con un número pequeño de PDUs, la precisión de la estimación de la SER puede ser limitada, ya que es una medida estadística que representa la proporción de símbolos recibidos incorrectamente respecto al total de símbolos transmitidos. Para obtener una estimación precisa, especialmente en condiciones de canal variable o ruido significativo, se requieren muestras estadísticas suficientes, conllevando que pocas PDUs puedan no ser representativas de la verdadera tasa de error. En este caso se ha escogido una cantidad de 2000 PDUs y que cada PDU transmitida está compuesta por 100 bytes, haciendo que el tiempo de procesamiento de cada transmisión no sea excesivo y no genere un error destacable.

Teniendo en cuenta estas consideraciones, los resultados presentes en GNU Radio son suficientemente parecidos a las simulaciones de Matlab, en general, como para decir que las diferencias apreciadas son muy ligeras y se pueden deber a la variabilidad numérica de las simulaciones.

Por otro lado, nos encontramos con una modulación QAM compuesta por 16 símbolos, cada uno compuesto por  $N = 4$  bits. Al igual que la constelación generada previamente, esta se formulará mediante codificación Gray, variando solamente un bit entre símbolos vecinos. En este caso, la Ecuación (2.35) solamente es válida para constelaciones impropias con  $\kappa \leq 0.8$ , ya que a partir de este valor se generaría un solapamiento entre los símbolos siendo imposible diferenciarlos. Las constelaciones impropias generadas a partir de una 16-QAM serán las recogidas en las Figuras 4.7-4.11.

Al utilizar un mayor número de bits en la modulación, se necesitará mayor potencia y, en su defecto, SNR para poder mantener la misma probabilidad de error de símbolo. De esta manera, las curvas teóricas presentan una tendencia similar a las calculadas para QPSK, pero desplazadas hacia la derecha [9].

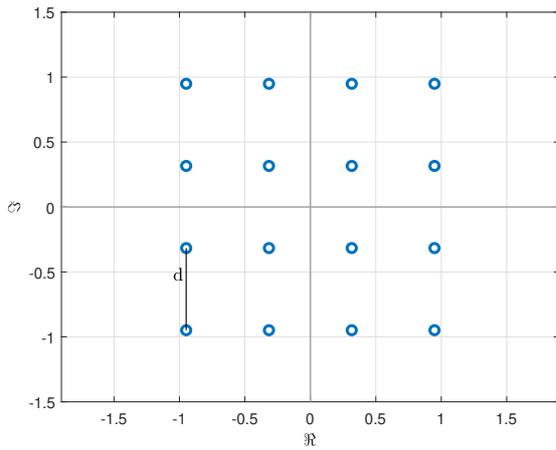


Figura 4.7.- Constelación 16-QAM estándar que es, a su vez, propia ( $\kappa = 0$ ).

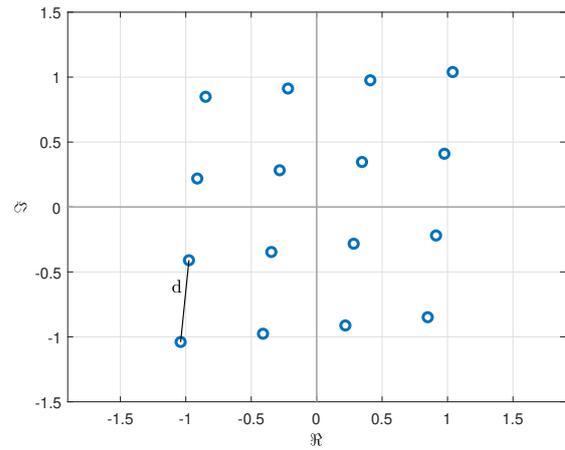


Figura 4.8.- Constelación 16-QAM impropia con  $\kappa = 0.2$ .

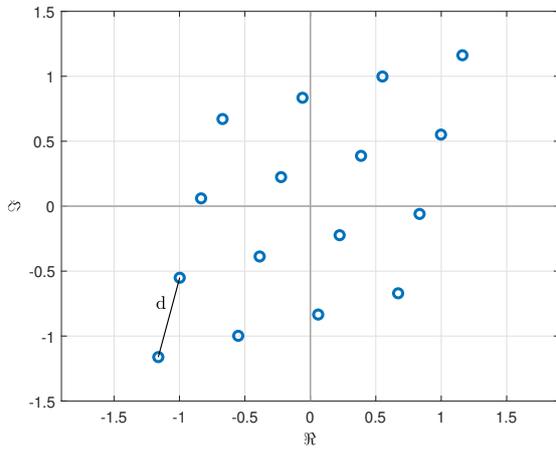


Figura 4.9.- Constelación 16-QAM impropia con  $\kappa = 0.5$ .

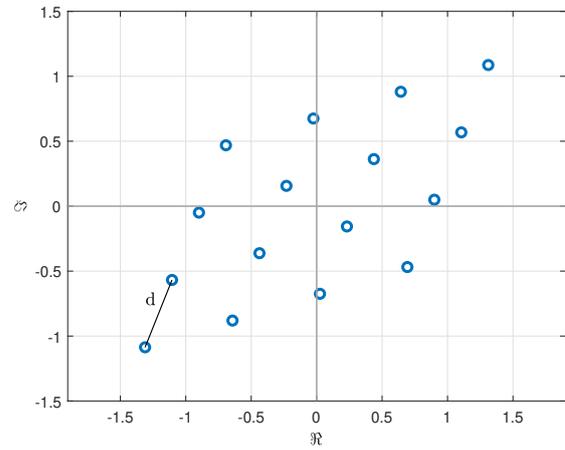


Figura 4.10.- Constelación 16-QAM impropia con  $\kappa = 0.65$ .

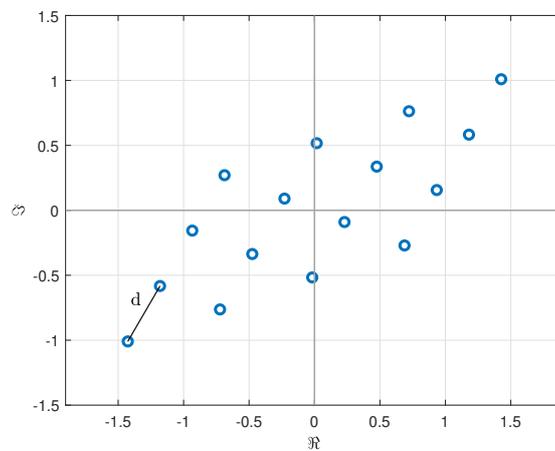


Figura 4.11.- Constelación 16-QAM impropia con  $\kappa = 0.8$ .

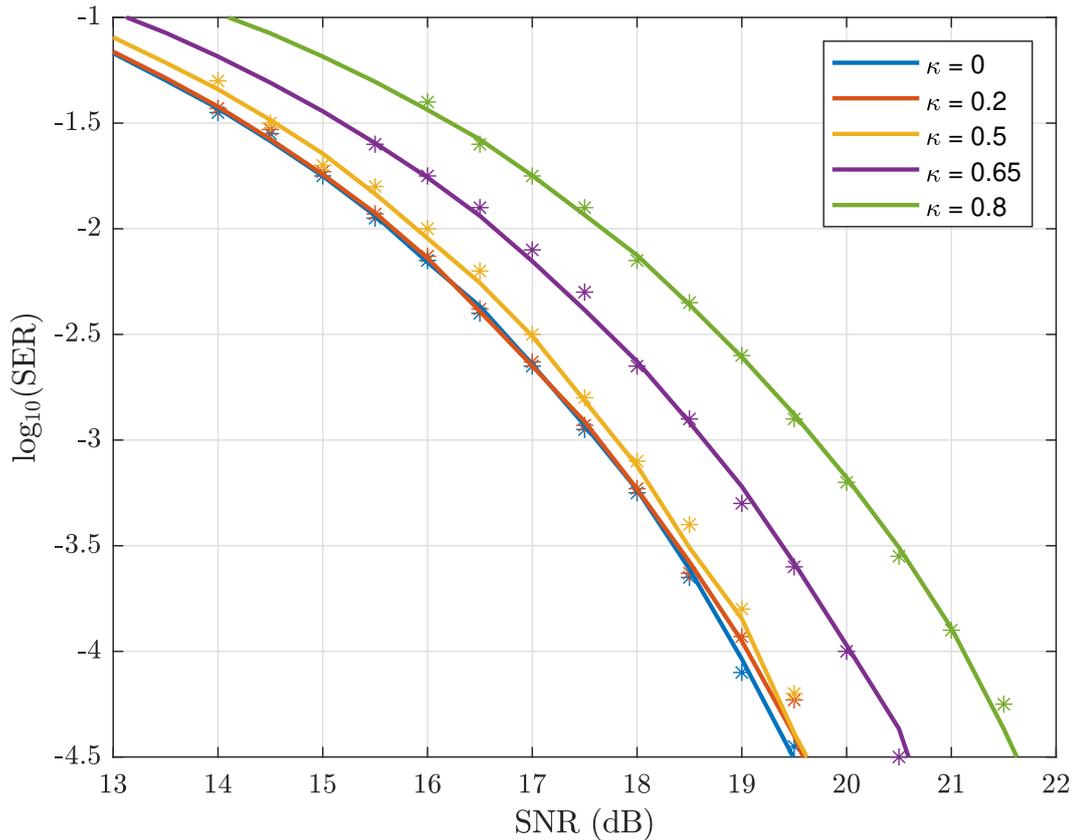


Figura 4.12.- Tasa de error de símbolo frente a relación señal-ruido de modulsiones 16-QAM impropias con distintos valores de  $\kappa$  en entorno software, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos.

Para constelaciones de mayor orden, como las que representan 4 bits/símbolo, la SNR requerida es generalmente más alta que para constelaciones de menor orden, como las de 2 bits/símbolo. Esto se debe a que, en constelaciones de mayor orden, los puntos de la constelación están más cerca unos de otros siempre que fijemos la energía media de la señal, lo que hace que el sistema sea más sensible al ruido y a las interferencias. Por lo tanto, es necesario una mayor SNR para mantener la misma probabilidad de error. En estos casos, es de interés aumentar el número de PDUs a analizar para garantizar una población estadística suficiente a partir de la que estimar la SER mediante los CRCs, pudiendo obtener una mejor estimación de las características del canal y resultados más estables. Esto ayudará a mejorar levemente la fiabilidad de los resultados de la comunicación.

Al igual que había sucedido en QPSK, los resultados se ajustan adecuadamente a los comportamientos teóricos esperados, siempre teniendo en cuenta los defectos inherentes al proceso de comunicación y los parámetros seleccionados para la simulación.

Adicionalmente, las modulsiones descritas por un factor de circularidad  $\kappa = 1$  definen constelaciones con símbolos equidistantes alineados en el plano, pero con una rotación, al igual que sus fronteras de decisión. Por ende, es conveniente hacer una comparación con

las modulaciones PAM, pues al estar dotadas por las mismas características la probabilidad de error de símbolo debería ser equivalente. A pesar de que con el algoritmo WLT, explicado en el Apartado 2.3.3, no se encuentra optimizada la modulación 16-QAM impropia con dicho  $\kappa$ , se puede confeccionar posicionando los símbolos en línea recta con una inclinación adecuada, es decir, equivalente a una 16-PAM rotada, representada en la Figura 4.13.

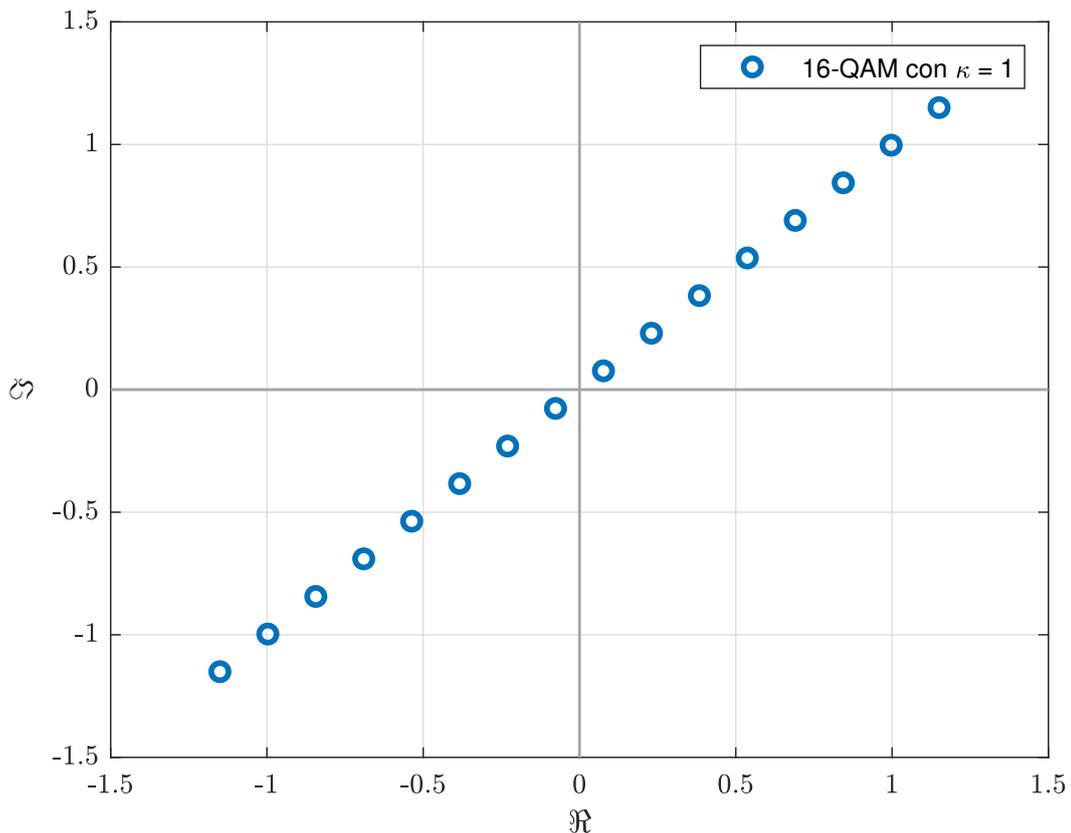


Figura 4.13.- Constelación 16-QAM totalmente impropia ( $\kappa = 1$ ) semejante a una 16-PAM rotada.

Los resultados de las constelaciones completamente impropias con el mayor factor de circularidad posible, Figura 4.14, definen perfectamente la tendencia de la probabilidad de error de las modulaciones PAM debido a la semejante alineación de los símbolos y regiones de decisión en ambas configuraciones.

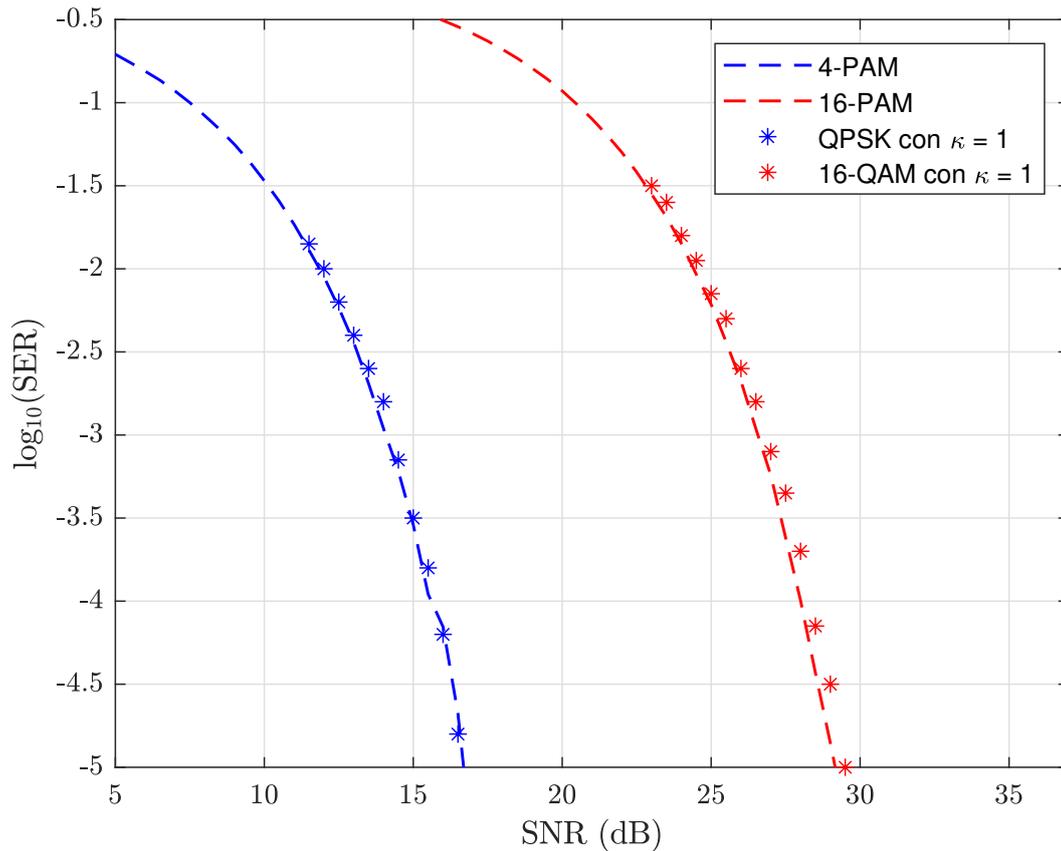


Figura 4.14.- Tasa de error de símbolo frente a relación señal-ruido de constelaciones impropias con  $\kappa = 1$  en el entorno software de en GNU Radio frente a simulaciones en Matlab de las modulaciones PAM.

## 4.2.- Escenarios hardware

Una vez se ha realizado el análisis de las simulaciones software y se ha corroborado su correcto comportamiento, se ejecutará el mismo sistema, pero en un escenario real, de forma que la comunicación sufrirá de perturbaciones que no se producían en las simulaciones y que se derivan de las imperfecciones de los componentes hardware. El proceso para recoger las muestras seguirá la misma línea, aunque se analizarán dos escenarios distintos. El primero de ellos constará de dos USRPB210 conectados mediante un cable de transmisión que actuará como canal. En el segundo caso, se procederá con una comunicación inalámbrica a una distancia de unas pocas decenas de centímetros. Estos procedimientos se detallarán a continuación.

#### 4.2.1.- Comunicación mediante medio guiado

Este escenario consistirá en dos USRP B210 conectados mediante un cable SMA, realizando un proceso de comunicación a 1 GHz, Figura 4.15. Para comunicarse con los dispositivos USRP se sustituirán al *Virtual Sink* y *Virtual Source* del sistema final (Figura 3.7) de la simulación software por dos bloques específicos, representados en la Figura 4.16. En ellos se deberá indicar la dirección del transmisor y receptor respectivamente, teniendo siempre en consideración la función de las interfaces de los dispositivos.



Figura 4.15.- Ilustración del montaje del escenario de comunicación mediante medio guiado.

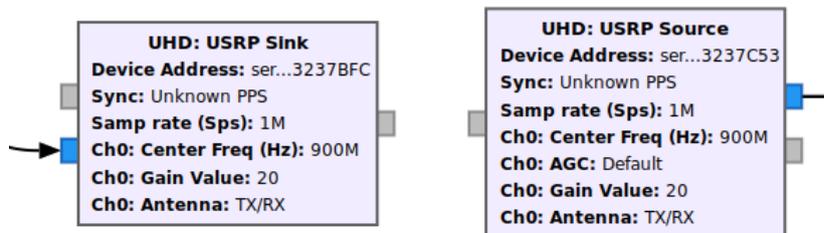


Figura 4.16.- Bloques de GNU Radio para comunicación con USRPs.

Habiendo configurado correctamente la dirección de los dispositivos se ha de considerar también el canal en cuestión. En este caso, nos encontramos ante un medio guiado de unos pocos decímetros, por lo que no generará grandes pérdidas en la comunicación. Aun así, es conveniente indicar una ganancia de 20 dB tanto en transmisión como en recepción para compensar la atenuación que se produzca en el canal.

En este punto, se tendría confeccionado el sistema final hardware, pudiendo recoger los resultados tal y como se realizó en la sección anterior.

Previo al análisis, hay que tener en consideración el ruido del canal y las condiciones variables de transmisión son factores adicionales que pueden afectar las mediciones. En un entorno real, la presencia de interferencias, desvanecimientos y otros fenómenos de propagación pueden introducir perturbaciones sobre la señal que no se reflejan completamente en las simulaciones, las cuales asumen condiciones ideales. Para tener cierto control sobre la SNR la señal transmitida se contamina con ruido cuya caracterización estadística es conocida, gaussiano de media 0 y varianza  $\sigma_x^2$ . La señal ruidosa transmitida llega al receptor tras sufrir perturbaciones en su camino, debidas al canal y a los elementos de la cadena receptora. Modelaremos estas perturbaciones como un ruido, también gaussiano, independiente del generado en transmisión de media 0 y varianza  $\sigma_c^2$ , que será desconocida. En condiciones de SNR baja, la señal de transmisión está mucho más contaminada por el ruido, lo que hace que los resultados medidos se desvíen significativamente de los valores simulados. Para entender por qué ocurre esto, consideremos la ecuación de SNR:

$$\text{SNR} = \frac{P}{\sigma_x^2 + \sigma_c^2}, \quad (4.1)$$

donde  $P$  representa la potencia de la señal transmitida, la cual se supondrá normalizada,  $\sigma_x^2$  es la potencia del ruido generada por el software (referida a la que existía en las secciones previas), y  $\sigma_c^2$  es la potencia del ruido del sistema real, incluyendo en este factor las interferencias de los componentes electrónicos de los dispositivos, el canal, etc.

En condiciones de alta SNR, la potencia del ruido del software  $\sigma_x^2$  es significativamente grande en comparación con  $\sigma_c^2$ , dominando en la ecuación. Esto implica que las imprecisiones y errores introducidos por la herramienta de simulación o el procesamiento digital afectan considerablemente la señal recibida. En este escenario, la señal está contaminada principalmente por errores numéricos y limitaciones inherentes al software utilizado al igual que sucedía en casos anteriores, provocando que los resultados no generen discrepancias significativas. Sin embargo, a medida que se reduce la SNR, el ruido del sistema real empieza a incrementarse progresivamente hasta llegar a ser significativamente mayor que el ruido del software, provocando que los resultados no reflejen una tendencia tan similar a las curvas teóricas. Estos comportamientos se muestran matemáticamente en la Ecuación (4.2), teniendo en cuenta que la situación más próxima se alcanzará cuando el ruido del sistema hardware sea mínimo,  $\sigma_x^2 \gg \sigma_c^2$ .

$$\begin{aligned} \text{SNR} &\stackrel{\sigma_x^2 \gg \sigma_c^2}{\approx} \frac{P}{\sigma_x^2} \\ \text{SNR} &\stackrel{\sigma_c^2 \gg \sigma_x^2}{\approx} \frac{P}{\sigma_c^2} \end{aligned} \quad (4.2)$$

Por ende, se observa como en las Figuras 4.17 y 4.18 las mediciones se acercan más a las curvas proporcionadas por las simulaciones cuanto mayor es el valor de SNR. A medida que se reduce, el ruido del canal protagoniza la peor tendencia, generando cada vez un mayor error cuanto mayor es el número de bits usado en la modulación y más alto es el valor de  $\kappa$ .

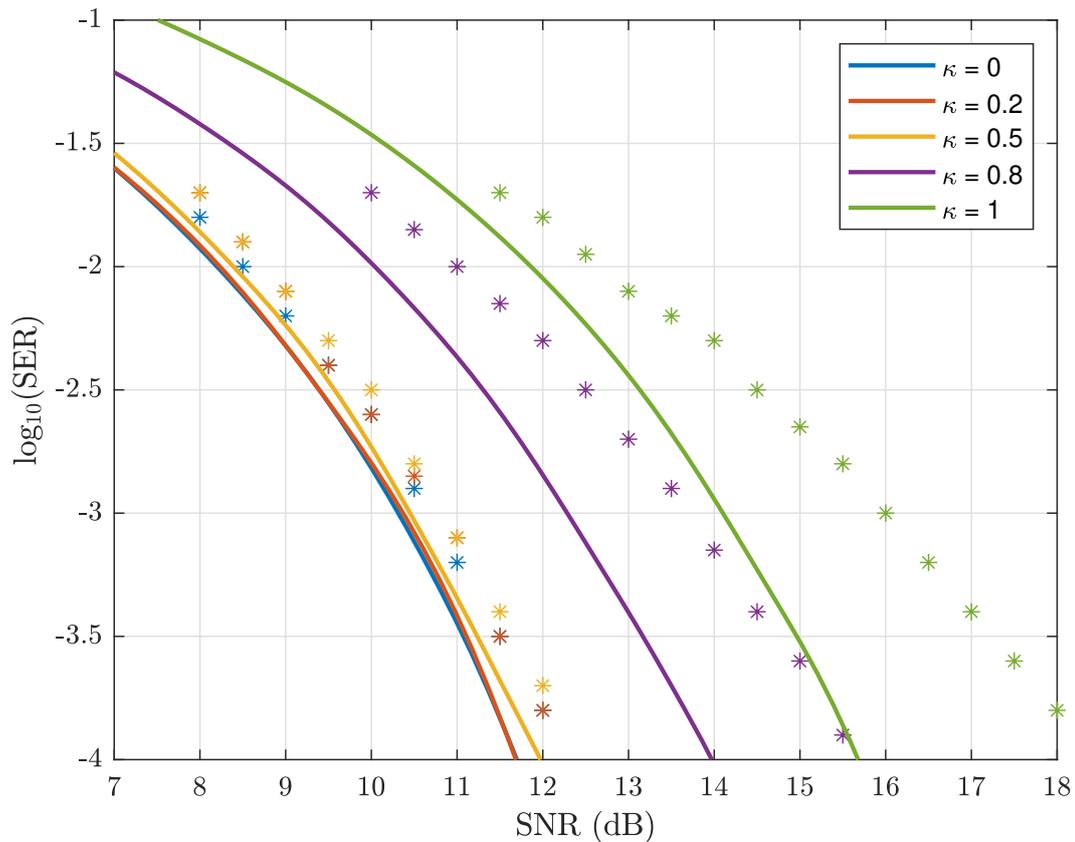


Figura 4.17.- Tasa de error de símbolo frente a relación señal-ruido de las modulaciones QPSK usando distintos valores de  $\kappa$  con USRPs conectados mediante cable SMA, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos.

Asimismo, en la Figura 4.18 no existe representación de la modulación completamente impropia, es decir, con  $\kappa = 1$ . Esto se debe principalmente a las limitaciones de los elementos hardware, los cuales no pueden garantizar la estabilidad de la comunicación.

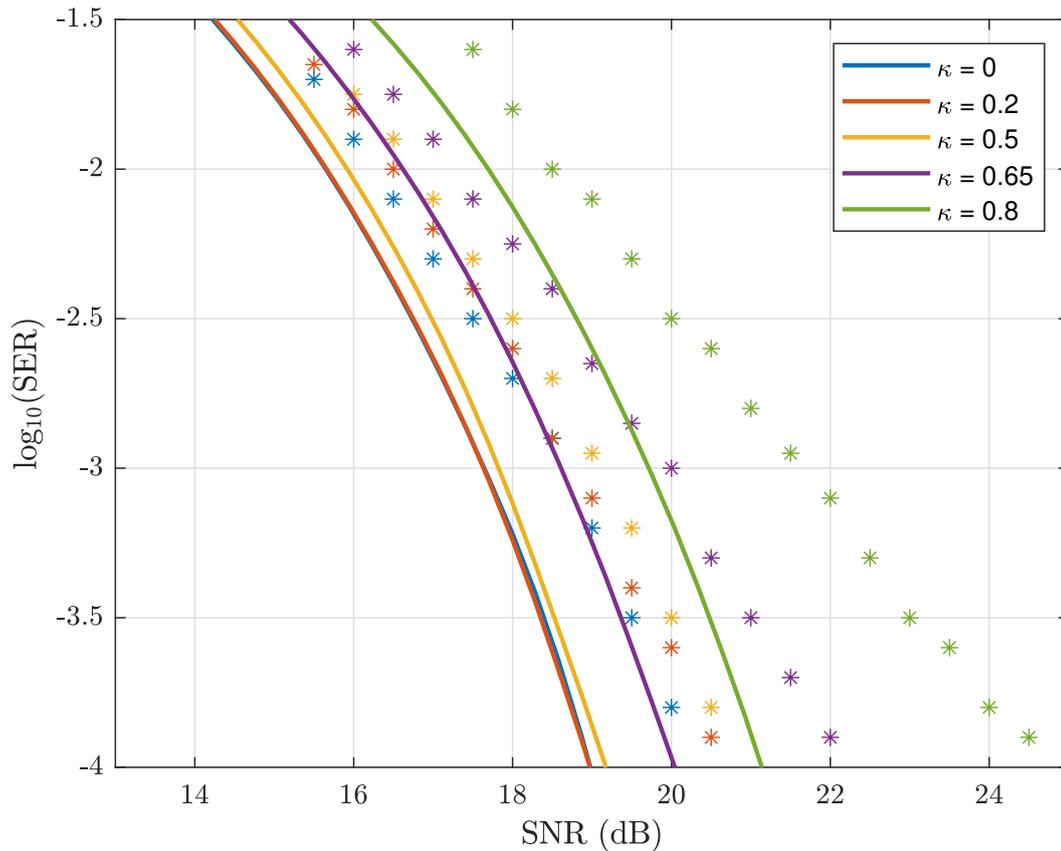


Figura 4.18.- Tasa de error de símbolo frente a relación señal-ruido de las modulaciones 16-QAM usando distintos valores de  $\kappa$  con USRPs conectados mediante cable SMA, representando las simulaciones de Matlab con líneas continuas y los resultados de GNU Radio con asteriscos.

#### 4.2.2.- Comunicación inalámbrica

En este segundo escenario, los USRP se comunicarán mediante dos antenas a una distancia de aproximadamente dos decímetros, teniendo como canal de transmisión el aire, Figura 4.19. La configuración del esquema GNU Radio será parecida a la realizada por el medio guiado, distinguiéndose en términos de ganancia. En este caso, el canal inalámbrico produce mayores pérdidas que un medio guiado, por lo que habrá que compensarlas aumentando la ganancia del transmisor a aproximadamente 50 dB. De esta forma, se mantendrá un canal con un comportamiento estable pudiendo obtener los resultados adecuados. De igual forma que en el medio guiado, la frecuencia de transmisión será de 1 GHz.

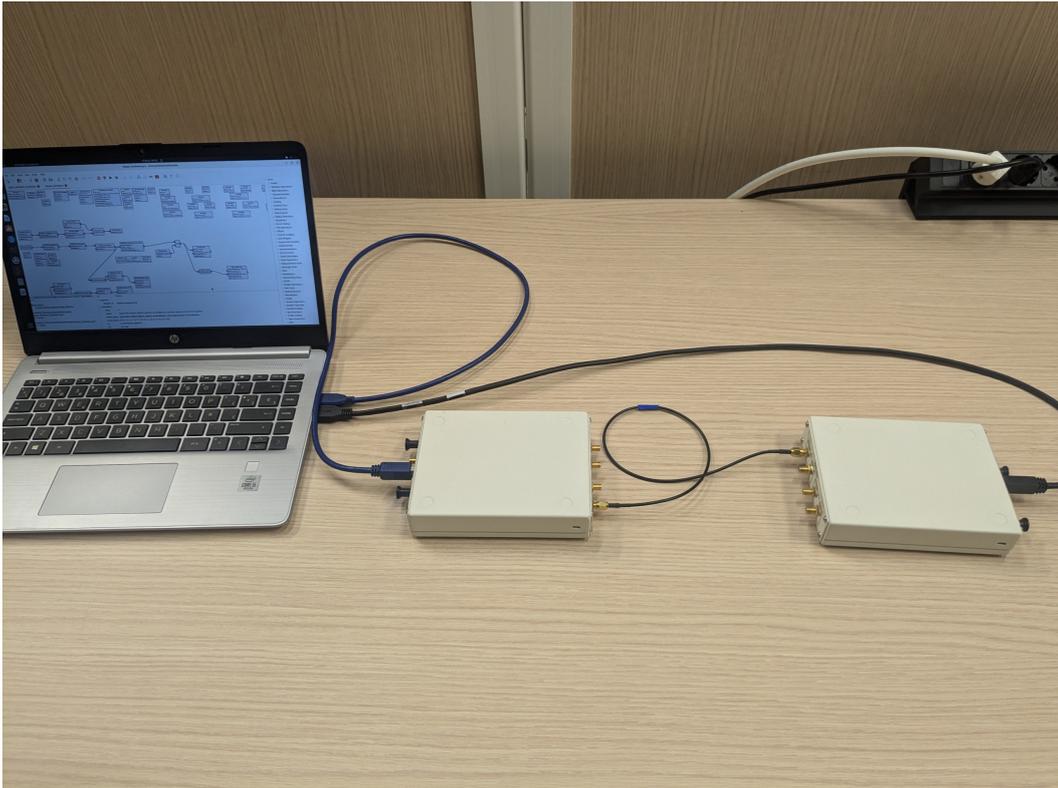


Figura 4.19.- Ilustración del montaje del escenario de comunicación inalámbrica.

En la Figura 4.17, se presenta la probabilidad de error de símbolo para la modulación QPSK. A diferencia de los escenarios de simulación y medio guiado, donde las condiciones del canal eran más controladas y el ruido se modelaba de manera ideal, el entorno inalámbrico introduce variabilidades adicionales debido a la interferencia, el desvanecimiento y la dispersión de la señal. Esto se traduce en una mayor probabilidad de error para los mismos valores de  $\kappa$  y SNR comparados con los resultados obtenidos en el medio guiado. En particular, en el medio guiado se observó que un incremento en  $\kappa$  mejoraba notablemente la SER, pero en el entorno inalámbrico, aunque la tendencia se mantiene, la mejora es menos pronunciada debido a las adversidades del canal real.

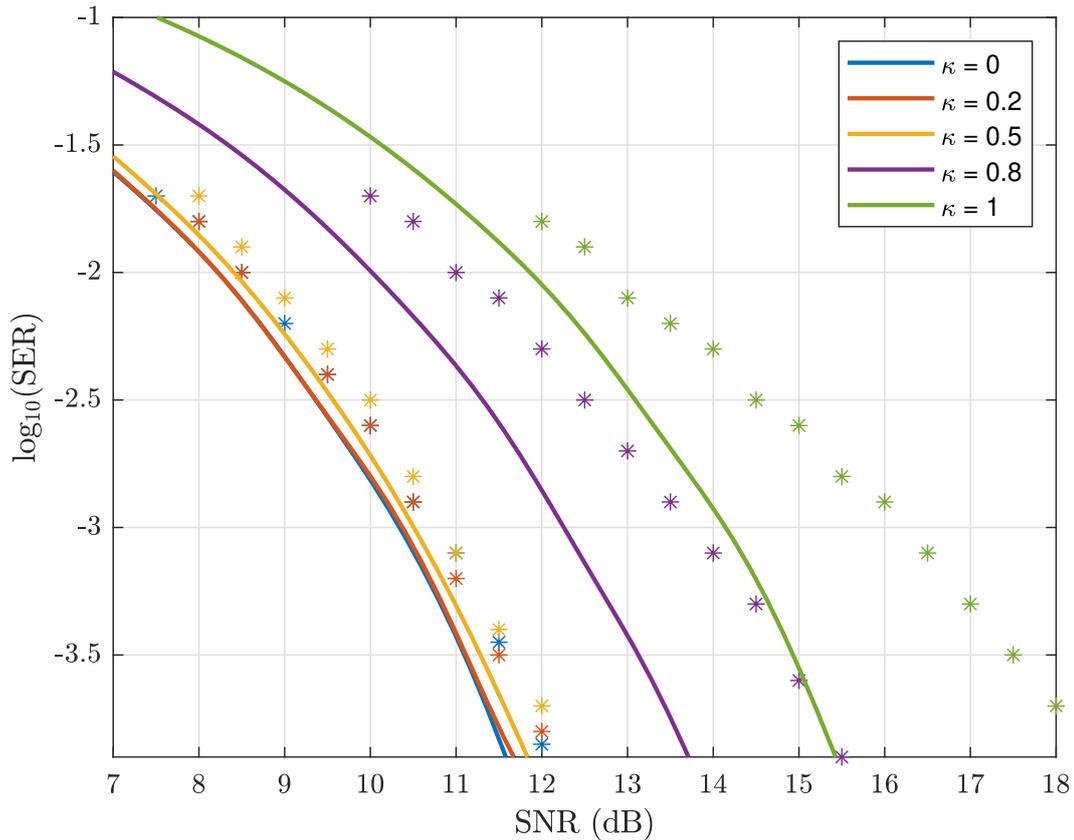


Figura 4.20.- Tasa de error de símbolo frente a relación señal-ruido de las modulaciones QPSK usando distintos valores de  $\kappa$  con USRPs conectados de forma inalámbrica.

Por otro lado, la Figura 4.18 muestra la tasa de error de símbolo para la modulación 16-QAM. Nuevamente, se observa que la SER es mayor en el entorno inalámbrico comparado con las simulaciones y el medio guiado. Esto es especialmente evidente en las modulaciones 16-QAM, donde la densidad de símbolos es mayor y, por ende, la sensibilidad al ruido e interferencia es también mayor. En el escenario de medio guiado, la separación entre símbolos permitía una discriminación más efectiva en el receptor, reduciendo la SER de manera más eficiente que en el entorno inalámbrico, donde la dispersión de la señal y el desvanecimiento juegan un papel más crítico [10].

Además, la implementación práctica con USRPs en un entorno inalámbrico demuestra las limitaciones de las condiciones ideales asumidas en las simulaciones. Las variaciones de la señal debidas a movimientos físicos, obstrucciones y otros dispositivos inalámbricos no se pueden modelar completamente en un entorno simulado o de medio guiado, lo que lleva a una discrepancia entre los resultados ideales y los observados en condiciones reales.

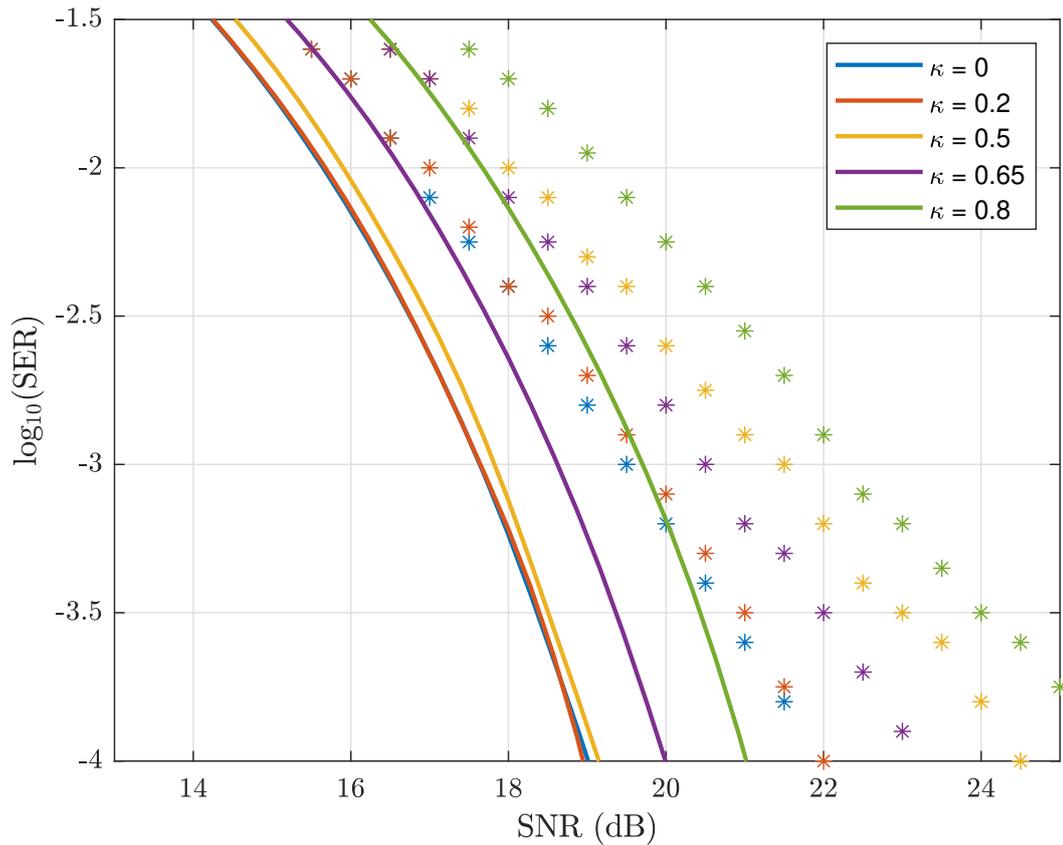


Figura 4.21.- Tasa de error de símbolo frente a relación señal-ruido de las modulaciones 16-QAM usando distintos valores de  $\kappa$  con USRPs conectados de forma inalámbrica.

Finalmente, es de interés poder observar una representación de todos los escenarios, tanto simulados en software como en hardware, para una modulación y un valor de kappa. En este caso se ha escogido una QPSK con  $\kappa = 1$ . Los fenómenos descritos durante todo el capítulo como ruido del canal, interferencias de los dispositivos electrónicos, etc, reflejan las diferencias notables en las curvas ilustradas por los escenarios reales. Habiendo investigado previamente el comportamiento de las modulaciones en un entorno más controlado como es un medio guiado, mediante la Figura 4.22 se corrobora que la influencia del canal en este escenario será mayor, provocando una pérdida de la tendencia asintótica a medida que aumenta la SNR.

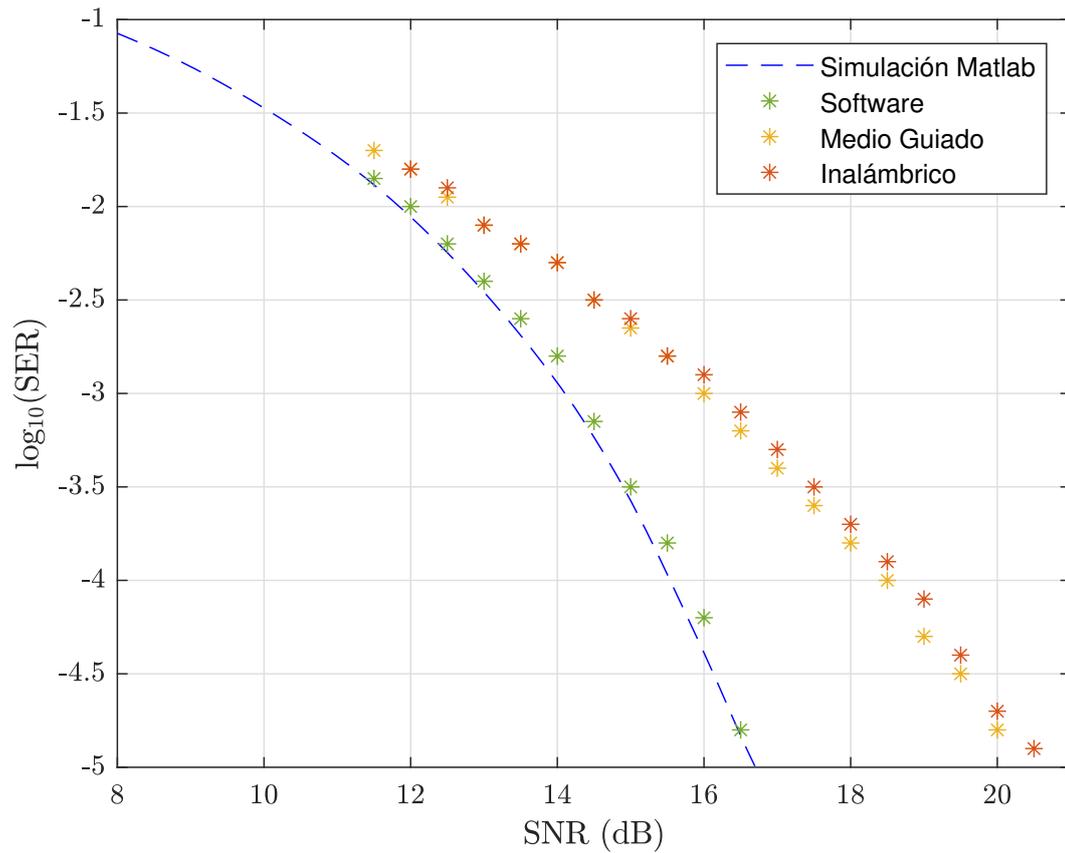


Figura 4.22.- Tasa de error de símbolo frente a relación señal-ruido de constelaciones QPSK impropias con  $\kappa = 1$  en todos los escenarios descritos.

## 5. Conclusiones y líneas futuras

El presente Trabajo de Fin de Grado ha explorado la implementación y evaluación de modulaciones impropias en el contexto de las comunicaciones digitales, utilizando la plataforma de SDR y GNU Radio. A lo largo del desarrollo de este estudio, se ha realizado un modelado teórico de las constelaciones impropias para sistemas de comunicaciones digitales para luego llevar a cabo una implementación utilizando GNU Radio en un entorno de desarrollo software que permitió la simulación y ajuste de las señales en un contexto controlado. Además, se ha utilizado hardware específico, como los dispositivos USRP, para analizar las señales en entornos reales con el objetivo de obtener los resultados de SER bajo diversas condiciones de operación.

En primer lugar, el análisis teórico y las simulaciones prácticas han confirmado la viabilidad de las modulaciones impropias. Estas modulaciones, al transformar los valores de las constelaciones tradicionales, permiten una mayor flexibilidad en la codificación de señales. A pesar de que no se ha comprobado durante el este estudio, esta característica puede resultar ventajosa en un sistema en el que la interferencia tenga un papel relevante.

Por otra parte, es importante destacar que las pruebas prácticas realizadas con dispositivos USRP B210 han revelado ciertas discrepancias entre los resultados ideales y los observados en condiciones reales. Estas diferencias se deben principalmente a factores como el ruido del canal y las interferencias de otros dispositivos electrónicos, cuyo modelado puede llegar a ser realmente complejo en un entorno de simulación.

La integración de las modulaciones impropias en entornos SDR utilizando GNU Radio ha sido uno de los aspectos más destacables de este trabajo. Las pruebas realizadas en condiciones controladas, como era el escenario software, han mostrado que estas modulaciones pueden mantener niveles de rendimiento similares a las modulaciones tradicionales en términos de SER. No obstante, es necesario considerar que las modulaciones impropias requieren una menor cantidad de ruido para mantener el mismo nivel de rendimiento en condiciones reales.

Finalmente, las líneas de trabajo futuras más destacadas de cara a la ampliación de este TFG se recogen a continuación:

- Implementar escenarios con interferencias en los que varios agentes transmitan simultáneamente y compartan la misma banda de frecuencias, sin utilizar división por código. Estos escenarios deberían variar en características tales como la distancia entre los usuarios y el uso de diferentes elementos de transmisión, entre otros. Esta diversidad refleja condiciones más realistas, permitiendo investigar cómo las modulaciones impropias se comportan bajo tales circunstancias.
- Expandir la recopilación de datos de las modulaciones impropias mediante la realiza-

ción de medidas dinámicas en una variedad más amplia de puntos en movimiento. Esto involucra configurar escenarios donde los dispositivos de comunicación se desplacen a lo largo de diferentes trayectorias y velocidades para simular escenarios de movilidad realista, tales como vehículos en tránsito o dispositivos portátiles en uso humano. Al analizar cómo las constelaciones impropias se comportan bajo estas condiciones dinámicas, se puede evaluar más efectivamente sus características en entornos realistas y cambiantes.

- Investigar cómo diferentes valores de  $\kappa$  afectan la robustez y eficiencia de la señal en diferentes escenarios de comunicación limitados por interferencia, pudiendo determinar las características y parámetros óptimos de modulaciones impropias para diferentes aplicaciones.
- Explorar modulaciones impropias que no se abordaron inicialmente e investigar cómo estas modulaciones se comportan bajo diferentes condiciones de canal propuestas indicando los beneficios prácticos puede ser un área fructífera de estudio. Esto también podría incluir el desarrollo de nuevas constelaciones de señales que mejoren la tolerancia al ruido.

# Referencias

- [1] Ignacio Santamaria, Pedro M. Crespo, Christian Lameiro, and Peter J. Schreier. Information-theoretic analysis of a family of improper discrete constellations. *Entropy*, 20(1), 2018. ISSN 1099-4300. doi: 10.3390/e20010045.
- [2] Jesús Alberto López-Fernández, Rafael González Ayestarán, Ignacio Santamaria, and Christian Lameiro. Design of asymptotically optimal improper constellations with hexagonal packing. *IEEE Transactions on Communications*, 67(8):5445–5457, 2019. doi: 10.1109/TCOMM.2019.2916857.
- [3] Álvaro Pendás-Recondo, Jesús Alberto López-Fernández, and Rafael González Ayestarán. Study of siso superposition coding performance from improper discrete signaling perspective. *IEEE Wireless Communications Letters*, 13(3):884–888, 2024. doi: 10.1109/LWC.2023.3349170.
- [4] MathWorks. *MATLAB User's Guide*. URL <https://www.mathworks.com/products/matlab.html>.
- [5] GNU Radio. *GNU Radio: The Free & Open-Source Radio Ecosystem*. URL <https://www.gnuradio.org/>.
- [6] GNU Radio Tutorials. *Implementing Improper Modulations with SDR*. URL <https://tutorials.gnuradio.org/implementing-improper-modulations/>.
- [7] Ettus Research. *Universal Software Radio Peripheral (USRP)*, . URL <https://www.ettus.com/>.
- [8] Ettus Research. *USRP B210*, . URL <https://www.ettus.com/all-products/usrp-b210/>.
- [9] Simarjeet Kaur, Navdeep Singh, Gagandeep Kaur, and Jasbir Singh. Performance comparison of bpsk, qpsk and 16-qam modulation schemes in ofdm system using reed-solomon codes. In *2018 International Conference on Recent Innovations in Electrical, Electronics Communication Engineering (ICRIEECE)*, pages 530–533, 2018. doi: 10.1109/ICRIEECE44171.2018.9008983.
- [10] Jihua Huang, Shikai Shen, and Xingyuan Chen. Modeling and studying of the 16-qam based system over the fading channel. pages 1458–1460, 12 2017. doi: 10.1109/ICCTEC.2017.00316.