

UNIVERSIDAD DE VALLADOLID



E.T.S.I.T TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE
TELECOMUNICACIÓN

Radiocomunicaciones con radio definida por software y GNU Radio

Autor: Marta Tábara Carbajo

Tutor: Ramón de la Rosa

Septiembre 2023

TÍTULO: Radiocomunicaciones con radio definida por software y GNU Radio
AUTOR: Marta Tábara Carbajo
TUTOR: Ramón de la Rosa Steinz
DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería Telemática

COMISIÓN EVALUADORA

PRESIDENTE: Ramón de la Rosa Steinz
SECRETARIO: Alonso Alonso Alonso
VOCAL: Juan Carlos Aguado Manzano
SUPLENTE 1: Alberto Izquierdo Fuente
SUPLENTE 2: Javier M. Aguiar Pérez

FECHA: 27 de septiembre de 2023

Resumen

Conocido el origen y las bases de la radio definida por software, se explicará brevemente la fundamentación teórica de las modulaciones analógicas y digitales para proceder a realizar numerosos experimentos de simulación haciendo uso del software GNU Radio. En un primer lugar dichas simulaciones tendrán lugar dentro de un mismo equipo para posteriormente poner en práctica su transmisión y recepción entre diferentes equipos utilizando el transmisor HackRF y un receptor RTL-SDR.

Abstract

Once the origin and basis of software-defined radio are introduced, the theoretical foundation of analogical and digital modulations will be addressed below in order to carry out further simulation experiments through GNU Radio software. At a first stage, said simulations will take place within the same unit so as to subsequently implement its transmission and reception between different units using HackRF transmitter and RTL-SDR receiver.

Palabras clave

Modulación, demodulación, transmisor, receptor, SDR, radiocomunicaciones, HackRF, RTL-SDR, analógico, digital, AM, FM, ASK, FSK, PSK, QAM

Agradecimientos

En primer lugar me gustaría dar las gracias al tutor de este proyecto, Ramón de la Rosa, por darme a conocer el mundo de la Radio Definida por Software y guiarme en el desarrollo de este proyecto.

Agradecer a todos los profesores que me he cruzado en este camino por impartirme conocimientos valiosos a lo largo de mi carrera académica.

En tercer lugar agradecer a todas las personas que me han hecho seguir hacia delante en los momentos más duros y que me han enseñado a disfrutar del camino.

Dar las gracias también a mi familia, por su apoyo en todo momento y su paciencia a lo largo de estos años de carrera.

Y por último, a mi abuelo, por no dejar de creer en mi.

Índice general

1. Introducción	13
1.1. Breve historia de las comunicaciones	13
1.2. Introducción a la radio definida por software	15
1.3. Motivación	16
1.4. Aplicaciones de la Radio Definida por Software	17
1.5. Estructura de la memoria	19
2. Estado del arte	21
2.1. Introducción	21
2.2. Historia de la Radio Definida por Software	21
2.3. Origen y motivación	22
2.4. Equipos de radio definida por software	23
2.4.1. Funcionamiento de un receptor SDR	24
2.4.2. Dispositivos	25
2.5. Diagramas de bloques de los dispositivos utilizados	33
2.6. Software SDR	35
3. Tecnología utilizada	41
3.1. Software	41
3.1.1. Instalación de GNU Radio Companion 3.10.1.1 en Linux	43
3.2. Hardware	43
3.2.1. HackRF One	43
3.2.2. RTL-SDR	45
4. Modulaciones analógicas	47
4.1. Introducción	47
4.2. Modulaciones analógicas	47
4.3. AM: Modulación en amplitud	48

4.3.1.	Fundamentos de la modulación en amplitud	48
4.3.2.	Simulación de la modulación en amplitud en GNU Radio	50
4.3.3.	Fundamentos de la demodulación en amplitud	53
4.3.4.	Simulación de la demodulación en amplitud en GNU Radio	54
4.4.	FM: Modulación en frecuencia	57
4.4.1.	Fundamentos de la modulación en frecuencia	57
4.4.2.	Simulación de la modulación en frecuencia en GNU Radio	61
4.4.3.	Fundamentos de la demodulación en frecuencia	62
4.4.4.	Simulación de la demodulación en frecuencia en GNU Radio	63
5.	Modulaciones digitales	65
5.1.	Introducción	65
5.2.	Modulación digital de amplitud: ASK	69
5.2.1.	Fundamentos de la modulación digital de amplitud . . .	69
5.2.2.	Simulación de la modulación digital de amplitud en GNU Radio	71
5.2.3.	Fundamentos de la demodulación digital de amplitud . .	72
5.2.4.	Simulación de la demodulación digital de amplitud en GNU Radio	75
5.3.	Modulación digital de frecuencia: FSK	76
5.3.1.	Fundamentos de la modulación digital de frecuencia . . .	76
5.3.2.	Simulación de la modulación digital de frecuencia en GNU Radio	79
5.3.3.	Fundamentos de la demodulación digital de frecuencia	82
5.3.4.	Simulación de la demodulación digital de frecuencia en GNU Radio	83
5.4.	Modulación digital de fase: PSK	85
5.4.1.	Fundamentos de la modulación digital de fase	85
5.4.2.	Simulación de la modulación digital de fase en GNU Radio	87
5.4.3.	Fundamentos de la demodulación digital de fase	90
5.4.4.	Simulación de la demodulación digital de fase en GNU Radio	92
5.5.	Modulación de amplitud en cuadratura: QAM	93
5.5.1.	Fundamentos de la modulación de amplitud en cuadra- tura	93

5.5.2.	Simulación de la modulación de amplitud en cuadratura en GNU Radio	96
5.5.3.	Fundamentos de la demodulación de amplitud en cuadratura	97
5.5.4.	Simulación de la demodulación de amplitud en cuadratura en GNU Radio	100
6.	Resultados de la transmisión y recepción analógica	103
6.1.	Introducción	103
6.2.	Modulación en amplitud	103
6.2.1.	Resultados de la transmisión AM en GNU Radio . . .	103
6.2.2.	Resultados de la recepción AM en GNU Radio	108
6.3.	Modulación en frecuencia	116
6.3.1.	Resultados de la transmisión FM en GNU Radio	116
6.3.2.	Resultados de la recepción FM en GNU Radio	120
7.	Resultados de la transmisión y recepción digital	125
7.1.	Introducción	125
7.2.	Modulación digital en amplitud	125
7.2.1.	Resultados de la transmisión ASK en GNU Radio . . .	125
7.2.2.	Resultados de la recepción ASK en GNU Radio	127
7.3.	Modulación digital en fase	132
7.3.1.	Resultados de la transmisión PSK en GNU Radio . . .	132
7.3.2.	Resultados de la recepción PSK en GNU Radio	134
8.	Conclusiones y líneas futuras	147
8.1.	Conclusiones	147
8.2.	Líneas futuras	148

Capítulo 1

Introducción

1.1. Breve historia de las comunicaciones

A lo largo de la historia de la humanidad, las formas de comunicación han sido fundamentales para el desarrollo y la evolución de las sociedades. Desde los gestos y sonidos primitivos de nuestros ancestros hasta la conectividad global de la era digital, el proceso de transmitir información, conocimiento y emociones ha sido una fuerza impulsora en la construcción de civilizaciones y la conexión entre culturas.

En sus inicios, las primeras formas de comunicación se basaban en gestos, gruñidos y señales corporales, permitiendo a los primeros seres humanos compartir información sobre peligros, alimentos y otras necesidades básicas. Sin embargo, con el tiempo, el lenguaje oral se desarrolló y se convirtió en un medio más complejo y eficiente para la comunicación, posibilitando la transmisión de ideas, historias y tradiciones culturales.

El siguiente gran avance en la historia de las comunicaciones fue la invención de la escritura. Los primeros sistemas de escritura, como los jeroglíficos egipcios y la escritura cuneiforme de Mesopotamia, surgieron hace miles de años y permitieron la preservación y transmisión de información en tablillas de arcilla y papiro. Este hito marcó el comienzo de la capacidad de almacenar conocimiento de manera más permanente.

Con el crecimiento y la expansión de las civilizaciones, surgió la necesidad

de comunicarse a larga distancia. Así, los sistemas de mensajería a caballo, como el servicio de correos de los antiguos persas, permitieron la transmisión de mensajes entre regiones distantes, agilizando los intercambios comerciales y culturales.

El siglo XIX fue testigo de dos invenciones que revolucionarían las comunicaciones: el telégrafo y el teléfono. El telégrafo, desarrollado por Samuel Morse, permitía el envío de mensajes a través de líneas telegráficas mediante un código de puntos y rayas, llevando la comunicación a una nueva velocidad. Posteriormente, el teléfono, inventado por Alexander Graham Bell, permitió la comunicación de voz en tiempo real a través de cables, facilitando la conexión entre personas y empresas.

El siglo XX trajo consigo nuevos hitos en las comunicaciones con la invención de la radio y la televisión. La transmisión inalámbrica de señales de radio revolucionó la forma en que las noticias, la música y el entretenimiento eran difundidos a nivel global. La televisión, por su parte, combinó imágenes en movimiento con audio para transmitir información y entretenimiento en tiempo real, transformando la manera en que las personas recibían y compartían contenido.

Sin embargo, el mayor salto en la historia de las comunicaciones ocurrió con la creación de Internet en la década de 1960. ARPANET sentó las bases para lo que se convertiría en la red de redes, revolucionando la forma en que las personas se comunican y acceden a la información. La llegada de Internet ha dado lugar al correo electrónico, la World Wide Web y las redes sociales, transformando por completo la comunicación y la sociedad en la era digital. [1]

En la actualidad, las comunicaciones continúan evolucionando a pasos agigantados con tecnologías como la inteligencia artificial, la realidad virtual y el internet de las cosas (IoT). Los teléfonos móviles se han convertido en smartphones que ofrecen una amplia gama de servicios de comunicación y acceso a la información desde cualquier lugar y en cualquier momento.

Este Trabajo de Fin de Grado tiene como objetivo explorar y analizar la radio definida por software (SDR), una tecnología que se enmarca en el continuo desarrollo de las comunicaciones. A lo largo de estas páginas se examinarán

los fundamentos de SDR, su relevancia en el ámbito de las comunicaciones modernas, las posibles aplicaciones que ofrece y se explorarán los desafíos y limitaciones que presenta.

En resumen, esta investigación busca contribuir al conocimiento y comprensión de la radio definida por software, una innovación que sigue transformando la forma en que nos comunicamos y nos conectamos en la sociedad actual. A través del análisis de su historia y estado actual, esperamos proporcionar una perspectiva completa y actualizada sobre esta emocionante y prometedora área de estudio.

1.2. Introducción a la radio definida por software

En el mundo de las comunicaciones, la tecnología avanza a un ritmo vertiginoso, brindando soluciones cada vez más eficientes y versátiles. Uno de los avances significativos en este campo es la Radio Definida por Software (SDR), una innovadora aproximación que ha revolucionado la forma en que concebimos los sistemas de radio.

La Radio Definida por Software es una tecnología que permite la implementación de funciones de radio mediante software, en lugar de utilizar componentes electrónicos específicos para cada tarea. En esencia, SDR reemplaza gran parte del hardware tradicional de radiofrecuencia por software que realiza las mismas funciones de forma flexible y reconfigurable. Esta característica única proporciona una serie de ventajas en comparación con los sistemas de radio convencionales.

A lo largo de estas páginas nos adentraremos en el fascinante mundo de la Radio Definida por Software. Exploraremos sus fundamentos teóricos, su evolución histórica y sus diversas aplicaciones en el ámbito de las comunicaciones. Además, analizaremos las ventajas que ofrece SDR, como la adaptabilidad a diferentes estándares de comunicación, la reducción de costos y la posibilidad de actualizaciones a través de software.

También se examinarán los desafíos que enfrenta esta tecnología, como la

gestión del espectro radioeléctrico y la necesidad de un mayor procesamiento de señales digitales. Asimismo, consideraremos el impacto de SDR en la industria de las telecomunicaciones y su potencial para transformar las redes inalámbricas y la conectividad global.

La Radio Definida por Software ha abierto un mundo de posibilidades en las comunicaciones, y su creciente adopción en diversos sectores promete seguir impulsando la innovación en el futuro. Este TFG busca proporcionar una visión panorámica de SDR, profundizando en sus características técnicas y explorando su funcionamiento.

A medida que se avance en este estudio se espera adquirir una comprensión inicial completa de la Radio Definida por Software y su papel en la evolución de las tecnologías de comunicación.

1.3. Motivación

La elección de la Radio Definida por Software (SDR) como tema para este Trabajo de Fin de Grado (TFG) surge del reconocimiento de su creciente relevancia en el campo de las comunicaciones. SDR ha demostrado ser una tecnología transformadora que promete cambiar la forma en que nos comunicamos y conectamos en la sociedad actual.

La principal motivación para abordar este tema radica en la versatilidad única que ofrece SDR. Al permitir la implementación de funciones de radio mediante software en lugar de hardware específico, SDR brinda una flexibilidad y adaptabilidad sin precedentes en la creación de sistemas de comunicación. Esta característica tiene el potencial de revolucionar diversas áreas, desde las redes inalámbricas y la telefonía móvil hasta la radiodifusión y la seguridad de las comunicaciones.

Además, SDR promete reducir costos y mejorar la eficiencia en el desarrollo de soluciones de comunicación. La capacidad de reconfigurar y actualizar funciones mediante software abre la puerta a una mayor reutilización de recursos, lo que podría conducir a significativas reducciones de gastos en la implementación y mantenimiento de infraestructuras de comunicación.

El dinámico panorama tecnológico actual también motiva este estudio. La rápida evolución de la Internet de las cosas (IoT), la inteligencia artificial y la 5G demanda soluciones de comunicación más eficientes, seguras y adaptables. SDR se presenta como una respuesta prometedora a estos desafíos, y explorar su potencial en este TFG puede ayudar a identificar áreas de investigación y desarrollo con un impacto significativo en el futuro de las comunicaciones.

Asimismo, la Radio Definida por Software enfrenta desafíos interesantes en el ámbito técnico y regulatorio. Temas como el manejo eficiente del espectro radioeléctrico, la optimización del rendimiento de los sistemas SDR y la interoperabilidad con tecnologías tradicionales son fundamentales para entender el alcance y las limitaciones de esta tecnología.

A lo largo de este trabajo nos adentraremos en un área en constante evolución y de una tecnología innovadora. La investigación en SDR nos permitirá explorar cómo esta tecnología puede mejorar la vida de las personas, impulsar la conectividad global y potenciar la próxima generación de sistemas de comunicación.

En resumen, la motivación para realizar este TFG acerca de la Radio Definida por Software radica en su potencial transformador en las comunicaciones, su versatilidad y eficiencia, su relevancia en el contexto tecnológico actual y la oportunidad de aportar al avance de esta emocionante área de estudio. El objetivo es profundizar en los fundamentos y aplicaciones de SDR, comprendiendo su impacto en la sociedad y explorando cómo puede contribuir a resolver los desafíos de comunicación del siglo XXI.

1.4. Aplicaciones de la Radio Definida por Software

La Radio Definida por Software (SDR) ha demostrado ser una tecnología revolucionaria con un potencial sorprendente para transformar diversas áreas de las comunicaciones. Su flexibilidad y adaptabilidad hacen que sea una herramienta poderosa para abordar desafíos en el ámbito de las telecomunicaciones y explorar nuevas aplicaciones en el contexto actual.

Un ejemplo destacado de la aplicabilidad de SDR se encuentra en el desarrollo de las redes 5G. En un experimento real llevado a cabo por investigadores, se implementó un sistema SDR que permitía la transmisión y recepción de señales 5G utilizando múltiples frecuencias y modulaciones. La flexibilidad de SDR permitió realizar ajustes en tiempo real, lo que facilitó la integración de nuevas tecnologías y protocolos en la red 5G. Este experimento demostró cómo SDR puede acelerar el desarrollo y la implementación de las redes de próxima generación, brindando una mayor capacidad y una experiencia de usuario más eficiente.

Otro ejemplo relevante es el uso de SDR en la radiodifusión y televisión. Con la migración hacia la televisión digital terrestre, se han llevado a cabo experimentos en los que se han utilizado sistemas SDR para transmitir señales de televisión en diferentes bandas de frecuencia y formatos de compresión. Estos experimentos han permitido optimizar la eficiencia del espectro y mejorar la calidad de la señal, lo que ha llevado a una mayor cobertura y una mejor experiencia para los espectadores.

En el campo de la seguridad y defensa, SDR ha demostrado su utilidad en experimentos que implican la rápida reconfiguración de radios y sistemas de comunicación para adaptarse a diferentes escenarios operativos. Por ejemplo, en operaciones militares, se ha utilizado SDR para establecer comunicaciones seguras y cifradas, lo que permite una comunicación confiable y confidencial entre unidades en el campo de batalla. Además, SDR ha sido empleado en experimentos de vigilancia y detección, donde la adaptabilidad de la tecnología permite la detección y seguimiento de objetos en diferentes frecuencias y condiciones ambientales.

En el ámbito de la radioafición y experimentación, SDR ha ganado popularidad entre los entusiastas debido a su capacidad para adaptarse a diferentes bandas de frecuencia y modulaciones. Por ejemplo, los radioaficionados han utilizado SDR para explorar comunicaciones en bandas de radioaficionados y participar en concursos de radio, donde la adaptabilidad de SDR ofrece una ventaja competitiva al permitir ajustes precisos en la configuración de los equipos.

En el contexto del Internet de las cosas (IoT), SDR también ha desempeñado

un papel significativo. Se han realizado experimentos en los que se utiliza SDR para adaptarse a los diferentes protocolos y estándares utilizados por los dispositivos IoT, lo que facilita la integración y gestión de dispositivos inteligentes en la red. La flexibilidad de SDR también ha sido valiosa en el ámbito de la gestión del espectro radioeléctrico en el IoT, donde se busca optimizar la eficiencia del uso del espectro y evitar interferencias entre dispositivos.

En resumen, SDR ha demostrado ser una tecnología versátil y poderosa con aplicaciones en diversas áreas de las comunicaciones. Los experimentos reales han mostrado su capacidad para mejorar la eficiencia, optimizar el espectro radioeléctrico y brindar soluciones adaptativas a los desafíos actuales. La creciente adopción de SDR en la industria y la comunidad académica promete seguir impulsando la innovación y mejorando la forma en que nos comunicamos y conectamos en la sociedad moderna.

1.5. Estructura de la memoria

Tras una breve introducción a las comunicaciones y la radio definida por software se explicará detalladamente el funcionamiento de esta. Además se presentará el software y hardware disponible profundizando en los utilizados para la realización de los experimentos mostrados posteriormente así como los pasos necesarios a seguir para su instalación.

Más tarde se explicarán los fundamentos teóricos de las principales modulaciones tanto analógicas como digitales y se realizarán simulaciones de las mismas utilizando el software GNU Radio.

Una vez presentadas las modulaciones se procederá a detallar los experimentos realizados transmitiendo señales entre diferentes equipos utilizando los dispositivos HackRF como transmisor y un RTL-SDR como receptor.

Finalmente se explicarán las conclusiones obtenidas gracias a la realización de este trabajo y las posibles líneas futuras de ampliación del mismo.

Capítulo 2

Estado del arte

2.1. Introducción

El objetivo de este capítulo reside en dar a conocer el origen de la radio definida por software, la arquitectura típica utilizada en los receptores y transmisores utilizados en dicha tecnología y su evolución.

2.2. Historia de la Radio Definida por Software

En 1984 el equipo de Texas Garland División de E-Systems Inc., ahora conocido como Raytheon, fundó el término "Software defined radio" que fue desarrollado por E-System, responsable de la popularización de la radio definida por software en diversas dependencias del gobierno. Esta radio fue creada como un receptor de banda base digital y demodulación de señales de banda ancha, con la posibilidad de cancelación de interferencia. Utiliza múltiples procesadores array y acceso a memoria compartida, incluso con miles de ajustes y filtros. En 1988, el primer transceptor de radio definida por software fue diseñado en Alemania por Helmuth Lang y Peter Hoehner en el establecimiento de Investigación Aeroespacial Alemán. El responsable de aprobar el tema Radio definida por Software (1991) y publicar el primer trabajo sobre el mencionado tema fue Joseph Mitola en 1992 [2].

Los orígenes de la radio definida por software datan de finales de 1970 en

sectores de defensa de Europa y Estados Unidos, donde se tiene constancia de las primeras iniciativas publicadas que formaban parte de un proyecto creado por militares llamado SpeakEasy. La Fuerza Aérea Norteamericana trabajó en un sistema conocido como Integrated Communications Navigation, Identification and Avionics system (ICNIA) que utilizaba un DSP en el cual se realizaban las funciones de programación y control para obtener una plataforma integrada para comunicaciones aéreas. El principal objetivo de este proyecto residía en poder incorporar fácilmente en el futuro nuevos estándares de codificación y modulación haciendo uso de un proceso programable que permitiese comunicarse con más de 10 radios militares que operaban en las bandas de frecuencia de entre 2 y 2000 MHz.

En de la década de los 90, el departamento de defensa estadounidense creó el programa Joint Tactical Radio Systems que tenía como objetivo reemplazar las radios existentes en el ejército estadounidense por un solo conjunto de radios definidas por software, con nuevas frecuencias y modos, agregadas a través de la carga sustituyendo múltiples tipos de radio en tierra y terminando con el problema de interoperabilidad entre dispositivos.

En 1996 fue fundado el foro SDR Forum, en la actualidad conocido como Wireless Innovation Forum, cuyos miembros se dedican al avance de las tecnologías que respaldan la utilización innovadora del espectro y el desarrollo de sistemas de comunicaciones inalámbricas, incluidos los sistemas de comunicaciones esenciales o críticos.

2.3. Origen y motivación

A lo largo de la evolución de las telecomunicaciones inalámbricas han aparecido diversas tecnologías para el intercambio de información entre dos puntos distantes, con requisitos cada vez más exigentes. Las incompatibilidades entre las diferentes tecnologías han supuesto un problema a la hora de reutilizar equipos o prestar determinados servicios, como por ejemplo los terminales de telefonía móvil.

La radio definida por software surge para solucionar estos inconvenientes de compatibilidad e interoperabilidad, definiendo un conjunto de procedimientos y técnicas orientadas a realizar el procesamiento de señales de radio por

medio de un dispositivo de propósito general, el cual puede ser modificado mediante software logrando así un cambio dinámico, automático y eficiente entre tecnologías sin tener que incurrir en costes. De ahí surge la necesidad de diseñar un dispositivo capaz de entregar en banda base, o en frecuencia intermedia (IF) baja, la señal radio para su procesamiento digital, incluso si la señal es analógica. El dispositivo debe ser configurable para que sea posible recibir distintos anchos de banda y frecuencias para posteriormente, con un procesado digital, poder realizar la sincronización y la detección de la señal recibida.

2.4. Equipos de radio definida por software

Tradicionalmente los equipos receptores y transceptores de radiocomunicaciones son equipos constituidos por multitud de componentes electrónicos, los cuales forman circuitos sintonizadores, etapas de frecuencia intermedia, detectores, amplificadores de baja frecuencia, etc..., es decir, están constituidos por "hardware". Posteriormente, en los años 1980's y 1990's se introdujeron microprocesadores en estos equipos para el control de funciones internas (controles desde teclados y pulsadores) y para añadir nuevas prestaciones (relojes, pantallas informativas, programadores, etc...), y también se introdujo la posibilidad de controlar los equipos de radio desde un ordenador, añadiendo al equipo de radio puertos de comunicación o interfaces para la conexión al ordenador. En estos casos, y usando el software adecuado, es posible controlar desde el ordenador numerosas funciones del equipo de radio, igual o mejor que desde los controles del propio equipo. También en la década de los noventa comenzó la introducción en los modernos equipos de radio de los chips DSP o "Procesadores Digitales de Señal", los cuales permiten mediante técnicas digitales realizar filtros de paso de banda y de supresión de ruidos, entre otras posibilidades, muy eficaces, mejor que los realizados tradicionalmente con circuitos analógicos.

Un chip DSP es básicamente una especie de CPU en miniatura, un chip microprocesador con un juego de instrucciones pequeño, pero capaz de ejecutarlas a una velocidad muy superior a la velocidad de una CPU corriente, gracias a una estructura simplificada y al reducido juego de instrucciones.

En cualquier caso, siempre se trata de equipos de radio realizados entera-

mente con componentes electrónicos, o sea, en términos informáticos se definirían como radios hardware”. Pero desde principios de la década del 2000 radioaficionados como Gerald Youngblood (AC5OG), comenzaron a investigar y desarrollar un nuevo concepto de equipos de radiocomunicaciones, los equipos de radio desarrollados por programa o radios software”, en siglas SDR (Software Defined Radio), en los que la parte hardware (circuitería) es mínima, y la mayor parte de las funciones que definen un equipo de radio se definen por software (programas) en un ordenador PC o de otro tipo, dotado de tarjeta de sonido (requisito necesario).

2.4.1. Funcionamiento de un receptor SDR

Un receptor de SDR sigue necesitando una etapa frontal de radiofrecuencia (RF). En esta etapa, los débiles campos electromagnéticos captados por la antena son convertidos a una señal de RF cuya débil potencia debe ser filtrada en banda, ya sea con filtros fijos o ajustables, y amplificada adecuadamente. Una vez realizado esto, una configuración frecuente en los receptores SDR con la cual se trabajará en este trabajo, es la de realizar una conversión directa de la señal RF a la banda de audio.

Mediante la etapa frontal, la señal de radiofrecuencia se baja al rango de las frecuencias de audio aunque sigue siendo una señal sin demodular. Por lo tanto, el siguiente paso consiste en realizar la demodulación de la señal; para ello se hace uso de un ordenador con tarjeta de sonido y un software adecuado.

No debe confundirse la conversión directa explicada anteriormente con el proceso de demodulación a través del cual se extrae la señal de voz (moduladora) de la señal de RF. Dicha demodulación se realiza en el ordenador mediante un software específico.

La señal desplazada en frecuencia ocupa un margen de frecuencias que entra dentro del margen que acepta en sus entradas analógicas la tarjeta de sonido. Dependiendo de la tarjeta empleada, una tarjeta de sonido puede aceptar señales de hasta 20 a 40 kHz, y más altas..

Gracias a los conversores analógico-digitales (A/D) que incluye la tarjeta de sonido del equipo, se digitalizan las señales presentes en la entrada, generando un flujo continuo de bits. El conversor A/D está conectado casi

directamente a la antena. La tarjeta de sonido además proporciona todas las funcionalidades para el procesamiento digital de señales que son necesarias para manejar la señal entrante ya digitalizada.

Debido a la digitalización de las señales de entrada, se transforman las señales analógicas de entrada continuas en el tiempo en una secuencia de señales discretas en el tiempo. Una vez muestreada la señal analógica, las señales eléctricas que las componen tienen un número fijo de valores posibles de amplitud.

El término muestreo mencionado anteriormente consiste en el proceso por el cual las señales analógicas de entrada se transforman en señales discretas en el tiempo. El teorema de Nyquist establece que cuando se digitalizan señales analógicas, estas deben ser muestreadas a una velocidad igual o superior al doble de la frecuencia más alta presente en dicha señal para que pueda recuperarse la señal original mediante el proceso inverso. Para limitar la frecuencia máxima de la señal analógica de entrada es recomendable emplear un filtro pasobanda o filtro antialiasing, evitando así que al recuperar la señal original aparezca una distorsión, aliasing. El aliasing aparece cuando no se cumple el teorema de Nyquist del muestreo.

Con motivo de que el oído humano puede llegar a percibir frecuencias en el rango de 20 Hz a 20 KHz, las señales de audio de alta fidelidad requieren frecuencias de muestreo de al menos a 40 KHz. Dicha velocidad de muestreo es soportada por la tarjeta de sonido. Una vez la señal ha sido digitalizada, es necesario procesarla en general demodulándola. Este proceso se lleva a cabo mediante cálculos matemáticos utilizando un software adecuado. Los resultados numéricos de estos tratamientos matemáticos son también representados por señales digitales, las cuales se llevan internamente a unos conversores Digital-analógicos (D/A) implementados en la propia tarjeta de sonido, para convertirlas a señales de audio equivalentes, que son amplificadas y conducidas a las salidas de altavoz de la tarjeta de sonido, o procesadas para mostrar la información en la pantalla del ordenador.

2.4.2. Dispositivos

Existen multitud de transmisores y receptores SDR en el mercado cuyos precios y características son muy dispares.

Antes de comentar los diferentes modelos es importante analizar la terminología, características y especificaciones requeridas por un buen equipo SDR.

- SNR - Al recibir una señal, uno de los datos más importantes a medir es la relación señal a ruido, el ratio entre la intensidad máxima de la señal y la del ruido de fondo.
- Ancho de banda - Cuanto mayor sea el ancho de banda, más anchura de la señal se podrá observar en la pantalla a la vez y mayor será la SNR. Cuanto mayor sea el ancho de banda que se desee conseguir, mayor deberá ser la potencia de la CPU con la que se vayan a procesar las señales.
- Ancho de banda sin aliasing - El ancho de banda al visualizar señales provenientes de SDR tiene a disminuir en los bordes y también a mostrar imágenes o aliasing de otras señales. El ancho de banda libre de aliasing es el que realmente se puede utilizar, siendo, por lo general, más pequeño que el ancho de banda anunciado.
- Sensibilidad - Cuanto más sensible sea el receptor, con más facilidad se podrán percibir señales débiles y más alta será la relación señal a ruido.
- ADC Conversor Analógico - Digital - El ADC es el componente principal en SDR. Se encarga de muestrear una señal analógica para convertirla en bits. Cuanto más alta sea la resolución del ADC, con mayor precisión se muestreará.
- Sobrecarga - Se produce sobrecarga cuando una señal es demasiado fuerte y satura el ADC, no dejando espacio para que sean medidas otras señales más débiles. Cuando esto ocurre, la sensibilidad puede verse reducida severamente y pueden observarse frecuencias imagen de otras señales.
- Rango dinámico - El rango dinámico es la habilidad de un sistema SDR para recibir señales débiles cuando hay señales fuertes cerca. Está directamente relacionado con la resolución del ADC aunque puede verse afectado por el funcionamiento del DSP. La necesidad de un alto rango dinámico se puede solventar gracias a un filtrado de RF. La sobrecarga

se produce cuando una señal fuerte comienza a saturar el ADC porque el rango dinámico no era lo suficientemente alto.

- Imágenes / Aliasing - Es más probable que los SDRs de menor calidad se sobrecarguen y muestren imágenes de señales fuertes en frecuencias en las que no deberían estar. Este problema puede solucionarse mediante el filtrado o utilizando un receptor cuyo rango dinámico sea alto.
- Ruido/Interferencia - Los buenos dispositivos SDR no deberían recibir nada sin una antena conectada. Si reciben señales sin antena, es posible que las señales de interferencia entren directamente desde la paca del circuito haciendo imposible filtrarlas. Además, los buenos dispositivos son capaces también de adaptarse a las interferencias provenientes del USB.
- Filtro RF - Un equipo SDR de alto rendimiento constará de múltiples filtros preselectores que se activan en función de la frecuencia que se esté escuchando.
- Center DC Spike - Un buen dispositivo debe mantener equilibradas la fase y cuadratura de las señales para evitar picos de corriente continua.
- Ruido de fase - La calidad de los osciladores de cristal utilizados determina el rendimiento del ruido de fase. El uso osciladores cuyo ruido de fase es bajo mejora la SNR para señales de banda estrecha y provoca menos mezcla recíproca. Dicha mezcla recíproca ocurre cuando el ruido de fase alto hace que una señal débil se pierda en el ruido de fase de una señal fuerte cercana.
- Estabilidad frecuencial - El receptor debería permanecer sin desviarse en la frecuencia al variar la temperatura. Para conseguir esto se suele usar un TCXO o un oscilador similar estable.
- Diseño RF - Cabe destacar la importancia del diseño general del sistema. Por ejemplo, se han de tener en cuenta las pérdidas en los componentes utilizados en la ruta RF. A medida que aumenta la complejidad del diseño, se añaden componentes progresivamente, lo que puede afectar al rendimiento del receptor.

- Software - El hardware solamente es la mitad de un equipo SDR, si el software utilizado no es compatible con la unidad, puede deshacer por completo su utilidad.

Explicadas las principales características de los equipos a tener en cuenta. A continuación se precisará información de múltiples dispositivos entre los cuales se encuentran los mas vendidos y populares [3].

Airspy



Figura 2.1: Dispositivo Airspy

El dispositivo Airspy 2.1 requiere el uso de un equipo rápido y moderno; su web recomienda el uso de al menos un procesador Intel i3 de 2.4 GHz. Por otra parte, los creadores tener en cuenta la puntuación de la CPU, debiendo ser superior a 3500 para un correcto funcionamiento.

Es necesario asegurarse también de tener un puerto USB 2.0 de alta velocidad y calidad. Si existen errores en el controlador, no se podrá garantizar el ancho de banda completo requerido.

Los requisitos pueden verse reducidos ligeramente si se utiliza la función

de empaquetado de bits que proporciona Airspy.

Este dispositivo únicamente soporta tasas de muestreo de 10MSPS, es decir, 10 MHz por lo que se requiere una CPU muy potente. Es cierto que dispone también de una frecuencia de muestreo de 2,5 MHz pero según estudios el rendimiento del dispositivo se ve reducido.

Para su instalación solo es necesario conectar el receptor al puerto USB del PC y Windows plug and play instalará automáticamente los controladores. Posteriormente se podrá utilizar el dispositivo con cualquier software adecuado para ello, como por ejemplo SDR# en Windows. También es compatible con HSDR, SDR-Console y GQRX en Linux.

SDRplay RSP



Figura 2.2: Dispositivo SDRplay

Para el SDRplay 2.2 no se ofrecen requisitos concretos pero se estima que las especificaciones mínimas son significativamente más bajas que las que necesita el Airspy. Además admite anchos de banda más pequeños que logran reducir los requisitos de la CPU.

Según algunos informes, mejora su funcionamiento utilizando ordenadores más antiguos con procesadores como Core2Duos, aunque esto pueda restringir su frecuencia de muestreo máxima.

Para configurar el RSP en primer lugar es necesario instalar sus controladores desde el sitio web de SDRplay. Debido a que el RSP no cuenta con un software oficial, lo siguiente será la instalación de un complemento para SDR# o cualquier otro software compatible, como HDSDR.

Es compatible con SDR#, HDSDR y SDR-Console mediante el uso de complementos. Con SDR# la compatibilidad está restringida ya que no permite el uso complementos de terceros.

Los controladores de Linux están disponibles al igual que los complementos para SoapySDR, CubicSDR, Pothos y GNU Radio. El software CubicSDR también permite utilizar el RSP en el Mac.

El RSP está diseñado con filtros RF de conmutación y un LNA MGA-68563 justo en el frontend que solo se activa para señales superiores a 60 MHz. El chip sintonizador MSi001 contiene un segundo LNA en su interior. El MGA-68563 se utiliza como preamplificador para compensar las pérdidas en los filtros y con el objetivo de reducir el ruido de los LNA internos del sintonizador MSi001. La figura de ruido del MGA-68563 es de 1 dB, su ganancia de 19,7 dB y su OIP3 de 20 dBm.

Según las especificaciones del SDRplay, el MSi001 tiene una figura de ruido de alrededor de 4,5 dB en modo VHF y UHF.

El RSP consta de 8 filtros conmutados de selección automática así como de un filtro IF regulable dentro del chip MSi001 que puede ayudar a superar la interferencia de señales fuertes en banda. Para utilizar este filtro interno se debe reducir el ancho de banda de FI en la pantalla de configuración del RSP. Esto significa que al reducir el tamaño del filtro de FI verán menos de 8 MHz.

En conclusión, se observa que el SDRplay pretende optimizar su sensibilidad mediante el uso de un LNA frontal y se enfoca en solventar los efectos de la sobrecarga mediante de bancos de filtros conmutados.

HackRF



Figura 2.3: Dispositivo HackRF

El Hack RF 2.3 es un transceptor, permite recibir y emitir señales aunque no de forma simultánea. Su rango de frecuencias se sitúa entre 1MHz y 6GHz con un ancho de banda visible de hasta 20MHz. Posee tres amplificadores distintos para la recepción y dos para la emisión con controles de ganancia además de otros dos amplificadores que funcionan tanto para la transmisión como para la recepción.

No se proporcionan requisitos específicos a excepción de la necesidad de un buen puerto USB de alta velocidad para ejecutar el HackRF con las frecuencias de muestreo más altas.

A la frecuencia de muestreo más alta, 20 MSPS, se estima que son necesarios requisitos de CPU similares a los de Airspy.

Su cadena de RF es mucho más larga y más compleja que en los anteriores. Posee un MAX5864 con un ADC de 8 bits. Con el objetivo de que el HackRF funcione en un rango tan amplio tanto en recepción como en transmisión, se colocan muchos diodos de conmutación en la cadena de RF. Cada uno de estos conmutadores provoca una pérdida de señal de 0,35 a 0,5 dB, lo que puede explicar por qué HackRF tenga una sensibilidad moderada.

Debe tenerse en cuenta que en la frecuencia de recepción, el HackRF introduce un pico de potencia en forma de ruido como consecuencia de la corriente continua de los componentes internos por lo que siempre es recomendable sintonizarlo a una frecuencia cercana y establecer la frecuencia de recepción en otro punto del espectro, por ejemplo a 1 MHz por debajo de la central. Finalmente, se debe establecer un filtro paso banda que elimine el resto de la señal para que de este modo se obtenga la señal emitida.

Su instalación tanto en Windows como en otros sistemas operativos es similar a la de un RTL-SDR. Es preciso instalar el controlador WinUSB para HackRF. Es compatible con SDR#, HDSDR y SDR-Console en Windows, GQRX en Linux y RF Analyzer en Android. Existen varios programas en forma de código en GitHub, pero no muchas aplicaciones *plug and play*. Está diseñado y enfocado para utilizarse con softwares como GNU Radio.

2.5. Diagramas de bloques de los dispositivos utilizados

Los receptores de audio convencionales se componen de una etapa frontal de RF a la cual le siguen diferentes etapas de conversión de frecuencia y una etapa final de demodulación y amplificación de audio. Como se ha explicado anteriormente, en el caso de SDR, algunas de estas etapas son sustituidas por software.

Conviene conocer el esquema genérico del receptor y transmisor que se utilizarán para realizar diferentes experimentos que se encontrarán detallados en los siguientes capítulos con el objetivo de comprender los posibles problemas y soluciones que surgirán debido a su uso.

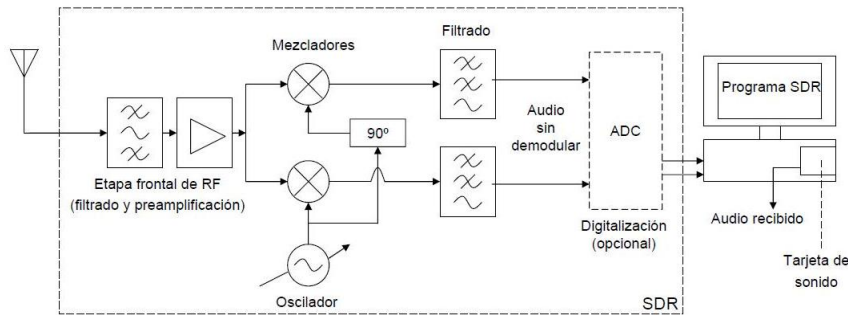


Figura 2.4: Esquema genérico de un receptor RTL-SDR

En la figura 2.4 se muestra un diagrama de bloques de conversión directa. En él se observa una etapa frontal de RF en la cuál se realiza el filtrado y amplificación de las señales captadas en la antena [4]. Una vez amplificada, la señal tomará dos ramas, en la primera se multiplicará con ayuda de un mezclador por el $\cos(2\pi * f_c * t)$ mientras que en la segunda será multiplicada por $\sin(2\pi * f_c * t)$, siendo f_c la frecuencia generada por el oscilador. Tras el mezclado, en ambas ramas la señal atravesará un filtro para llegar a un conversor analógico digital en el cual se digitalizarán ambas señales. Las muestras complejas llegarán al equipo donde serán procesadas con ayuda del software que se desee [5].

Una característica de los receptores SDR es la aparición de señales espúreas en forma de portadoras, espaciadas de la señal principal varias decenas o centenas de kHz , la mayoría ligeramente por encima del ruido interno del receptor. Para las espúreas más potentes, algunos receptores SDR cuentan con una función llamada dither, que consiste en añadir cierto ruido para "difuminar" dichas espúreas, a costa de una ligera subida del ruido de fondo del receptor. Estas señales son causadas en su mayor parte por el receptor. Estos productos de intermodulación se distinguen fácilmente porque el nivel de estos productos no es proporcional al nivel de las señales que los generan dado que surgen en el dominio digital. Es aconsejable para reducir las utilizar antenas que no resuenen en otras bandas. Posteriormente se verá como estas señales afectan a la recepción y qué posibles soluciones existen para contrarrestarlas.

Para la transmisión el proceso realizado es el inverso. Un transmisor SDR consta de varios bloques funcionales que trabajan juntos para convertir señales digitales en señales de radiofrecuencia. En la figura 2.5 se muestra un posible esquema empleado para ello.

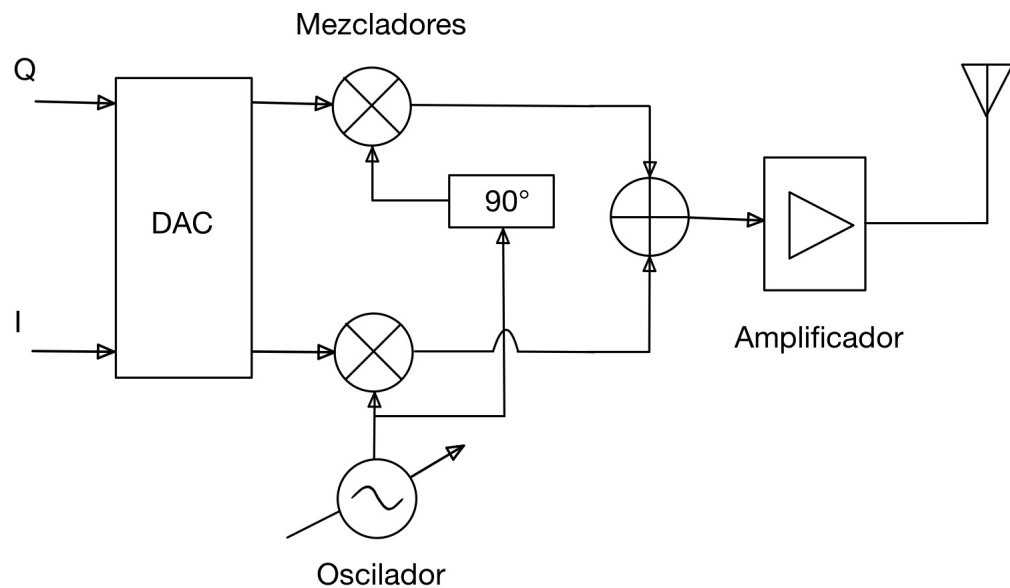


Figura 2.5: Esquema genérico de un transmisor SDR

En un transmisor SDR genérico entran dos flujos de muestras; uno de ellos en fase (I) y otro en cuadratura (Q). En primer lugar se realiza una conversión digital analógica por cada flujo de muestras utilizando un DAC; gracias a él, la señal modulada digital se convierte en una señal analógica que podrá ser transmitida a través del aire. Posteriormente, utilizando mezcladores y con ayuda de un oscilador local se multiplican las muestras en fase por $\cos(2\pi f_c t)$ y las muestras en cuadratura por $\sin(2\pi f_c * t)$ que serán posteriormente filtradas y sumadas siendo la señal resultante de la suma $m(t) \cos(2\pi f_c t) = Im(t) \cos(2\pi f_c t) - Qm(t) \sin(2\pi f_c t)$.

$Im(t)$ y $Qm(t)$ representan las componentes en fase y en cuadratura de la señal modulada como se ha explicado previamente. Estas componentes se pueden expresar matemáticamente como: $I(t) = \cos(2\pi f_c t)$ y $Q(t) = \sin(2\pi f_c t)$.

Posteriormente, haciendo uso de una antena, la señal será transmitida por el aire.

2.6. Software SDR

En el mundo de la radio definida por software existen numerosos softwares que se utilizan para la configuración, el control, la visualización y el procesamiento de señales de radio. A continuación se explican algunos de los más populares:

- **GNU Radio** es un software de procesamiento de señales de código abierto que se utiliza en sistemas de radio definidos por software. Es compatible con varios tipos de hardware SDR y permite a los usuarios diseñar su propio software de radio personalizado utilizando bloques de procesamiento de señales. GNU Radio tiene una amplia gama de características, incluyendo la capacidad de sintonizar múltiples frecuencias de radio, decodificar señales digitales y analizar espectros de frecuencia. Se profundizará en él a lo largo del próximo capítulo ya que es el software que elegido para realizar todos los experimentos que se describirán a lo largo del presente trabajo.
- **SDRSharp** (también conocido como SDR#) es un software de radio

definido por software popular y gratuito para Windows. Es compatible con varios tipos de hardware SDR y tiene una interfaz de usuario amigable y fácil de usar.

Tiene capacidad para recibir y decodificar una amplia gama de señales de radio, incluidas AM, FM, SSB, CW, ADS-B, DMR, POCSAG, APRS y más. Además, SDRSharp tiene una amplia gama de opciones de configuración, lo que permite a los usuarios personalizar su experiencia de radio definido por software.

SDRSharp también tiene características avanzadas, como la capacidad de analizar y visualizar espectros de frecuencia en tiempo real y la capacidad de grabar y reproducir transmisiones de radio. También cuenta con una función de descodificación automática, que permite a los usuarios descodificar señales digitales automáticamente y sin intervención manual.

Posee una función de visualización de waterfall, que muestra una vista en tiempo real del espectro de frecuencia a lo largo del tiempo. También tiene una función de búsqueda automática de frecuencia, que permite a los usuarios encontrar automáticamente señales de radio y ajustar la frecuencia en consecuencia.

Su capacidad para recibir una amplia gama de señales de radio, junto con sus características avanzadas, lo convierten en una herramienta para su uso en Windows.

- HSDR es un software para Windows compatible con varios tipos de hardware SDR. Su interfaz gráfica de usuario intuitiva y fácil de usar permite a los usuarios sintonizar y decodificar señales de radio en tiempo real.

Tiene capacidad para decodificar señales de radio digital, como DAB, DRM, HD Radio y DRM +. Además, HSDR es capaz de recibir señales de radio analógicas como AM, FM, LSB y USB. También posee una función de búsqueda automática de frecuencia, que permite a los usuarios encontrar señales de radio automáticamente y ajustar la frecuencia en consecuencia.

HDSDR cuenta con características avanzadas, como la capacidad de analizar y visualizar espectros de frecuencia en tiempo real y la capacidad de grabar y reproducir transmisiones de radio. Además, tiene una función de filtrado para mejorar la calidad de la señal y reducir el ruido.

Otra propiedad interesante es la función de mapeo de espectro, que muestra una vista detallada del espectro de frecuencia a lo largo del tiempo. También tiene una función de anotación, que permite a los usuarios agregar etiquetas y notas a las transmisiones de radio grabadas para facilitar la identificación y la búsqueda posterior.

Este software es una opción popular para uso en Windows. Su capacidad para decodificar señales de radio digital y analógicas, junto a sus características avanzadas, lo convierten en una buena herramienta.

- **SDRangel** es también un software libre y de código abierto ejecutable en múltiples plataformas, incluidas Windows, Linux y macOS. Es compatible con una amplia gama de hardware SDR y tiene una interfaz gráfica de usuario intuitiva y fácil de usar.

Una de las características notables de SDRangel es su capacidad para decodificar una amplia variedad de señales de radio, incluidas FM, AM, SSB, CW, DAB, DVB-T, ADS-B y más. Además, SDRangel tiene una amplia gama de opciones de configuración que permiten a los usuarios personalizar su experiencia de radio definido por software.

SDRangel también cuenta con características avanzadas, como la capacidad de analizar y visualizar espectros de frecuencia en tiempo real, así como la capacidad de grabar y reproducir transmisiones de radio. Posee una función de detección automática de señal, que permite a los usuarios encontrar y decodificar automáticamente señales de radio, y una función de seguimiento de satélite, que permite a los usuarios rastrear satélites y recibir sus transmisiones.

Otra de sus características es su capacidad para integrarse con otros programas de radio definida por software, lo que permite a los usuarios crear flujos de trabajo personalizados y aumentar su eficiencia y pro-

ductividad. Además, su comunidad activa de desarrolladores y usuarios que contribuyen al software y ofrecen soporte técnico a través de foros y otras plataformas en línea.

En general, SDRangel es una opción sólida y popular para aquellos que buscan un software de radio definido por software de código abierto para su uso en múltiples plataformas. Su capacidad para decodificar una amplia gama de señales de radio y sus características avanzadas lo convierten en una herramienta útil para los radioaficionados.

- **CubicSDR** es también un software gratuito y de código abierto que se ejecuta en los sistemas operativos macOS y Linux. Es compatible con varios tipos de hardware SDR y tiene una interfaz gráfica de usuario intuitiva y fácil de usar.

Su principal cualidad es su capacidad para decodificar señales de radio digital y analógicas, incluyendo AM, FM, LSB, USB, CW, DAB, DVB-T, ADS-B y más. Además, CubicSDR tiene una función de búsqueda automática de frecuencia que permite a los usuarios buscar y sintonizar señales de radio automáticamente.

CubicSDR también cuenta con características avanzadas, como la capacidad de analizar y visualizar espectros de frecuencia en tiempo real y la capacidad de grabar y reproducir transmisiones de radio. Además, tiene una función de mapeo de espectro que permite a los usuarios ver la actividad de radio en una determinada frecuencia en un período de tiempo determinado.

Se integra fácilmente con otros programas de radio definida por software, lo que permite a los usuarios crear flujos de trabajo personalizados y aumentar su eficiencia y productividad. Además, CubicSDR tiene una comunidad activa de desarrolladores y usuarios que contribuyen al software y ofrecen soporte técnico a través de foros y otras plataformas en línea.

Resulta una opción sólida y popular de software para SDR de código abierto para su uso en sistemas operativos macOS y Linux. Su capacidad para decodificar una amplia gama de señales de radio y sus

características avanzadas hace que sea utilizado por numerosos radioaficionados.

- **HD Radio Decoder** es un software gratuito y de código abierto que permite la decodificación de señales de radio HD, una tecnología utilizada en la transmisión de señales de radio de alta calidad en formato digital. Este software necesita del sistema operativo Windows y es compatible con una amplia gama de hardware SDR.

Como ya se ha mencionado, una de las características más destacables de HD Radio Decoder es su capacidad para decodificar señales de radio HD en tiempo real, lo que permite a los usuarios escuchar estaciones de radio en alta definición. Además, el software también es capaz de decodificar señales analógicas de radio FM, lo que lo convierte en una herramienta útil para aquellos que desean mejorar la calidad de audio de sus estaciones de radio locales.

HD Radio Decoder también cuenta con características avanzadas, como la capacidad de analizar y visualizar espectros de frecuencia en tiempo real y la capacidad de grabar y reproducir transmisiones de radio HD. Además, posee una función de búsqueda automática de frecuencias que permite a los usuarios buscar y sintonizar señales de radio automáticamente.

Otra característica interesante de HD Radio Decoder es su interfaz gráfica de usuario fácil de usar, que permite a los usuarios configurar fácilmente su hardware SDR y ajustar la configuración del software para obtener la mejor calidad de audio posible.

En resumen, este software es de utilidad para decodificar señales de radio HD y mejorar la calidad de audio de sus estaciones de radio locales. Su capacidad para decodificar señales de radio HD en tiempo real y sus características avanzadas lo convierten en una herramienta interesante para los radioaficionados.

- **Linrad** es un software gratuito y de código abierto que se centra en la recepción y procesamiento de señales de radio débiles, como las señales de radio astronómicas, y está disponible para Windows y Linux.

Una de las características notables de Linrad es su capacidad para procesar señales de radio débiles y reducir el ruido de fondo, lo que mejora la calidad de la señal de radio. Además, el software también es capaz de realizar un seguimiento preciso de la frecuencia y el tiempo, lo que permite una decodificación precisa de señales de radio de alta velocidad.

Linrad también cuenta con una amplia gama de funciones avanzadas, como la capacidad de procesar señales de radio de hasta 24 bits y la capacidad de realizar un análisis de espectro en tiempo real. Además, este software también es compatible con una amplia gama de hardware SDR, lo que permite a los usuarios elegir el hardware que mejor se adapte a sus necesidades.

Otra característica interesante de Linrad es su interfaz gráfica de usuario altamente personalizable, que permite ajustar la configuración del software y la visualización de datos para adaptarse a necesidades específicas. Además, el software también tiene una función de calibración automática que ayuda a los usuarios a ajustar su hardware SDR para obtener la mejor calidad de señal posible.

Se puede considerar una herramienta valiosa para el procesamiento y mejora de calidad de señales de radio débiles. Su capacidad para procesar señales de hasta 24 bits, realizar un seguimiento preciso de la frecuencia y el tiempo, y su amplia gama de características avanzadas lo convierten en una opción sólida para los radioaficionados.

Capítulo 3

Tecnología utilizada

3.1. Software

GNU Radio es una herramienta de desarrollo abierta y libre que proporciona una serie de bloques a través de los cuales es posible realizar un procesamiento de la señal con el fin de implementar sistemas de radio definida por software. Está enfocado tanto a su utilización en entornos de simulación como a la integración con hardware permitiendo crear radios de bajo coste. Su arquitectura modular y su capacidad para trabajar con diferentes dispositivos de hardware y software la convierten en una herramienta muy flexible y adaptable a las necesidades específicas de cada usuario.

Para todos los sistemas de radio definidos por software, la gran capacidad de configuración es fundamental. De esta forma, adquiriendo una simple radio genérica que utilice procesamiento de señal por software, se puede reemplazar cualquier tipo de radio. Actualmente solo es posible procesar en GNU Radio señales concretas, pero conociendo el funcionamiento matemático de un sistema de transmisión es viable configurar GNU Radio para recibirlo.

Como se ha mencionado anteriormente, una de las principales ventajas de GNU Radio es su capacidad para trabajar con radios definidas por software (SDR), que son dispositivos de hardware que pueden configurarse y reconfigurarse mediante software para realizar diferentes tareas de procesamiento de señales. Esto permite que los usuarios puedan diseñar y desarrollar sistemas de comunicación personalizados y adaptados a sus necesidades específicas.

Gracias a GNU Radio Companion, el entorno gráfico proporcionado por GNU Radio, se pueden construir aplicaciones aunque también es viable programarlas utilizando el lenguaje Python o C++ para casos en los que se requiera un alto rendimiento. De esta forma se pueden desarrollar sistemas de radio en tiempo real y con un rendimiento alto empleando el entorno de desarrollo de aplicaciones proporcionado.

El funcionamiento de GNU Radio se basa en el procesamiento de señales digitales en tiempo real. Para lograr esto, utiliza un conjunto de bloques funcionales que se conectan entre sí para crear un diagrama de flujo gráfico que representa el flujo de procesamiento de señales que se realizará. Estos bloques pueden realizar una amplia variedad de tareas, como la modulación, demodulación, filtrado, muestreo, decodificación, codificación, y mucho más. Los bloques se organizan en diagramas de flujo que representan el flujo de datos que se procesará.

En GNU Radio, cada bloque tiene una entrada y una salida, y los bloques se conectan entre sí mediante cables que representan el flujo de datos entre los bloques. Cada cable tiene un tipo de datos asociado que indica el tipo de información que se está transmitiendo. Por ejemplo, puede haber cables de tipo de datos flotante para la transmisión de señales de audio o cables de tipo de datos complejos para la transmisión de señales moduladas. Cuando se ejecuta un diagrama de flujo en GNU Radio, los bloques se procesan en tiempo real para llevar a cabo las tareas de procesamiento de señales que se han definido en el diagrama de flujo. Los datos se mueven a través de los cables entre los bloques, y cada bloque realiza su tarea específica en el flujo de datos.

Es importante tener en cuenta que GNU Radio no realiza la función de la radio en sí misma, sino que proporciona una plataforma de software para el diseño y desarrollo de sistemas de procesamiento de señales de radio. Para utilizar GNU Radio en un sistema de radio real, es necesario contar con un hardware que pueda recibir y transmitir señales de radio, dispositivos de radio definida por software (SDR) y una tarjeta de sonido con capacidad de entrada y salida de señales.

3.1.1. Instalación de GNU Radio Companion 3.10.1.1 en Linux

Para instalar la versión 3.10.1.1 que es la utilizada en los siguientes epígrafes se deben llevar a cabo los siguientes pasos [6]:

En primer lugar es necesario abrir el terminal. En este caso es preciso añadir un nuevo repositorio PPA, en este caso el de GNU Radio, para ello se teclea el comando que se muestra a continuación:

1. **sudo add-apt-repository ppa:gnuradio/gnuradio-releases**

Posteriormente se realiza una actualización del gestor de paquetes:

2. **sudo apt-get update**

Finalmente, se realiza la instalación mediante el siguiente comando:

3. **sudo apt-get install gnuradio python3-packaging**

Es posible que dependiendo de la versión que se desee instalar el repositorio a instalar cambie. En este caso, es sencillo buscar una guía de instalación en la comunidad virtual de GNU Radio la cuál se encuentra en el siguiente enlace: https://wiki.gnuradio.org/index.php/Main_Page.

3.2. Hardware

3.2.1. HackRF One

Previamente se ha hecho una introducción a los dispositivos de esta marca. En este caso se profundizará en las características del HackRF One en concreto.

Este periférico de radio definida por software es capaz de transmitir o recibir señales de radio de 1 MHz a 6 GHz, diseñado para permitir la prueba y el desarrollo de tecnologías de radio modernas y de próxima generación. De acuerdo a las especificaciones de su fabricante, HackRF One es una plataforma de hardware de código abierto que se puede usar como un periférico USB o programado para operaciones independientes.

Las características principales del dispositivo HackRF One establecidas por el fabricante, Greats Scott Gadgets, se muestran a continuación:

- Frecuencia operativa de 1 MHz a 6 GHz
- Transceptor semidúplex.
- Hasta 20 millones de muestras por segundo
- Muestras de cuadratura de 8 bits (8 bits I y 8 bits Q)
- Compatible con GNU Radio, SDR # y más ganancia de RX y TX configurable por software y filtro de banda base.
- Alimentación del puerto de antena controlada por software (50 mA a 3,3 V)
- Conector de antena hembra SMA
- Entrada y salida de reloj hembra SMA para sincronización.
- Botones convenientes para programar.
- Cabezales de clavija internos para expansión.
- USB 2.0 de alta velocidad.
- Alimentado por USB.
- Hardware de código abierto

Instalación de controladores del HackRF One en Linux

Para poder utilizar el HackRF en linux es imprescindible instalar sus controladores, para ello es conveniente seguir los siguientes pasos:

La instalación se realiza desde el terminal. En primer lugar se actualiza el gestor de paquetes:

1. `sudo apt-get update`

Una vez actualizado ya es viable instalar el controlador; para ello se utilizará el siguiente comando:

2. **sudo apt-get install hackrf**

Tras estos pasos ya se estará en disposición de utilizar el HackRF ya que el equipo logrará controlarlo.

3.2.2. **RTL-SDR**

Con el fin de recibir la señal transmitida desde el HackRF se utilizará un RTL-SDR genérico como es el RTL2832U que puede ser utilizado como un dongle DVB-T o como un receptor SDR. Su chip es capaz de transferir las muestras I/Q que se utilizarán en el proceso de demodulación.

Por lo general, este tipo de receptores pueden recibir en el rango de frecuencias de 50 KHz a 1766 MHz. Contienen un ADC de 8 bits. Es posible utilizarlo conectándolo mediante un USB al ordenador aunque es necesario instalar un controlador previamente.

Instalación de controladores del RTL-SDR en Linux

Para poder utilizar el RTL-SDR en linux es necesario instalar previamente sus controladores, como se ha explicado con anterioridad, para ello es conveniente seguir los siguientes pasos: La instalación se realiza desde el terminal. En primer lugar se actualiza el gestor de paquetes:

1. **sudo apt-get update**

Una vez actualizado ya es viable instalar el controlador; para ello se utilizará el siguiente comando:

2. **sudo apt-get install osmocom-nitb**

Tras estos pasos ya se estará en disposición de utilizar el USRP ya que el equipo logrará controlarlo.

Capítulo 4

Modulaciones analógicas

4.1. Introducción

El proceso de modulación de una señal consiste en la modificación de algún parámetro de la señal portadora, ya sea la amplitud, la fase o la frecuencia, con el fin de poder transmitir información desde un equipo transmisor hasta otro receptor. Las modificaciones en la portadora se realizan de acuerdo a las características de la señal de información o moduladora para obtener como resultado una señal modulada [7]. Típicamente, la señal portadora suele ser un seno o un coseno cuya frecuencia es muy superior a la de la señal de información.

En función de la naturaleza de las señales moduladora y portadora se distinguen distintos tipos de modulaciones:

	Portadora analógica	Portadora digital
Señal analógica	Modulación analógica	Modulación por pulsos
Señal digital	Modulación digital	Códigos de línea

Cuadro 4.1: Modulaciones monoportadoras

4.2. Modulaciones analógicas

En este tipo de modulaciones, tanto la señal portadora como la moduladora son analógicas.

Sea $x(t)$ la señal a transmitir (señal de potencia) en banda base y de ancho de banda B , donde $X(f) = 0, |f| > B$ (asumiendo que B es el espectro positivo de la señal) y $c(t)$ la portadora caracterizada como $c(t) = A_c \cos(2\pi f_c t + \phi_c)$, donde A_c es la amplitud, f_c la frecuencia y ϕ_c la desviación de fase de la portadora. Esta notación se utilizará posteriormente, llamando $y(t)$ a la señal modulada, la resultante.

En función de cuál sea el parámetro de la señal portadora que module la señal de información, se distinguen distintos tipos de modulaciones analógicas:

- Modulación en amplitud
- Modulación en frecuencia
- Modulación en fase

4.3. AM: Modulación en amplitud

4.3.1. Fundamentos de la modulación en amplitud

La modulación en amplitud es un tipo de modulación lineal cuyo principio de funcionamiento es hacer que la amplitud de la señal portadora dependa de la señal moduladora.

Típicamente se trabajará con una señal de tipo senoidal para generar la forma de onda de la señal [8]. La señal portadora será de la forma:

$$c(t) = A_c \cos(\omega_c t); \omega_c = 2\pi f_c \quad (4.1)$$

En la ecuación anterior 4.1 A_c y f_c hacen referencia a la amplitud y la frecuencia de la portadora respectivamente.

Una vez definida forma de la portadora, la modulación en amplitud se puede entender como el proceso en el que la amplitud de la onda portadora $c(t)$ se varía en torno a un valor medio, linealmente, con la señal en banda base $m(t)$ 4.2.

$$s(t) = A_c [1 + k_a m(t)] \cos(\omega_c t) \quad (4.2)$$

En donde k_a es la constante de sensibilidad que aparece en la generación de la señal modulada $s(t)$. Al valor máximo de $k_a m(t)$ se le llama índice de modulación. A continuación se muestra un diagrama de bloques 4.1 que permite observar gráficamente lo explicado.

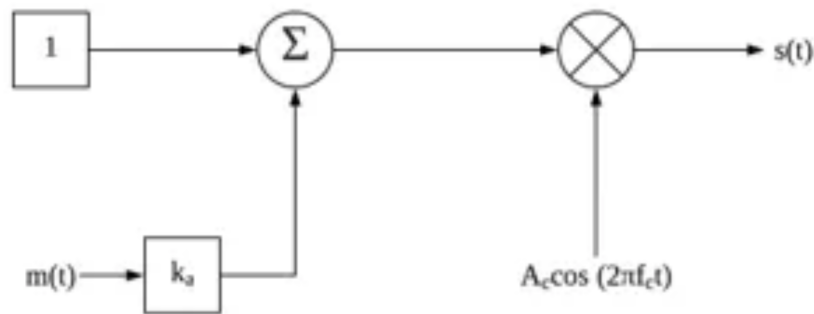


Figura 4.1: Diagrama de bloques AM

En la señal modulada 4.2 se podrá observar la envolvente que presentará la misma forma que la señal banda base. Cabe resaltar que la envolvente es una función positiva, por lo tanto, la envolvente a tener en cuenta sería la que contiene únicamente valores positivos. El índice de modulación permite controlar la modulación o sobremodulación de la onda portadora. En el caso de sobremodulación se generan inversiones de fase de la portadora siempre que se cruza por cero, en dicho caso se provocará distorsión de la envolvente. A continuación, se muestran gráficas donde se observan estos casos.

La señal mostrada 4.2 corresponde a un caso en el que no se da sobremodulación [9]. Posteriormente se mostrará un caso de sobremodulación 4.7, en los cambios de fase señalados no se logra observar de la mejor manera dichos cambios, esto se debe al cambio de fase de $s(t)$ cada vez que la señal $m(t)$ pasa por cero.

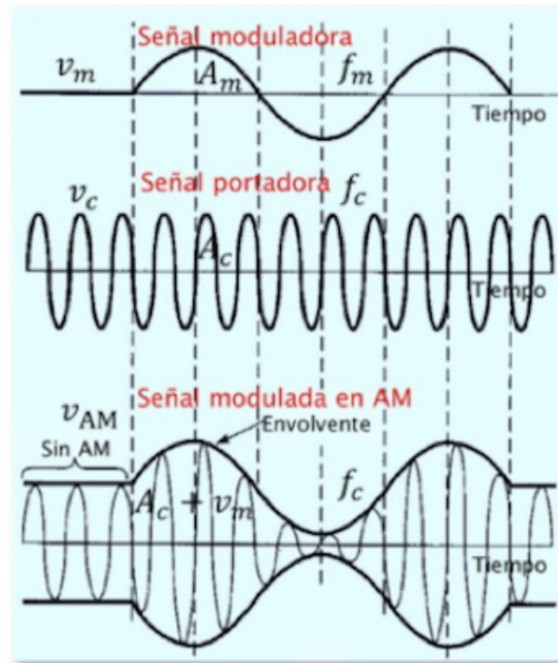


Figura 4.2: Señal modulada en amplitud

4.3.2. Simulación de la modulación en amplitud en GNU Radio

Tras haber explicado los fundamentos teóricos de la modulación en amplitud se procede a simularla mediante el software explicado en el capítulo 3, GNU Radio.

En la figura 4.4 se muestra el esquema completo de modulación en dicho software.

En primer lugar se toma una señal de información, en este caso será un coseno de frecuencia 1kHz y amplitud 1. Posteriormente se añade un bloque multiplicador con un parámetro variable con el objetivo de poder añadir la ganancia de entrada deseada antes de modular la señal, su valor no debe ser superior a 1 ya que de ser así se produciría sobremodulación; en este caso el valor elegido será de 0.7.

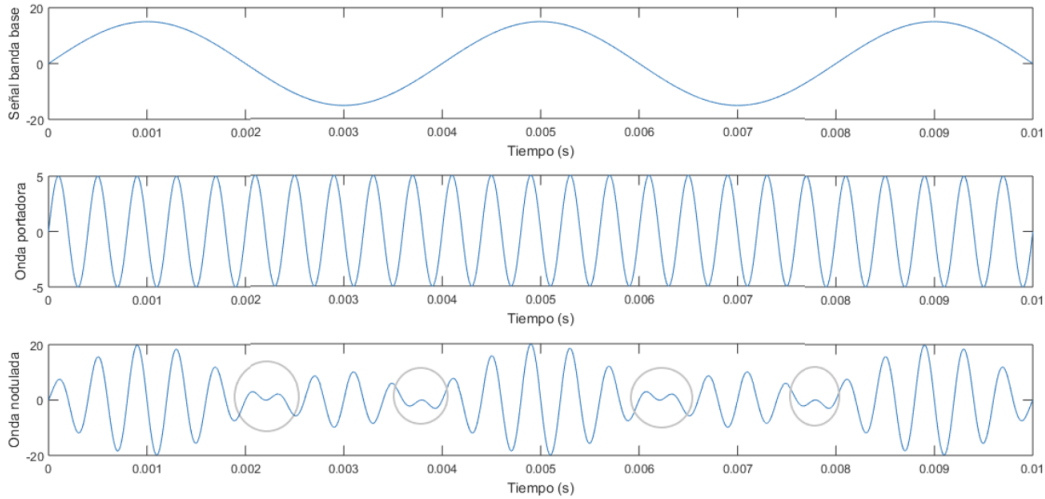


Figura 4.3: Señal con sobremodulación

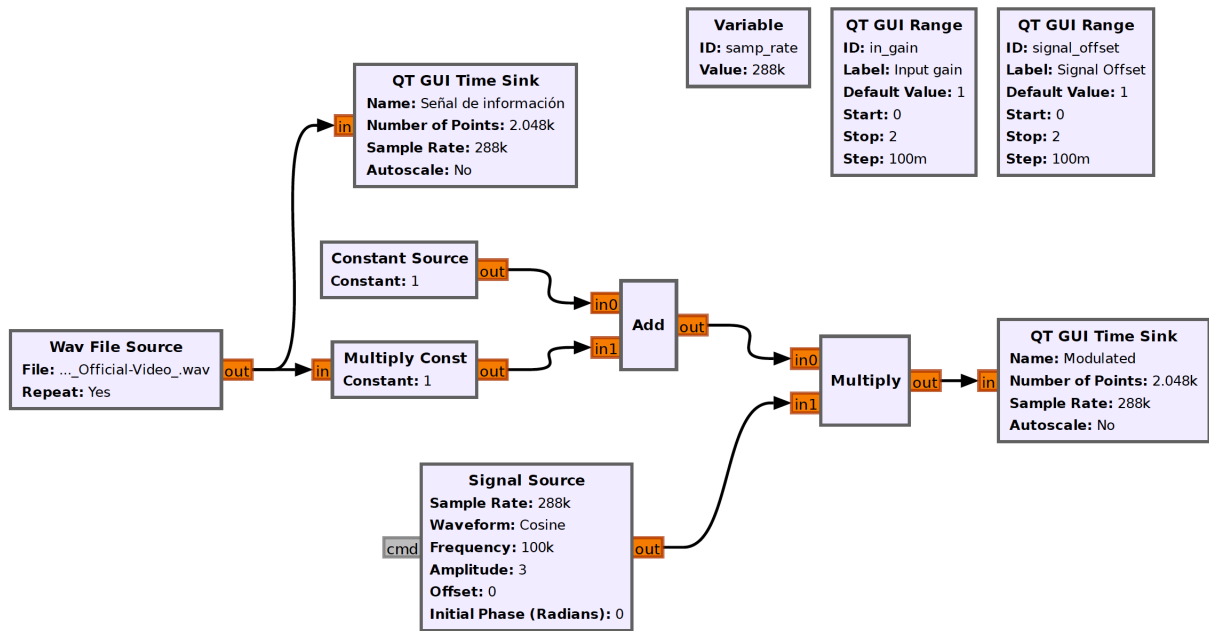


Figura 4.4: Esquema de modulación AM en GNU Radio

Siguiendo la expresión 4.2 detallada en la sección anterior, se suma una constante de 1 a la señal que seguidamente se multiplicará por un coseno para obtener la señal modulada. Dicho coseno es la señal portadora, en este caso se elige una frecuencia de 100 kilohercios y una amplitud de 3.

A continuación se muestra una evolución de la señal 4.5. En ella se observa lo explicado anteriormente. La imagen superior presenta la señal de información o moduladora mientras que la inferior corresponde a la señal modulada; en ella se observa cómo la portadora modifica su amplitud en función de la señal moduladora.

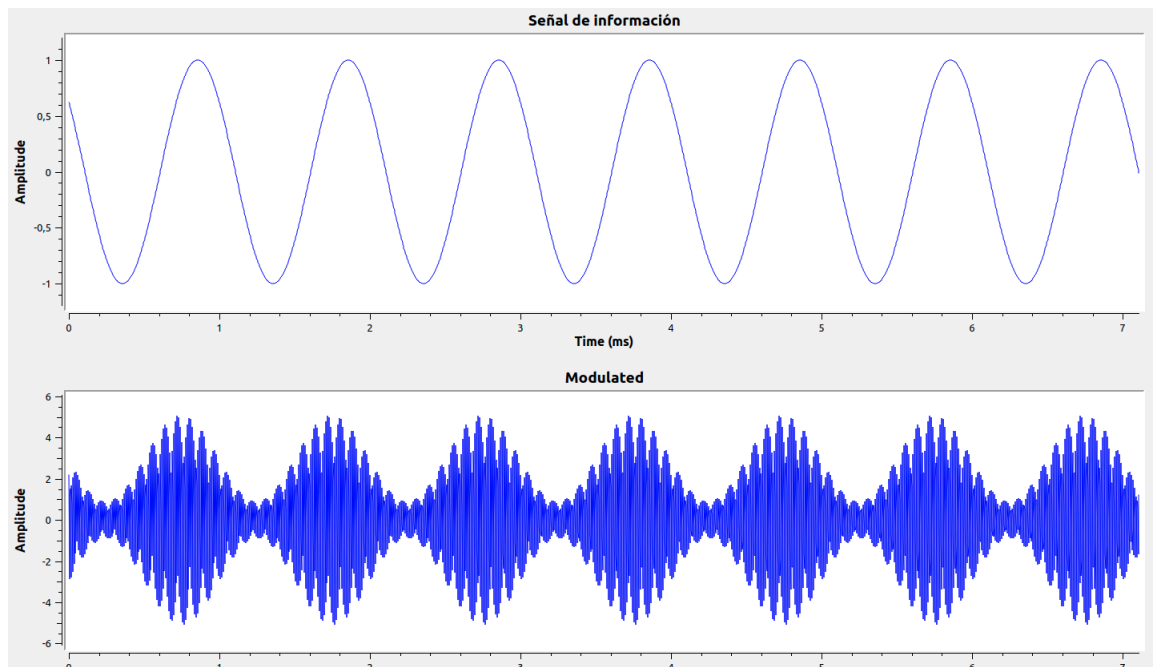


Figura 4.5: Señal de información y señal modulada AM

Finalmente se muestra otro ejemplo para el cuál se elige como señal de información un archivo .wav que contiene audio adecuado para el programa. En este caso la señal modulada resultante se muestra en la figura 4.6.

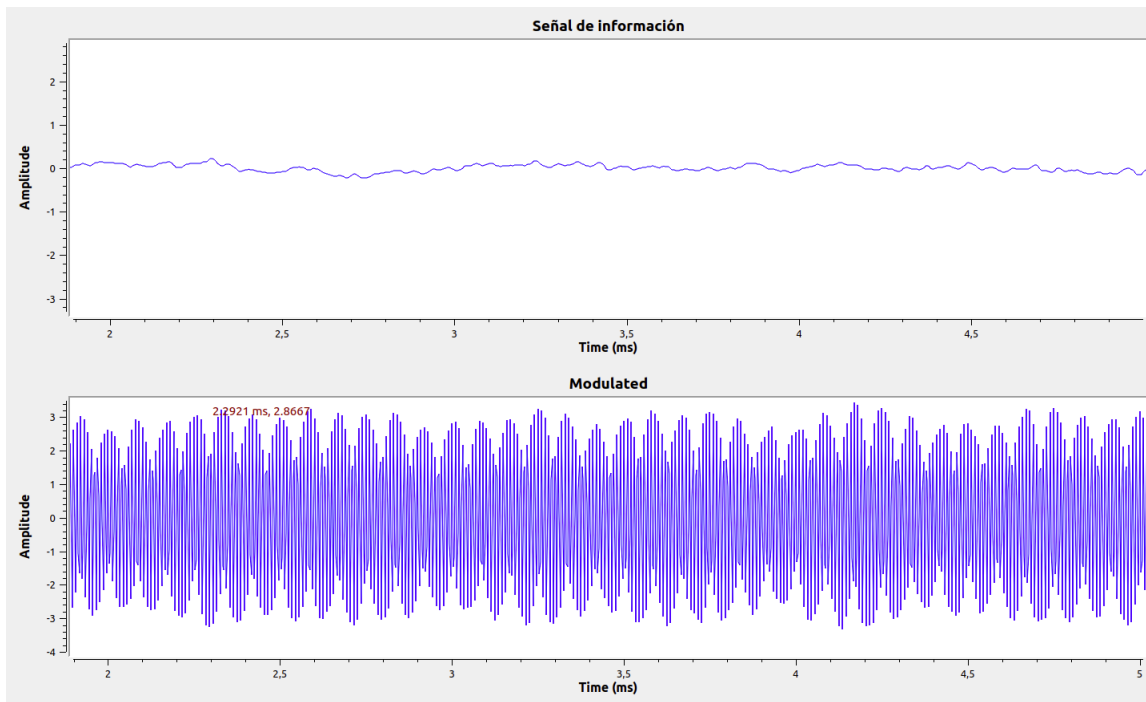


Figura 4.6: Señal de información y señal modulada AM utilizando un archivo de audio

4.3.3. Fundamentos de la demodulación en amplitud

Para demodular una onda AM, la portadora y la porción de la envolvente que lleva la información se trasladan del espectro de radiofrecuencia a la fuente original de información. Se pueden utilizar dos métodos diferentes para realizar dicha demodulación [10]:

- En una señal AM típica, la portadora se mantiene constante en amplitud y frecuencia, lo que permite una demodulación relativamente sencilla. Sin embargo, en algunas aplicaciones o condiciones de transmisión, puede ser beneficioso reducir o incluso suprimir completamente la portadora para ahorrar energía o mejorar la eficiencia del sistema.

Cuando se elimina o se reduce significativamente la portadora en una señal AM, la demodulación convencional utilizando un detector de envolvente ya no es efectiva. En su lugar, se requiere una demodulación

coherente, que consiste en recuperar tanto la amplitud como la fase de la señal original.

Para realizar la demodulación coherente, se utiliza un detector de fase. Este detector compara la fase de la señal modulada con una referencia de fase precisa. Aquí es donde entra en juego el oscilador a nivel local sincronizado con la portadora original. El oscilador local genera una señal de referencia con la misma frecuencia y fase que la portadora original, lo que permite al detector de fase realizar la comparación adecuada.

Al tener un oscilador local sincronizado con la portadora original, se logra una demodulación coherente precisa. El detector de fase puede extraer tanto la amplitud como la fase de la señal modulada, lo que permite recuperar la información transmitida de manera efectiva.

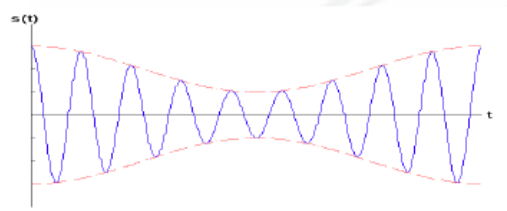
- El detector de envolvente es un método muy simple de demodulación que consta de un rectificador y un filtro paso bajo. Idealmente produce una señal de salida que sigue la envolvente de la señal de entrada. Para demodular una señal AM se requiere una señal de banda estrecha en la que no se produzca sobremodulación. A continuación se muestra un ejemplo de una señal a demodular 4.3.3 y la correspondiente salida del detector de envolvente 4.3.3.

4.3.4. Simulación de la demodulación en amplitud en GNU Radio

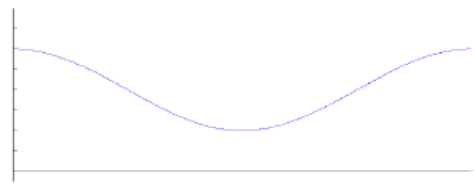
Conocidos los fundamentos teóricos de la modulación en amplitud, resulta conveniente realizar su simulación en GNU Radio.

Se pretende recuperar la señal de información original partiendo de la modulada, para ello se propone el esquema de la figura 4.8 en GNU Radio:

Para demodular la señal el primer paso a tener en cuenta es su filtrado, en este caso se utiliza un filtro paso banda cuyo ancho de banda es de aproximadamente 20 kilohercios centrándolo en la frecuencia central de la portadora de 100 kilohercios con la que se moduló la señal; de esta forma se seleccionan únicamente las frecuencias de interés.



(a) Señal modulada AM



(b) Señal demodulada

Figura 4.7: Señal modulada en amplitud y su demodulación mediante un detector de envolvente.

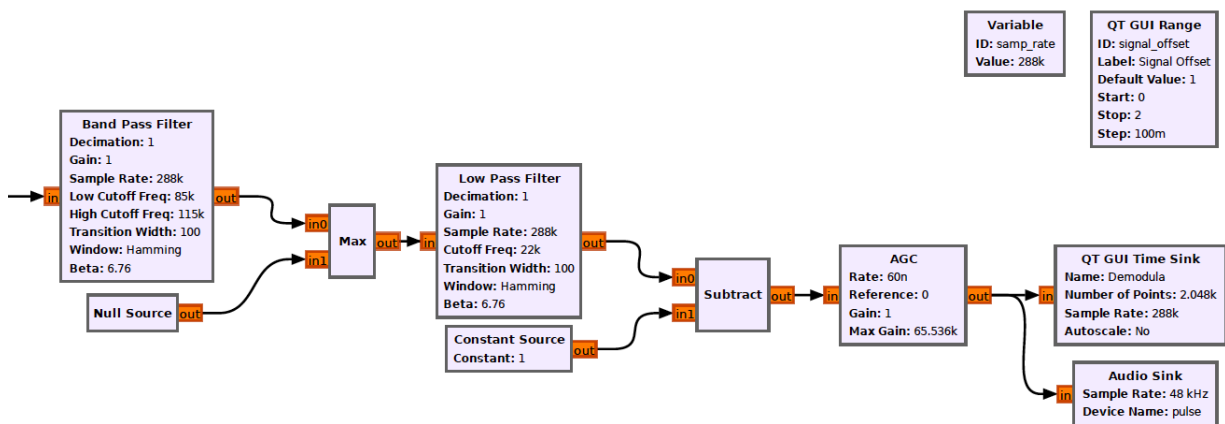


Figura 4.8: Esquema de demodulación AM en GNU Radio

Seguidamente se simula el comportamiento de un diodo utilizando un bloque diseñado en GNU Radio que selecciona el máximo de dos entradas, una de ellas la señal filtrada y la otra una fuente de ceros; de esta forma se eligen únicamente los semiciclos positivos. La salida de este bloque se filtra utilizando

do un filtro paso bajo para seleccionar su envolvente; la frecuencia de corte del filtro utilizado será de 22 kilohercios ya que debe ser la de la señal, en este caso, de audio.

Una vez simulado el detector de envolvente es necesario eliminar el offset que se había añadido en la modulación. Por último, para ajustar la ganancia, dado que la señal es de audio, se utiliza un bloque de control automático de ganancia.

A continuación se muestra la señal modulada a la entrada del demodulador conjuntamente con la señal demodulada 4.17; de esta forma se podrá observar cómo se logra recuperar la señal original de información tras el proceso detallado anteriormente.

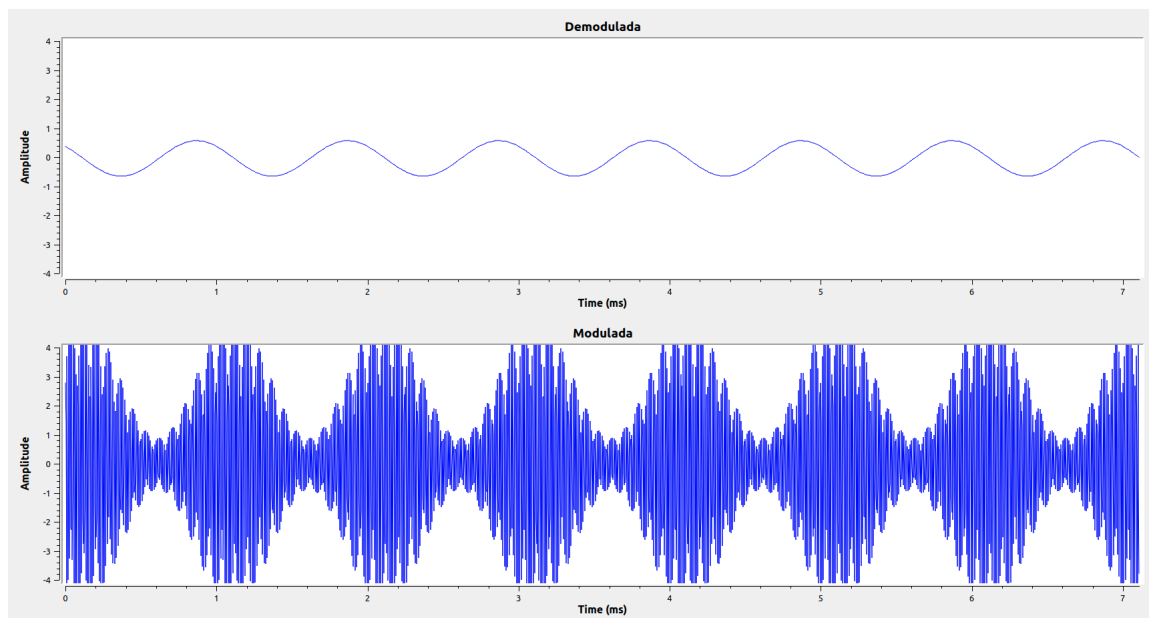


Figura 4.9: Señal modulada y señal demodulada AM

Al aumentar el valor del multiplicador situado a la entrada del modulador a valores superiores a 1 se observan los efectos de la sobremodulación tanto en la señal modulada como en la demodulada como se aprecia en la figura 4.10.

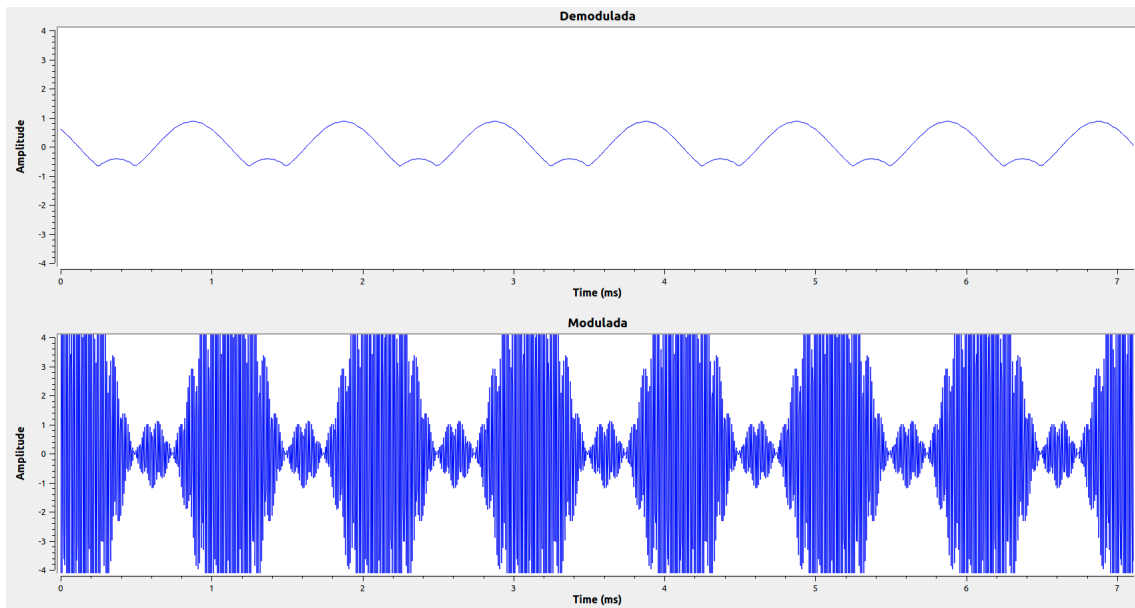


Figura 4.10: Sobremodulación

En último lugar se prueba a demodular la señal de audio simulada anteriormente. Se observa en la figura 4.11 como se logra recuperar la señal de información y gracias al bloque *Audio Sink* se escucha el audio sin modificaciones respecto de la señal de partida.

4.4. FM: Modulación en frecuencia

4.4.1. Fundamentos de la modulación en frecuencia

La modulación en frecuencia es una modulación angular que transmite la información a través de las variaciones de frecuencia de una onda portadora a diferencia de la modulación en amplitud en la cuál la frecuencia permanece constante mientras varía la amplitud. La frecuencia instantánea de la señal modulada varía de forma lineal con la señal en banda base [8].

La señal moduladora se introduce en la frecuencia de la portadora 4.3 que suele ser de la siguiente forma:

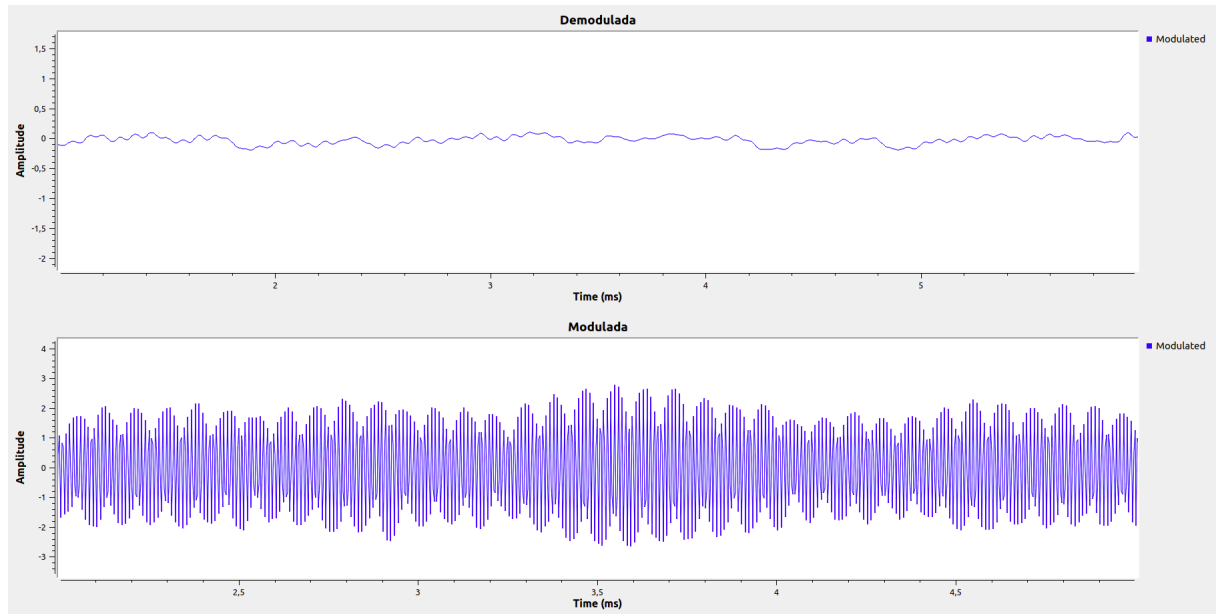


Figura 4.11: Señal modulada y señal demodulada AM utilizando un archivo de audio

$$V_p(t) = V_p \cos(\omega_p(t)t) \quad (4.3)$$

La señal modulada FM resultante, $s(t)$, se obtiene al variar la frecuencia angular de la señal portadora en función de la señal moduladora. La variación de frecuencia está determinada por el índice de modulación, β .

La expresión matemática de la señal modulada FM puede ser escrita como:

$$s(t) = A_c * \cos(\omega_c(t) * t + \beta * \sin(\omega_m(t) * t)) \quad (4.4)$$

En esta expresión, la frecuencia angular instantánea de la señal modulada FM, $\omega(t)$, viene dada por:

$$\omega(t) = \omega_c(t) + \beta * m(t) \quad (4.5)$$

Donde $m(t)$ es la señal moduladora y β es el índice de modulación.

En la fórmula, el término $\beta * \sin(\omega_m(t) * t)$ representa la desviación de frecuencia que varía sinusoidalmente en función de la señal moduladora. A medida que la amplitud de la señal moduladora varía, la frecuencia de la señal portadora se desvía proporcionalmente, creando la modulación en frecuencia. De esta forma se consigue expresar matemáticamente esa variación de la frecuencia de la señal portadora según la de la señal moduladora.

Cabe destacar que en esta descripción se utiliza una aproximación lineal, conocida como modulación de frecuencia en banda estrecha (NBFM - Narrowband Frequency Modulation). En casos más generales, la relación entre la frecuencia angular y la señal moduladora puede ser más compleja y requerir una representación matemática más precisa, como la serie de Taylor o una expansión de Bessel.

A continuación se muestra un diagrama de bloques de un modulador FM de banda estrecha 4.12 en el que se presenta un esquema gráfico de la expresión matemática 4.4 de la modulación FM.

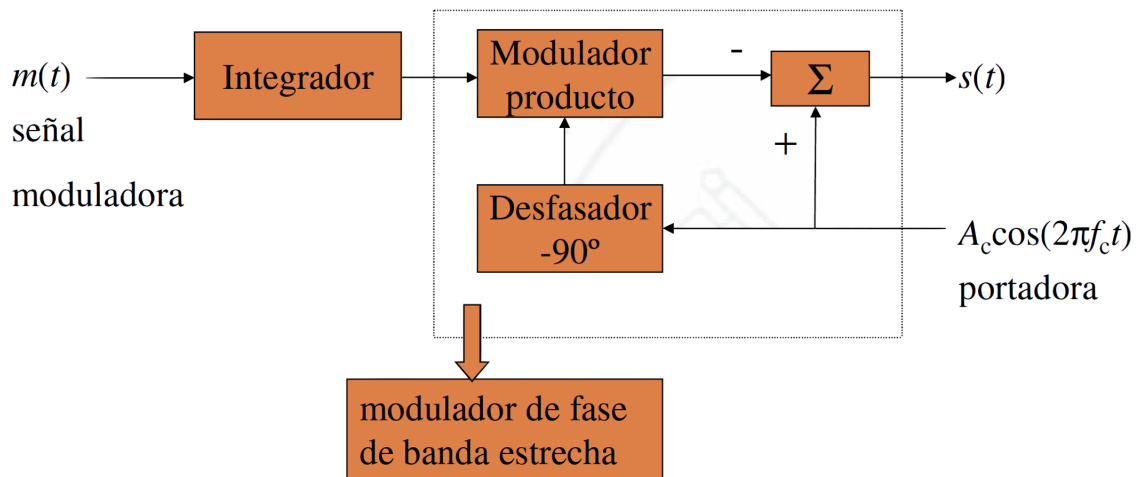


Figura 4.12: Diagrama de bloques FM

En la figura 4.13 se muestra gráficamente la forma de la señal modulada en función de la señal moduladora y de la portadora.

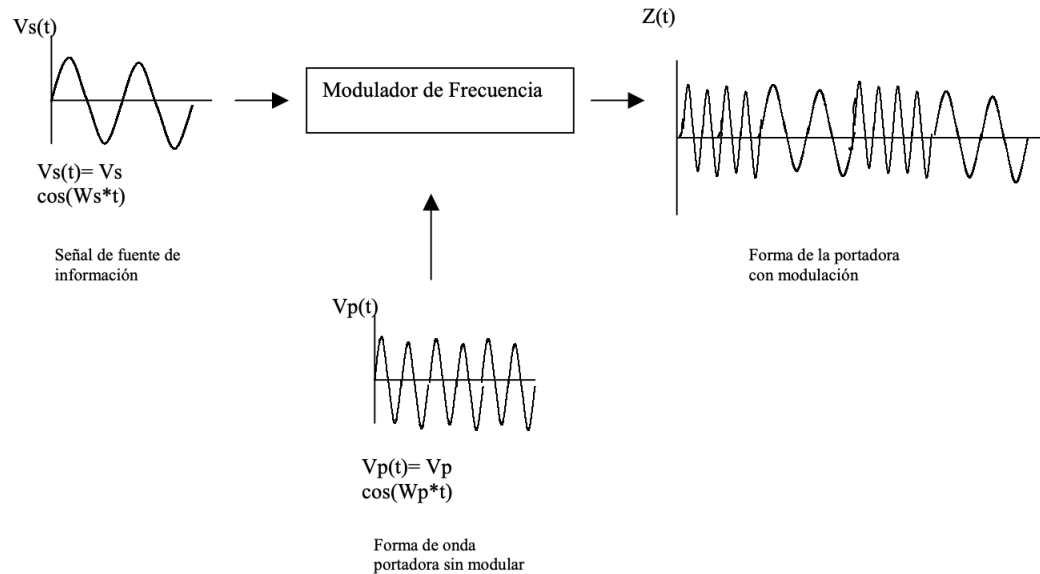


Figura 4.13: Diagrama de bloques FM

Como se ha explicado, en la modulación FM, la frecuencia de la onda portadora se modifica proporcionalmente a la amplitud de la señal moduladora. Esto significa que cuanto mayor sea la amplitud de la señal moduladora en un instante dado, mayor será la desviación de la frecuencia de la onda portadora en ese instante.

Cuando se realiza la modulación FM, el resultado es la aparición de bandas laterales en el espectro alrededor de la frecuencia de la onda portadora. Estas bandas laterales son múltiplos enteros de la frecuencia de la señal moduladora y se colocan simétricamente a frecuencias $f_m, 2f_m, 3f_m, \dots$ tanto por encima como por debajo de la frecuencia de la portadora a diferencia de la modulación en amplitud, donde sólo se originan 2 bandas laterales.

La presencia de un número infinito de bandas laterales se debe al hecho

de que la señal moduladora puede tener componentes de frecuencia en un rango infinito de valores. Cualquier señal moduladora, como la música o la voz humana, puede tener componentes de frecuencia en un amplio espectro.

4.4.2. Simulación de la modulación en frecuencia en GNU Radio

Conocidos los fundamentos teóricos de la modulación en frecuencia se procede a simularla utilizando GNU Radio para su mejor comprensión.

A continuación se muestra el esquema utilizado para la simulación 4.14.

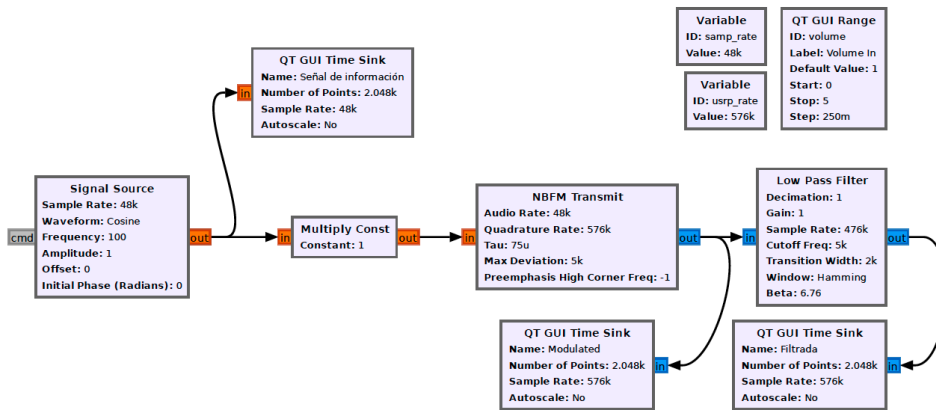


Figura 4.14: Esquema de modulación FM en GNU Radio

Partiendo de una señal moduladora generada, en este caso se ha escogido un coseno de frecuencia 100 Hz, se coloca un multiplicador para poder variar la ganancia de entrada según se desee.

Una vez amplificada la señal, se utiliza un bloque existente en GNU Radio, *NBFM Transmit*, gracias al cuál se consiguen generar señales moduladas en banda base. A continuación es necesario colocar un filtro paso bajo, *Low Pass Filter*, que elimina las frecuencias no deseadas que pueden aparecer durante la modulación FM, como las bandas laterales fuera del rango deseado, ayudando además a limitar la señal a un ancho de banda estrecho.

En la figura 4.15 se muestra la salida del modulador, previamente filtrada. Se puede observar cómo se ha transformado la señal inicial de información, situada en la parte superior, en otra modulada en la que se observan cambios en la frecuencia. Es ahí, en dichos cambios, donde queda recogida la información que gracias al proceso de demodulación, se podrá recuperar.

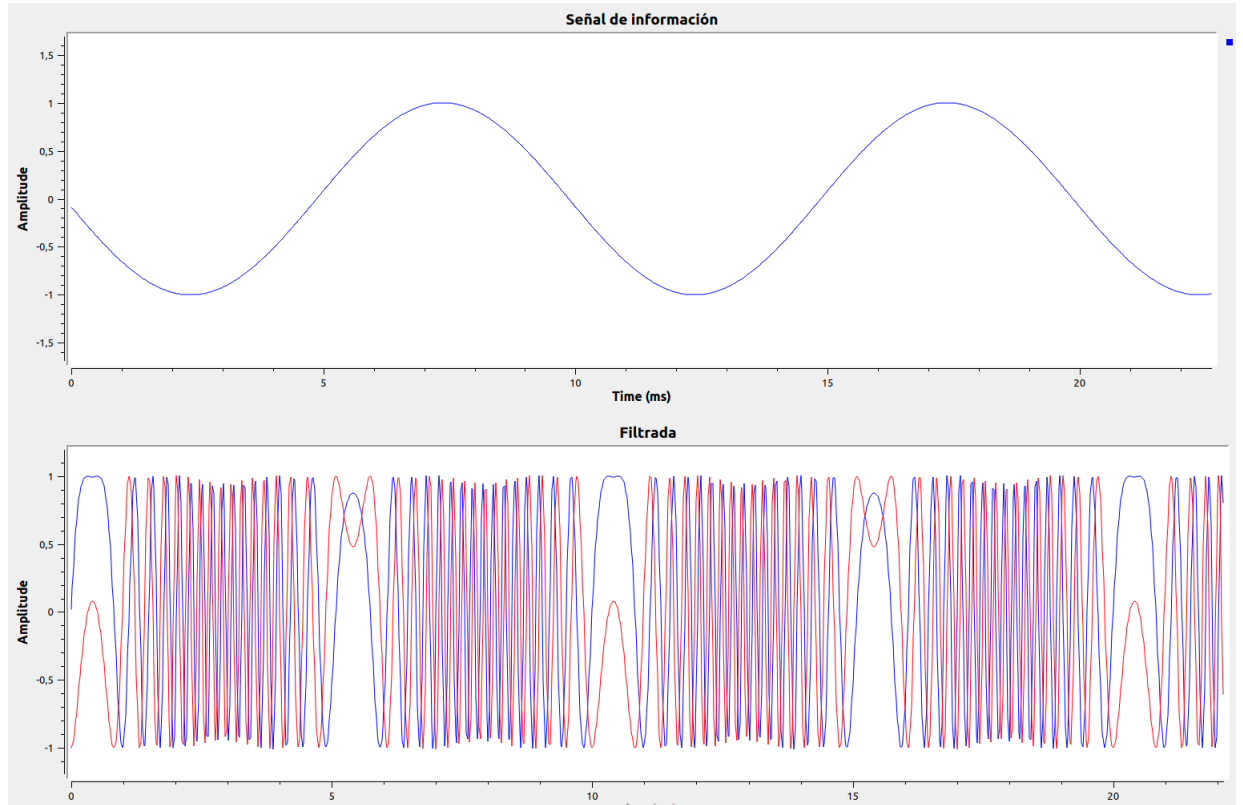


Figura 4.15: Señal de información y señal modulada FM

4.4.3. Fundamentos de la demodulación en frecuencia

La demodulación en frecuencia es el proceso que permite recuperar la señal moduladora de la señal FM; de esta forma la salida del demodulador FM será una señal con una amplitud instantánea proporcional a la frecuencia instantánea de la señal de entrada.

Existen dos esquemas básicos de demodulación FM:

- Discriminador de frecuencia: Idealmente es un derivador, un circuito pendiente, seguido de un detector de envolvente. El circuito pendiente se caracteriza por una función de transferencia que es imaginaria pura (simetría impar) y varía de forma lineal dentro del intervalo de frecuencias dado.
- Bucle enganchado en fase (PLL): Un circuito PLL es un sistema de control automático de bucle cerrado, que tiene la capacidad de rastrear la fase de la señal de entrada, como un filtro de seguimiento de banda estrecha; de esta forma se puede extraer la señal a pesar del ruido que tiene incorporado [11].

4.4.4. Simulación de la demodulación en frecuencia en GNU Radio

Tras haber explicado los esquemas básicos de la demodulación en frecuencia se procede a simularla en GNU radio para mejorar su comprensión.

A continuación se muestra el esquema empleado para la simulación de la demodulación 4.16. En él se utiliza una señal de entrada modulada en frecuencia. El primer bloque empleado es el *Simple Squelch* con un umbral de -50 decibelios, dicho bloque se utiliza en simulación para ajustar el nivel donde no se tiene audio, evitando así interferencias en ausencia de señal. El squelch es una función común en los receptores de radio que se utiliza para silenciar el audio de salida o suprimir la salida de datos cuando la señal de entrada cae por debajo de un cierto nivel. Esto evita que se escuche ruido estático o información no deseada cuándo no hay una señal válida presente [12].

El bloque destinado a demodular la señal es el *Narrow Band Receive*. Cabe destacar que los parámetros, *Audio Rate* y *Quadrature Rate* deben ser múltiplos entre sí.

A la salida del demodulador se coloca un bloque multiplicador con el fin de aumentar la ganancia de la señal. Este bloque tendrá un parámetro variable para poder modificarla como se desee.

Finalmente se podrá escuchar la señal de audio demodulada utilizando un bloque *Audio Sink*.

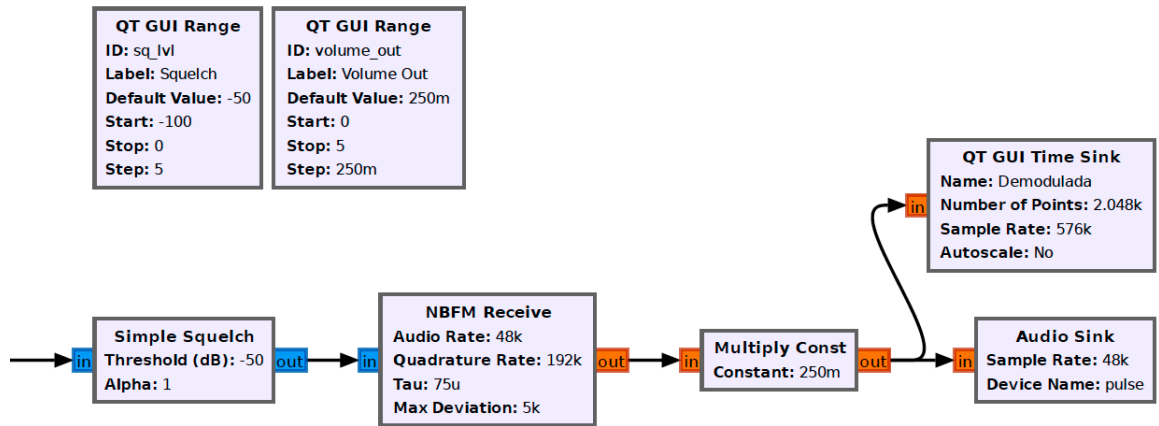


Figura 4.16: Esquema de demodulación FM en GNU Radio

Posteriormente se muestra la señal demodulada. En ella se puede apreciar como partiendo de una señal modulada se logra recuperar una señal de información similar a la modulada en el apartado 4.4.2. En este caso se ha extraído la información registrada en los cambios de frecuencia de la portadora para conformar la señal demodulada.

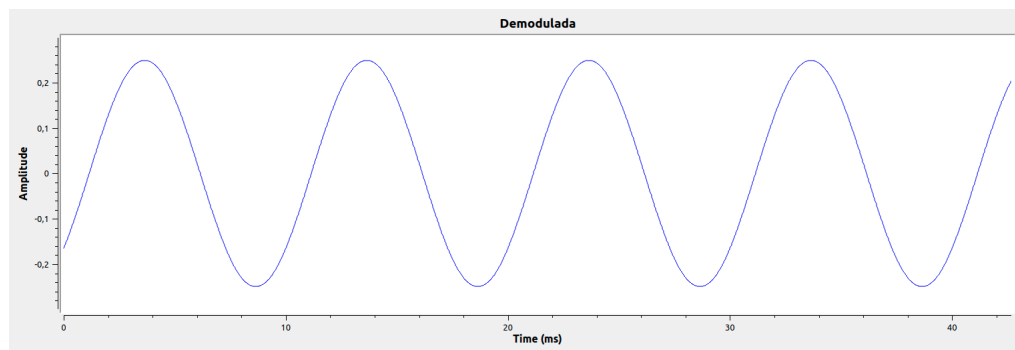


Figura 4.17: Señal modulada y señal demodulada FM

Capítulo 5

Modulaciones digitales

5.1. Introducción

Las modulaciones digitales se caracterizan por una fuente de información digital, lo que las distingue de las modulaciones analógicas en las cuales la fuente de información es analógica.

La señal de información mencionada será ahora un flujo de bits. Dicho flujo de bits se dividirá en grupos y dependiendo del tamaño de estos se obtendrán los M niveles de la modulación. Los niveles se calculan gracias a la expresión $M = 2^k$ en la que k representa el número de bits agrupados. Por ello, las señales moduladas se recogen en un conjunto finito de símbolos que las componen.

Así mismo, los símbolos forman conjuntos que representan un valor discreto, los cuales se denominan palabras o códigos. Existen códigos binarios, cuaternarios, etc. El código binario es el más sencillo, en él solo hay dos símbolos posibles.

A continuación se muestra el esquema básico de un sistema PCM, del inglés *Pulse Code Modulation*. Este procedimiento de modulación se utiliza para transformar una señal analógica en una secuencia de bits y viceversa [13].

Siguiendo el esquema 5.1 se observa que la señal analógica situada en la entrada, $m(t)$, es muestreada teniendo en cuenta el teorema del muestreo

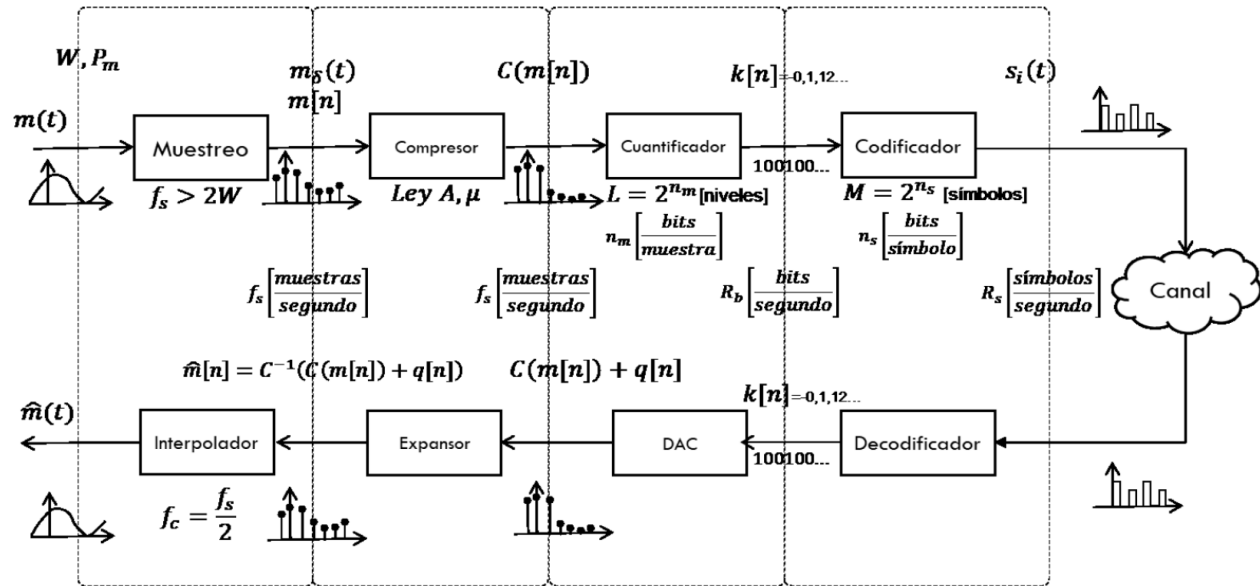


Figura 5.1: Esquema básico de un sistema PCM

de Nyquist. De ello se obtendrían n muestras de la señal. La frecuencia de muestreo utilizada se obtendrá dividiendo el número de muestras tomadas entre el tiempo empleado; $f_s = \text{muestras/segundo}$.

Obtenida la señal discreta $m[n]$ pasará por un compansor que generalmente empleará algoritmos de cuantificación basados en las leyes A o μ con el fin de fijar los niveles de cuantificación utilizados posteriormente, obteniéndose de esta manera la señal $C(m[n])$. Dicha señal entrará en el cuantificador que llevará a cabo un proceso mediante el cual se asigna un determinado valor discreto a cada uno de los niveles de tensión obtenido a la salida del compansor.

La señal cuantificada se codificará asignando a cada nivel de cuantificación un código binario distinto para su posterior modulación y transmisión.

El proceso inverso servirá para obtener una señal analógica partiendo de una secuencia de bits; para ello, la señal recibida y previamente demodulada

entrará en el decodificador que se encargará de asignar códigos binarios a las señales recibidas para que posteriormente, en el conversor digital-analógico se le atribuya a cada código binario un nivel de tensión.

En el expansor tendrá lugar el proceso inverso llevado a cabo en el compansor para transformar según los niveles de cuantificación, la señal que entra en él en una secuencia de *deltas* con sus correspondientes amplitudes. Esta señal será equivalente a la muestreada, por lo que, colocando un interpolador, se podrá recuperar una señal muy similar a la de entrada en el sistema, $m(t)$.

Existe una serie de conceptos importantes que es necesario conocer para entender lo que se planteará posteriormente.

Como se ha mencionado previamente, en las modulaciones digitales se transmiten pulsos, debido a ello surgen problemas de distorsión como la interferencia entre símbolos. Surge a consecuencia de un filtrado inadecuado, por lo cual, los pulsos se deforman y dispersan invadiendo a los adyacentes. En la figura 5.2 se muestra de forma gráfica la interferencia entre símbolos.

El diagrama de ojo es un método representado para el análisis del comportamiento de los enlaces de transmisión. Su representación permite analizar las formas de onda de los pulsos que se propagan en un enlace de comunicaciones con el objetivo de poder observar sus formas, niveles de ruido, desfases... Gracias a ellos se podrá apreciar la interferencia entre símbolos, el ruido o interferencias y los errores de sincronismo en el receptor. En la figura 5.3 se muestra un ejemplo de cómo se puede observar la distorsión producida en las señales examinando los diagramas de ojo resultantes [14].

De igual modo, resulta de interés comprender el concepto de decisor que es el encargado de asignar a cada valor de la señal a demodular un símbolo diferente en función del umbral propuesto de modo que se consiga minimizar la probabilidad media de error de símbolo. Dicha probabilidad resulta importante para cuantificar las prestaciones del sistema de comunicaciones.

Otra idea relevante es la de constelación, la representación de los símbolos en un espacio de L dimensiones. Esta interpretación se debe a que el conjunto de señales forman un espacio vectorial por lo que se puede encontrar una base ortonormal que represente dicho espacio. En la constelación

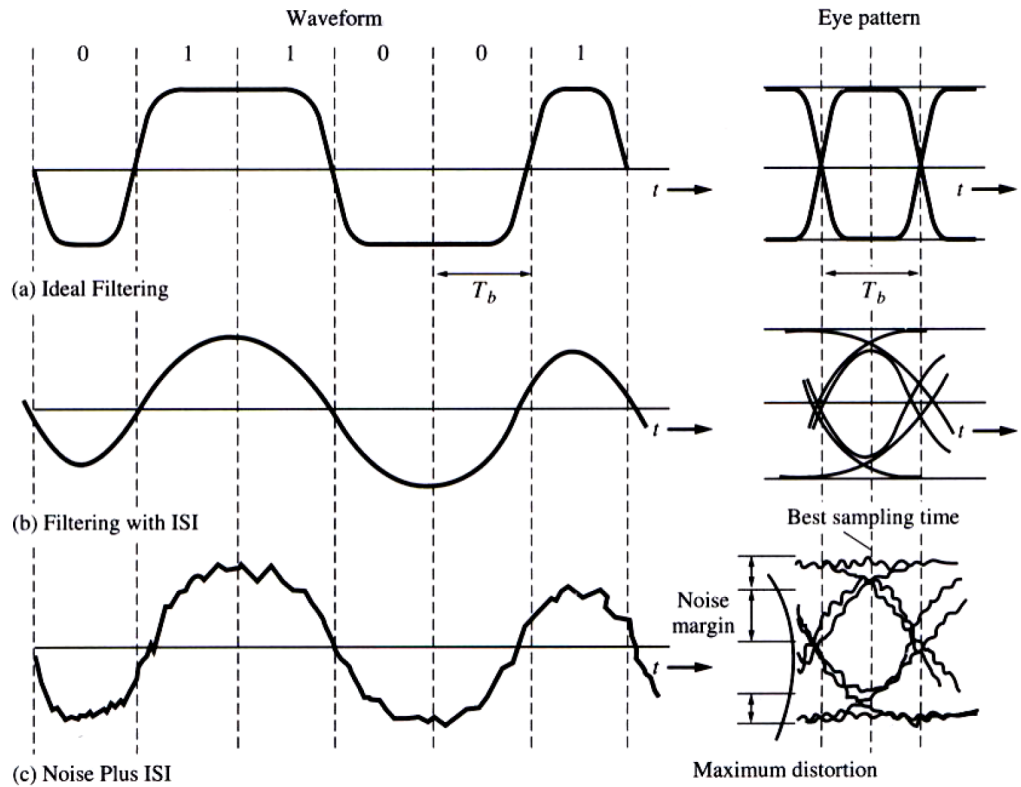


Figura 5.2: Interferencia entre símbolos

se representan en el plano complejo los estados de símbolo en términos de amplitud y fase, es decir, la relación de amplitud y de fase de una portadora modulada digitalmente. Cuanto más distantes se encuentren los puntos del origen de coordenadas, mayor será la energía necesaria para transmitirlos. Estos diagramas son especialmente útiles para la detección del tipo de interferencia y la distorsión en una señal.

Dentro de las modulaciones digitales se distinguen dos tipos en función de la forma de transmisión de los pulsos:

- Modulaciones en banda base: La señal modulada $s(t)$ se transmite en torno a frecuencia cero.
- Modulaciones paso banda: La señal modulada $s(t)$ se transmite en torno

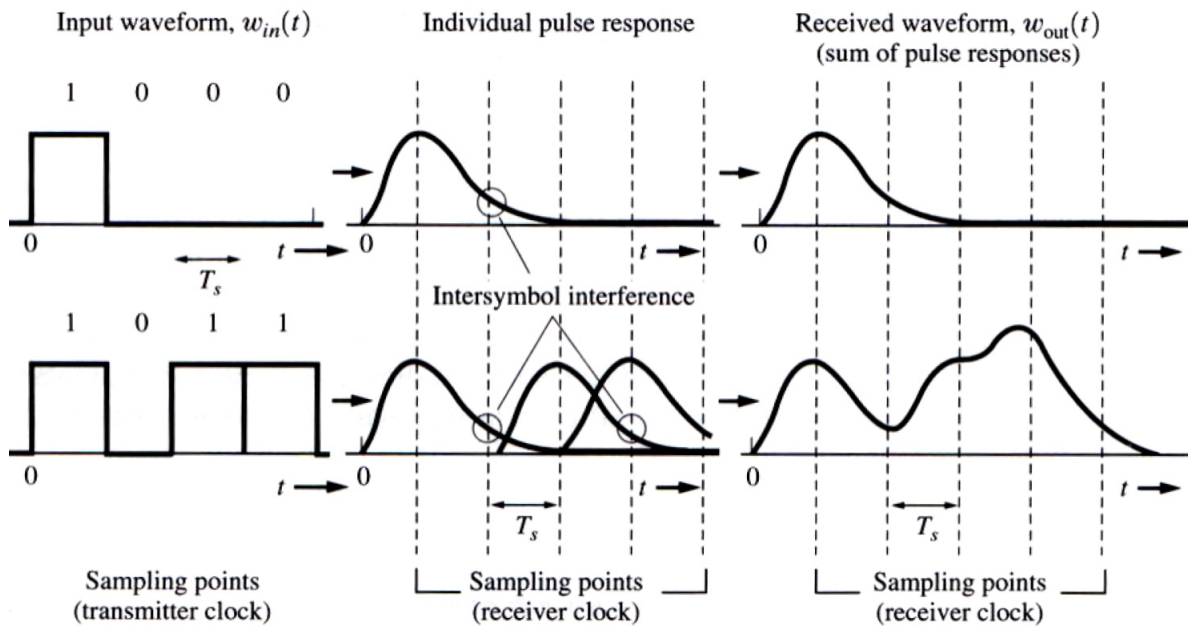


Figura 5.3: Interferencia entre símbolos

a una frecuencia no nula, generalmente mucho mayor que el ancho de banda.

5.2. Modulación digital de amplitud: ASK

5.2.1. Fundamentos de la modulación digital de amplitud

La modulación digital de amplitud, también conocida como modulación de amplitud por desplazamiento, es un tipo de modulación paso banda. En ella, los datos digitales se representan como variaciones de amplitud en la portadora. De esta forma, la amplitud de la señal portadora analógica varía conforme a la corriente de bit, manteniendo la frecuencia y la fase constantes dando lugar así a la señal modulada.

El nivel de amplitud es utilizado para representar los símbolos. En el caso de que sea una modulación 2-ASK, únicamente se representarán dos símbolos,

el 0 y el 1 y solamente se tendrán dos amplitudes posibles. Para la 4-ASK se representarán cuatro símbolos: 00, 01, 10 y 11; por tanto habrá cuatro amplitudes diferentes. En conclusión, para el caso M-PSK se representarán M símbolos y se tendrán M amplitudes diferentes.

En la figura 5.4 se observa una representación gráfica de la modulación 2-ASK. En ella se advierte que cuando el símbolo es uno, se multiplica la portadora por una amplitud A, mientras que cuando el símbolo es cero, se multiplica por una amplitud nula. Es recomendable que el periodo de la señal portadora sea múltiplo del periodo de símbolo, o en este caso de bit. En la situación planteada, se distingue fácilmente que el periodo es tres veces el de la portadora.

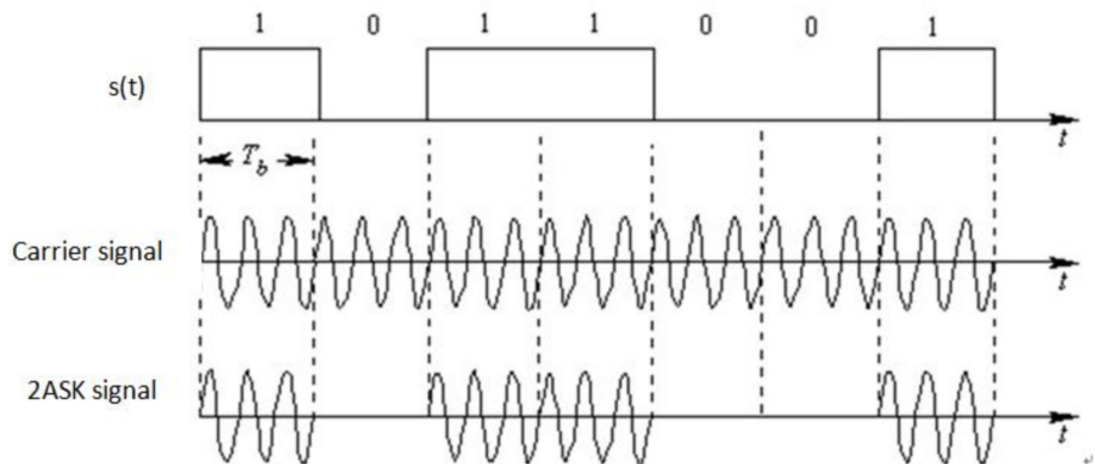


Figura 5.4: Modulación ASK

La modulación 2-ASK es conocida también como *ASK Binaria*, *BASK*, y si tiene un valor de amplitud 0, recibe el nombre *On-Off Keying*, *OOK*.

La modulación ASK puede entenderse también como un desplazamiento en frecuencia de una modulación PAM, es decir, multiplicarla por una portadora. Por lo tanto, la señal modulada ASK puede expresarse como en la ecuación 5.1

$$S_m(t) = A_m g(t) \cos(\omega_c t), m = 1, 2, \dots, M, 0 \leq t \leq T \quad (5.1)$$

En la ecuación 5.1, $g(t)$ es un pulso de forma arbitraria y periodo el del símbolo. Su frecuencia, como se ha explicado anteriormente, deberá ser preferiblemente menor y múltiplo de la de la portadora.

Las amplitudes A_m de una modulación ASK se toman, generalmente, como en una modulación por amplitud de pulsos asimétrica, es decir, comienzan en cero y se separan uniformemente.

5.2.2. Simulación de la modulación digital de amplitud en GNU Radio

Al igual que en secciones anteriores, tras conocer los fundamentos teóricos se procede a simular la modulación B-ASK en GNU Radio para una mejor aproximación al conocimiento del software así como para su mejor comprensión. El esquema utilizado para ello será el recogido en la figura 5.5.

En este caso se utilizará como moduladora una señal cuadrada de frecuencia 100 Hercios y como portadora un seno de frecuencia 1 kilohercio. Se observa que, como se ha explicado en la sección anterior, la frecuencia de la portadora se fija a un valor 10 veces superior que la de la señal de información.

La simulación de esta modulación en GNU Radio resulta muy simple ya que bastará con multiplicar la señal de información por la portadora para obtener al señal modulada.

Se pueden examinar las muestras de la señal de entrada, la portadora y la modulada en la figura 5.6.

Resulta fácil percibir cómo se ha registrado la información contenida en la señal moduladora 5.2.2 en la señal portadora 5.2.2, obteniéndose como resultado la señal modulada 5.2.2 en la que se observan variaciones de amplitud entre cero y uno; el periodo de estas variaciones coincide con el flujo de amplitudes mostrado en la señal cuadrada.

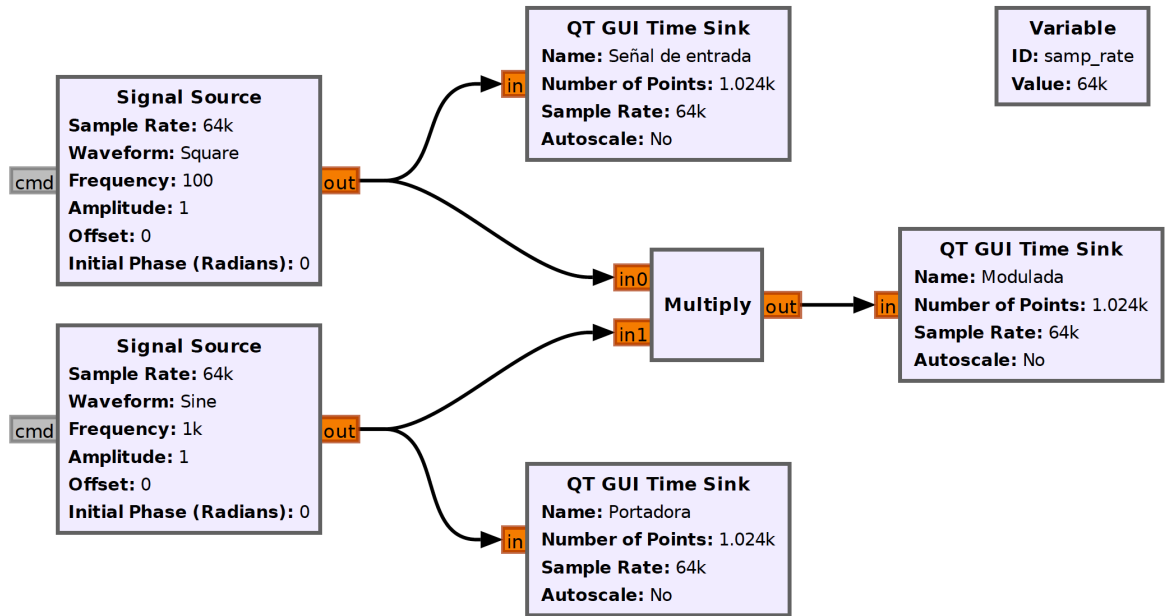


Figura 5.5: Esquema de modulación ASK en GNU Radio

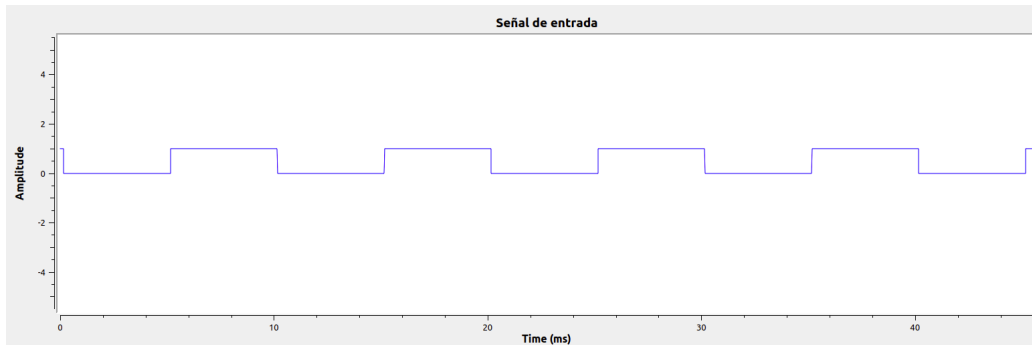
5.2.3. Fundamentos de la demodulación digital de amplitud

Tras haber transformado una secuencia binaria en una señal modulada se hace necesario explicar el proceso inverso en el cual se partirá de la señal modulada para conseguir finalmente recuperar la información binaria.

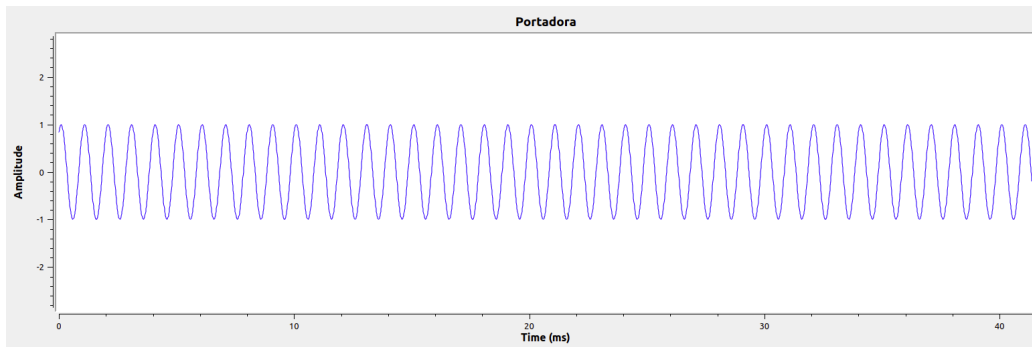
Según lo explicado hasta el momento, se puede extraer la conclusión de que es posible entender la modulación ASK como una variante de la modulación AM por lo que el proceso de demodulación será similar también al de la demodulación en amplitud.

Existe dos métodos diferentes para recuperar la señal original: la de demodulación por detección síncrona y la demodulación por detección de envolvente. En las siguientes líneas se profundizará en este último ya que ofrece mejores prestaciones que la primera.

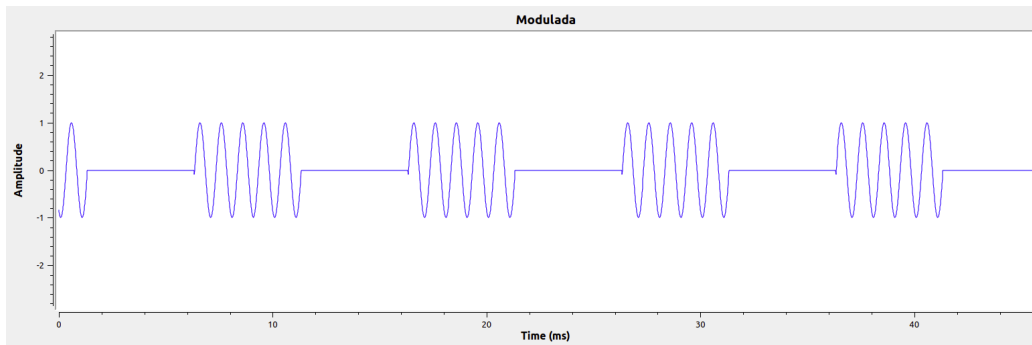
El receptor basado en detección de envolvente se encarga de eliminar la por-



(a) Señal moduladora



(b) Señal portadora



(c) Señal modulada ASK

Figura 5.6: Señales obtenidas mediante simulación en GNU Radio.

tadora de alta frecuencia a la vez que bloquea la parte negativa de la señal modulada. Como consecuencia, la salida del detector será la envolvente po-

sitiva sumada a una componente de continua y en ocasiones componentes diente de sierra de alta frecuencia que podrán ser eliminadas utilizando un filtro pasobajo.

Un esquema representativo de este tipo de demodulación puede ser el recogido en la figura 5.7. Es conveniente filtrar previamente la señal recibida utilizando un filtro paso banda para eliminar el ruido y las posibles interferencias adyacentes.

El último bloque del esquema hace referencia al decisor, para el cual λ sería el umbral.

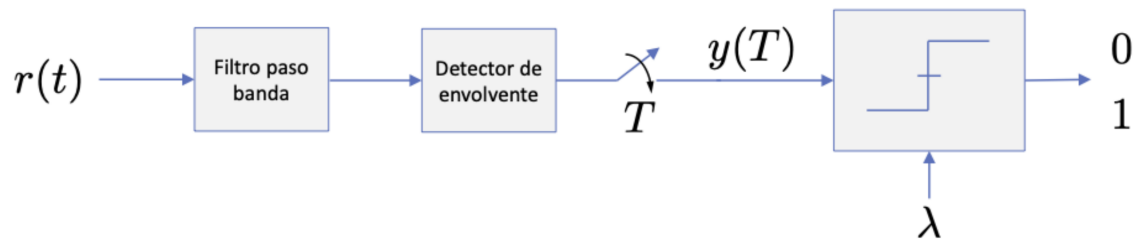


Figura 5.7: Esquema de demodulación ASK

Como se ha descrito previamente, las señales se pueden representar geométricamente en una constelación. En este caso se distribuyen uniformemente a lo largo del eje que forma la base. Posteriormente se presenta un ejemplo de una constelación para una modulación 4-ASK 5.8 en la que se observa lo explicado anteriormente.

En el eje X se representan las amplitudes; cuanto mayor es la separación del origen, mayor es la energía de la señal. En el eje Y se representarían los cambios de fase de la señal, en este caso inexistentes por lo que la componente en cuadratura es nula y sólo existe componente en fase.

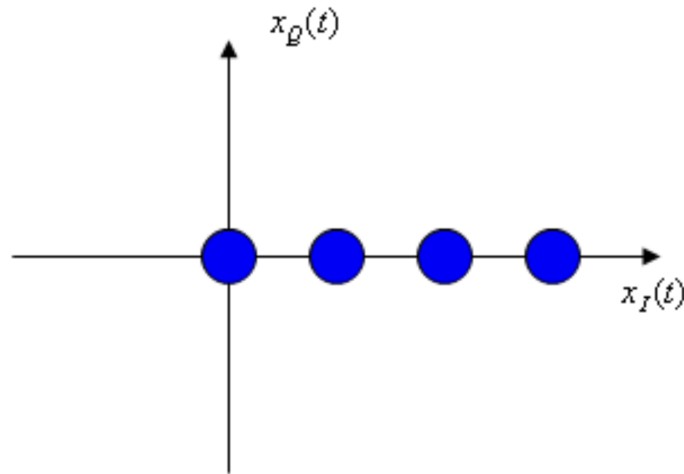


Figura 5.8: Constelación 4-ASK

5.2.4. Simulación de la demodulación digital de amplitud en GNU Radio

A continuación se procede a simular la recuperación de la señal original modulada en la sección 5.3.2 empleando el software GNU Radio. Para ello se utiliza el esquema recogido en la figura 5.9.

En él se recoge la señal modulada anteriormente 5.2.4 y se transforma al conjunto complejo; su parte imaginaria será cero ya que no variará su fase, únicamente lo hará su amplitud. De esta forma se conseguirá mostrar la constelación de entrada al demodulador que se observa en la figura 5.2.4. En ella se aprecia como los símbolos se desplazan entre -1 y 1. Esto se debe a que las amplitudes de la señal modulada se sitúan entre estos valores.

A continuación, gracias al bloque *Threshold* que actúa como decisor, se consigue asignar a cada valor de la señal un símbolo, o como en este caso, elegir la amplitud que tenía la señal antes de ser modulada.

Por último se observa la señal demodulada 5.2.4 y la constelación asociada a ella 5.2.4. Se contempla como se ha conseguido recuperar la señal original a través de este proceso de demodulación, además de haber obtenido una constelación cuyos puntos están perfectamente definidos en 0 y 1.

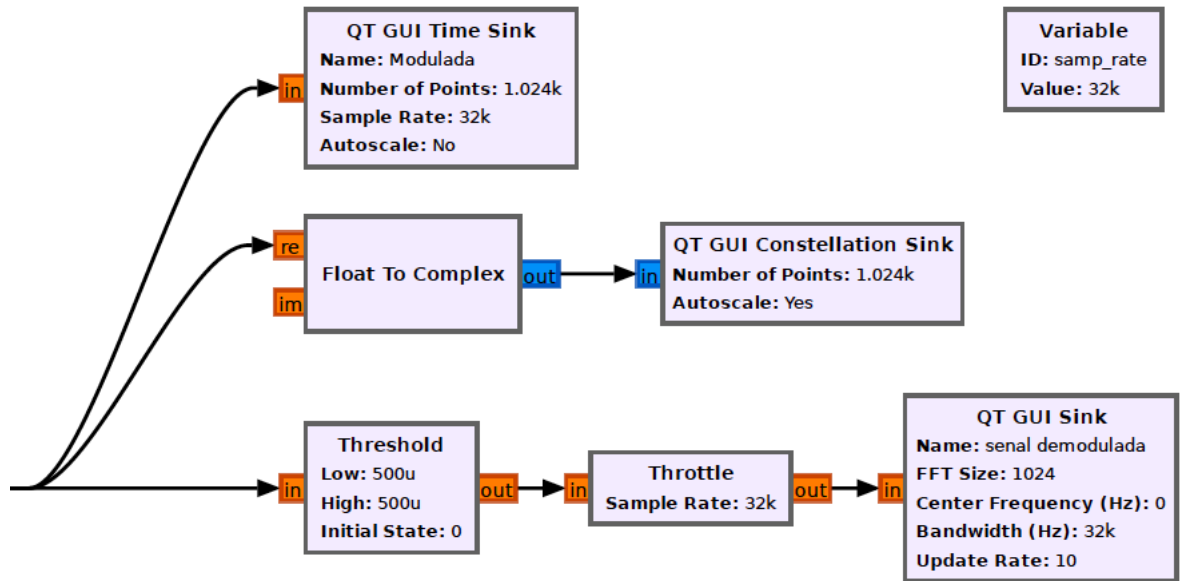


Figura 5.9: Esquema de demodulación ASK en GNU Radio

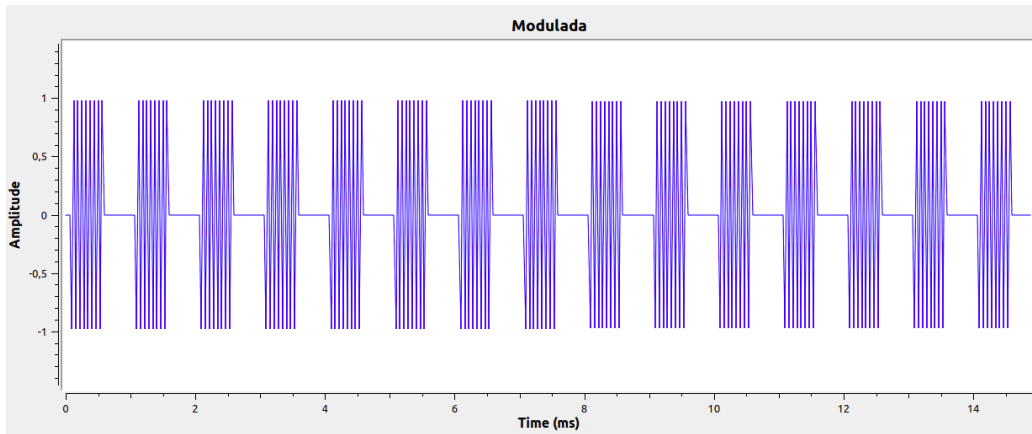
5.3. Modulación digital de frecuencia: FSK

5.3.1. Fundamentos de la modulación digital de frecuencia

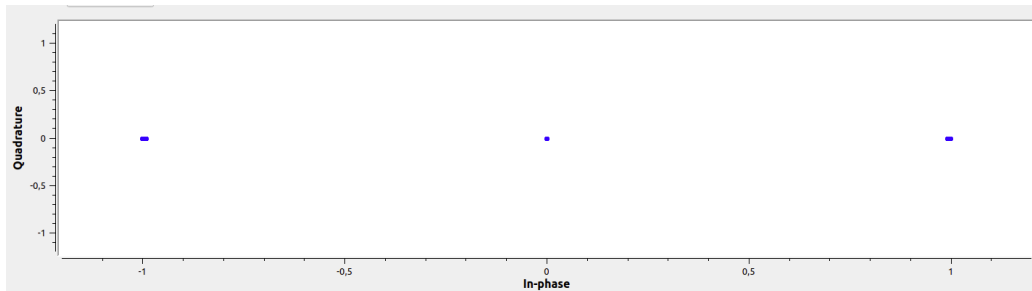
La modulación digital por desplazamiento de frecuencia es una técnica de modulación para la transmisión digital de información basada en la utilización de diferentes frecuencias para cada símbolo. Esta modulación se puede entender también como una modulación en frecuencia sobre una modulación PAM.

Para modular la señal se utilizan portadoras de frecuencias distintas para cada símbolo. De esta forma, la frecuencia de la señal modulada varía conforme al símbolo, manteniendo la amplitud y la energía de la señal constantes dando lugar así a la señal modulada.

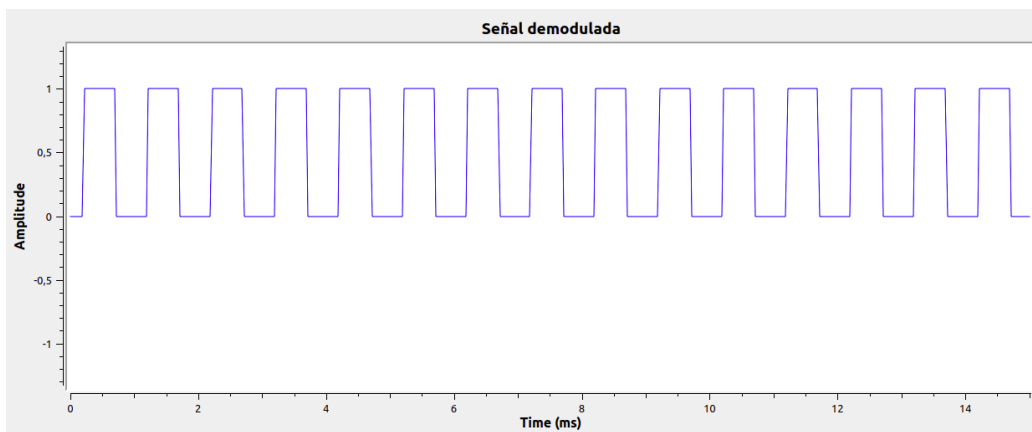
La señal modulada FSK puede expresarse como la ecuación 5.2.



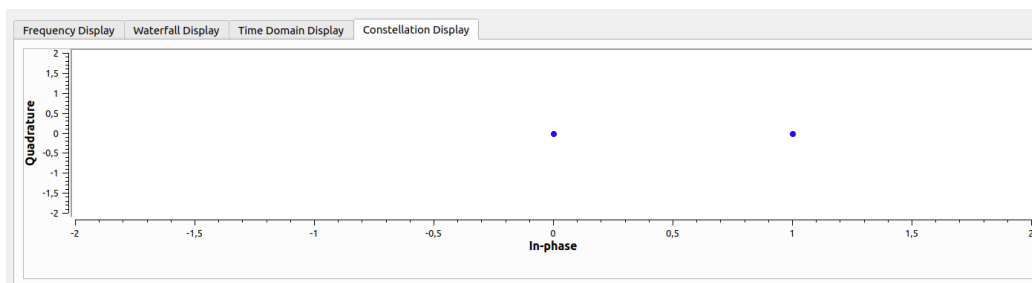
(a) Señal modulada



(b) Constelación en la entrada del demodulador



(c) Señal demodulada ASK



(d) Constelación en la salida del demodulador

Figura 5.10: Señales obtenidas mediante simulación en GNU Radio.

$$S_m(t) = Ag(t) \cos(2\pi f_m t), f_m = f_c + \frac{1}{2}(2m - M - 1) \Delta_f \quad m = 1, 2, \dots, M \quad (5.2)$$

En ella, f_c hace referencia a la frecuencia de la portadora $g(t)$ es un pulso de periodo el del símbolo y de forma arbitraria. Δ_f es la separación de frecuencias entre símbolos consecutivos, tomada con el objetivo de que las señales sean ortogonales entre sí. Para ello serán necesarios tantos vectores para definir la base del subespacio, como señales tenga el alfabeto. De esta forma el número de dimensiones del espacio vectorial coincidirá con el número de símbolos utilizados en la modulación.

El esquema 5.11 muestra un esquema sencillo de implementación de un modulador B-FSK o 2-FSK que ayudará a la comprensión de su funcionamiento [15].

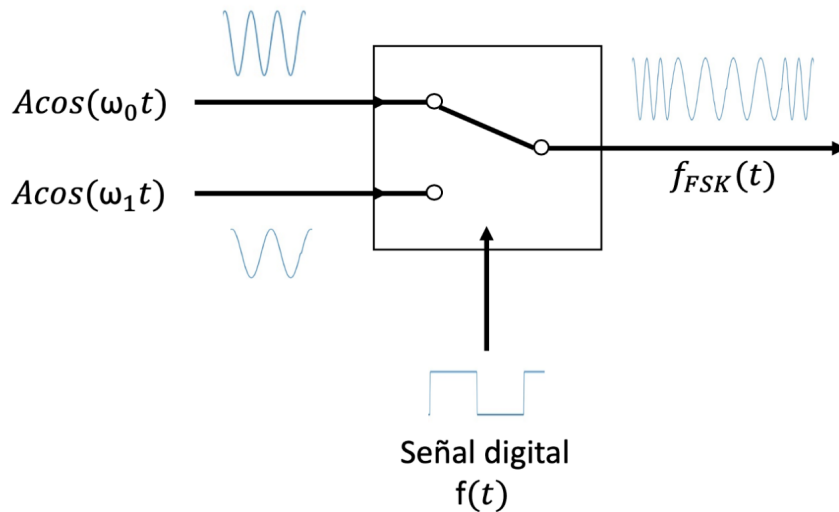


Figura 5.11: Esquema de demodulación FSK

En el ejemplo se parte de una señal binaria de información con niveles de 0 y 1 voltio; se utilizan dos señales portadoras de alta frecuencia cuyas frecuencias son f_0 y f_1 con amplitud A . Utilizando un multiplexor analógico cuya señal de selección corresponde a la señal binaria de información; así

cuando la señal de selección $f(t)$ tiene una amplitud de 0V, la señal que se encontrará a la salida será $A \cos(\omega_0 t)$ mientras que si la amplitud es de 1V la salida será $A \cos(\omega_1 t)$. Gracias a ello, la señal en la salida a lo largo del tiempo se corresponderá con $f_{FSK}(t)$, la señal modulada FSK. En la figura 5.12 se puede observar los efectos de la modulación con respecto a la señal de información.

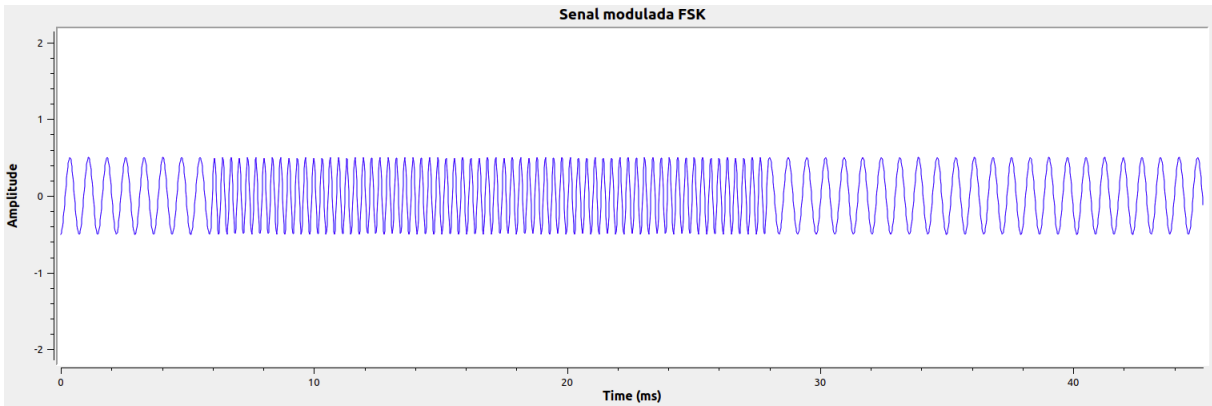


Figura 5.12: Señal modulada FSK

5.3.2. Simulación de la modulación digital de frecuencia en GNU Radio

Tras haber explicado los fundamentos teóricos de la modulación digital de frecuencia se procede a simularla en GNU radio para mejorar su comprensión.

En este caso se modulará una secuencia de bits por lo que en primer lugar será necesario transformar dicha secuencia en una señal lista para ser modulada; para ello se utilizará el esquema mostrado en la figura 5.13.

El bloque *Random Source* genera un vector de 1.024k muestras de números aleatorios comprendidos entre el 0 y el 256 en binario. Posteriormente el bloque *Unpack K Bits* selecciona los K bits menos significativos de cada byte y los expande en K bytes de 0 uno. Posteriormente en el bloque *Repeat* se indica el número de veces que se desea que se repita la entrada, actuando como el factor de interpolación. Es entonces cuando se transforma la señal en complejos para su visualización 5.14.

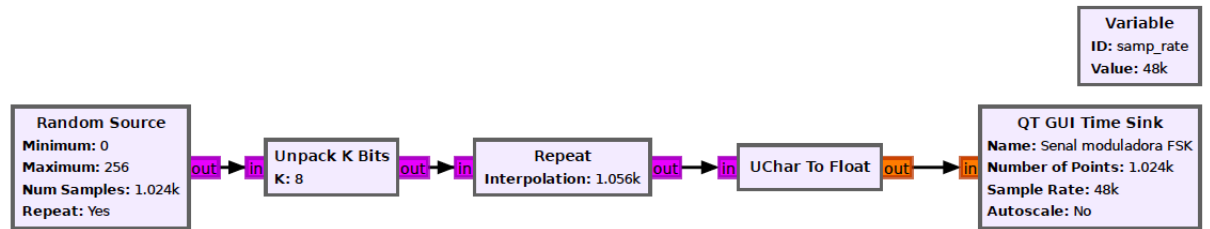


Figura 5.13: Diagrama utilizado para generar la señal moduladora en GNU Radio

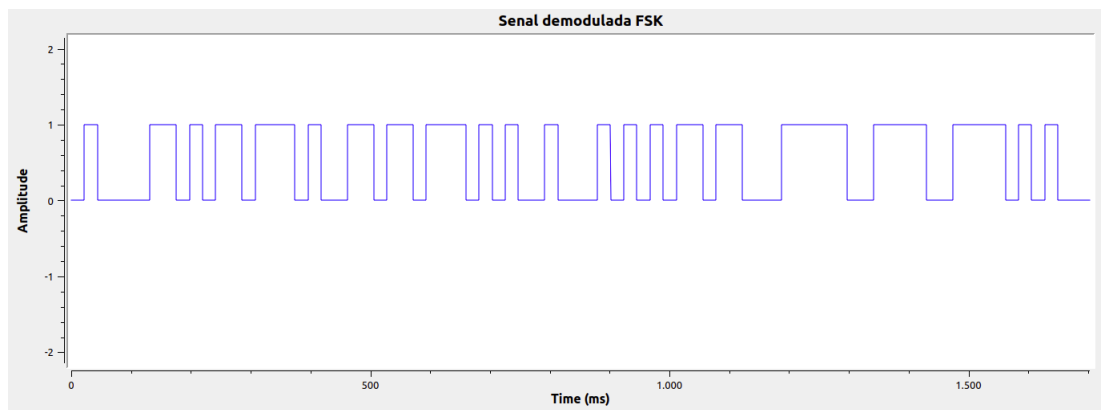


Figura 5.14: Señal moduladora

Teniendo la señal de información deseada se lleva a cabo su modulación. El esquema 5.15 corresponde al diagrama utilizado para modular la señal generada.

En él, la señal entrante es multiplicada para modificar su amplitud por la deseada. Posteriormente, con la intención de eliminar el offset de la señal, se utiliza el bloque *Add Const.*

El oscilador controlado por voltaje *VCO* genera una señal sinusoidal cuya frecuencia se basa en la amplitud de la entrada. De esta forma se obtendría finalmente la señal modulada que se observará en la imagen 5.16.

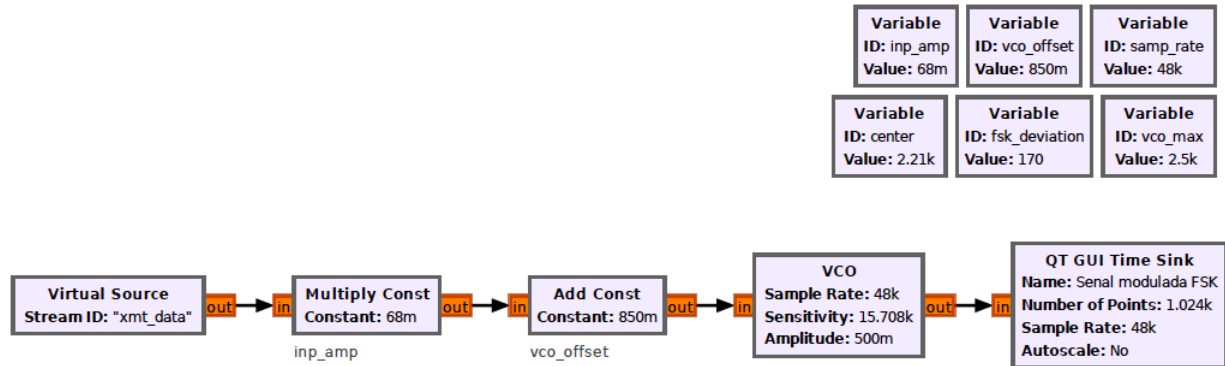


Figura 5.15: Esquema de modulación FSK en GNU Radio

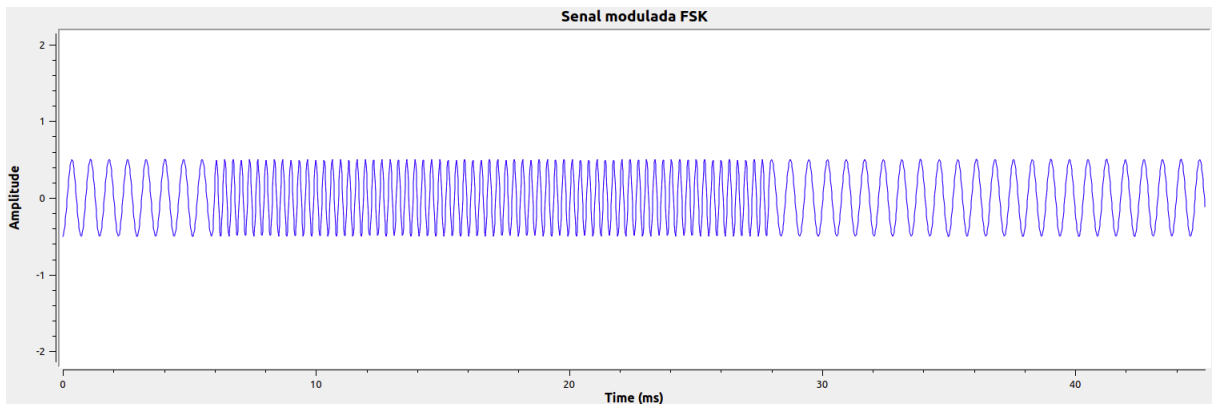


Figura 5.16: Señal modulada

En ella se observan los cambios de frecuencia relativos a los cambios de información en la señal. Para que estos cambios sean más abruptos con el fin de poderlos percibir a simple vista, es necesario aumentar la desviación en frecuencia; cuanto mayor sea esta, mayor será la diferencia entre las frecuencias utilizadas para la transmisión de cada símbolo.

5.3.3. Fundamentos de la demodulación digital de frecuencia

Habiendo modulado de señal se desea recuperar la información original mediante el proceso de demodulación. Para ello se pueden utilizar dos procedimientos diferentes; mediante un detector de filtros duales o utilizando un detector PLL.

La demodulación mediante un detector de filtros duales consta de un detector compuesto por filtros paso banda a la salida de un comparador. Estos filtros extraen los espectros de la señal centrados en las frecuencias utilizadas para cada símbolo en la modulación. A cada filtro le sigue un detector de envolvente semejantes al utilizado en la demodulación ASK. A la salida de estos se encuentra el comparador diferencial que consigue recuperar la señal de información [16].

El proceso explicado anteriormente puede observarse de forma gráfica en la figura 5.17.

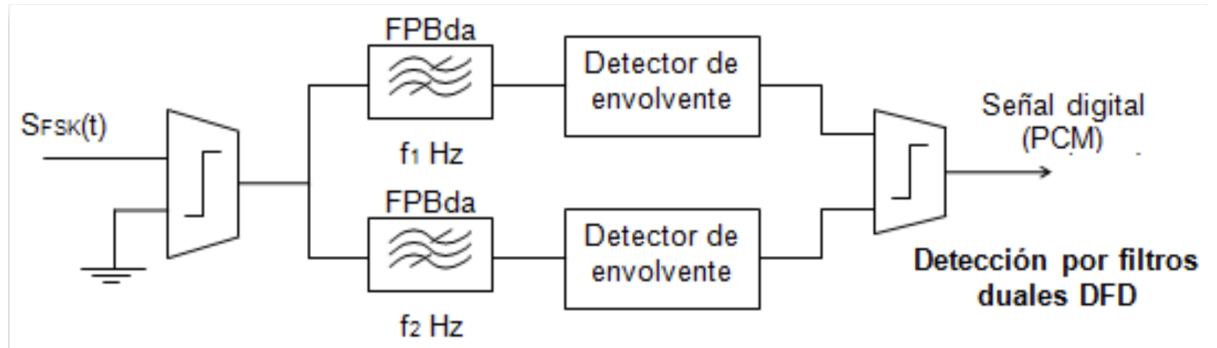


Figura 5.17: Esquema de la demodulación mediante detector de filtros duales

La demodulación utilizando una PLL, *Phase Locked Loop*, utiliza en primer lugar un filtro paso banda seguido de un comparador con el objetivo de conseguir una señal sin ruido y de amplitud fija a la entrada de la PLL.

El oscilador controlado por tensión, *VCO*, genera una señal de la misma frecuencia y fase que la señal de entrada. Posteriormente, en el multiplica-

Se obtiene una diferencia de fase entre la señal de entrada al PLL y la salida del VCO; gracias a un filtro paso bajo se calcula la señal de error que producirá una variación de la tensión. Es decir, cada vez que en la entrada se produzca un cambio de frecuencia o de fase se producirá una variación en la tensión de control que anulará el error de fase de forma que la PLL siga la señal de entrada. Finalmente, gracias a un comparador logra recuperar la señal digital original.

La disposición del circuito detallado previamente se observa en el diagrama de la figura 5.18.

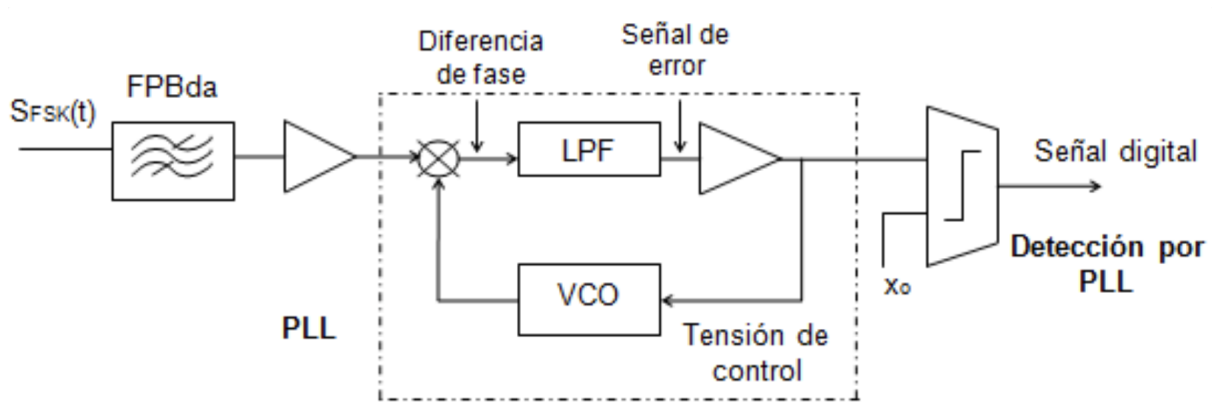


Figura 5.18: Esquema de la demodulación mediante PLL

El resultado de utilizar cualquiera de los dos procedimientos descritos anteriormente se corresponde con la señal modulada previamente.

5.3.4. Simulación de la demodulación digital de frecuencia en GNU Radio

Se pondrá en práctica lo explicado en la sección anterior simulando la demodulación en frecuencia en GNU Radio. Para ello se utilizará el diagrama de bloques de la figura 5.19.

En él se observa como la señal modulada pasa en primer lugar por el bloque *Frequency Xlating FIR Filter* que se encarga de aplicar una traslación de frecuencia a la señal además de realizar un submuestreo de la señal utilizando

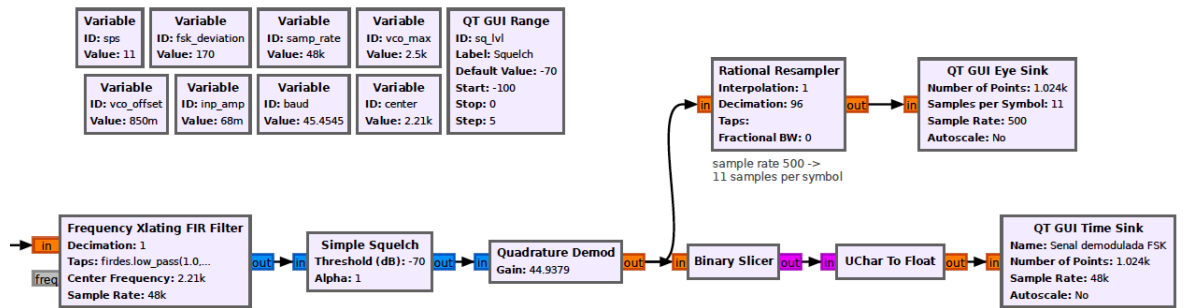


Figura 5.19: Esquema de demodulación FSK en GNU Radio

un filtro FIR diezmadador. Seguidamente se utiliza el bloque *Simple Squelch* que silencia la señal cuando su potencia es menor al umbral definido.

El *Quadrature Demod* lleva a cabo la demodulación FSK; en él se introduce una señal en banda base compleja para obtener a la salida la frecuencia de la señal en relación con la de muestreo y multiplicada por la ganancia para, posteriormente, calcular su argumento. De esta forma se consigue demodular la señal obteniéndose la señal mostrada en la figura 5.20.

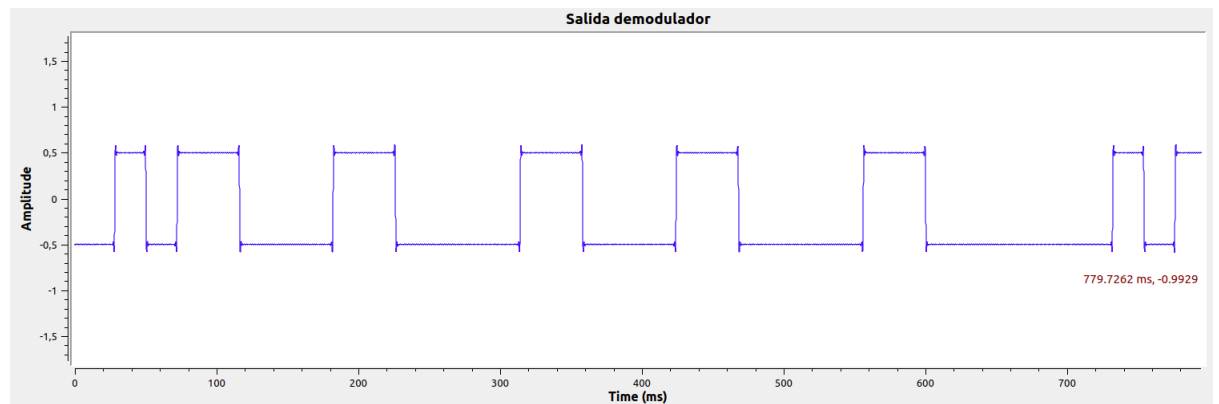


Figura 5.20: Señal obtenida a la salida del demodulador en GNU Radio

Como se observa, la señal no es exactamente la moduladora utilizada previamente ya que presenta un pequeño rizado y toma valores negativos.

El bloque *Binary Slicer* fragmenta un valor de tipo *float* de forma que la entrada positiva produce un 1 binario y la negativa un 0. Finalmente, para poder observar la señal demodulada será necesario transformar el flujo de *unsigned chars* en una cadena de datos de tipo *float*.

La señal obtenida finalmente se muestra en la gráfica correspondiente a la figura 5.21.

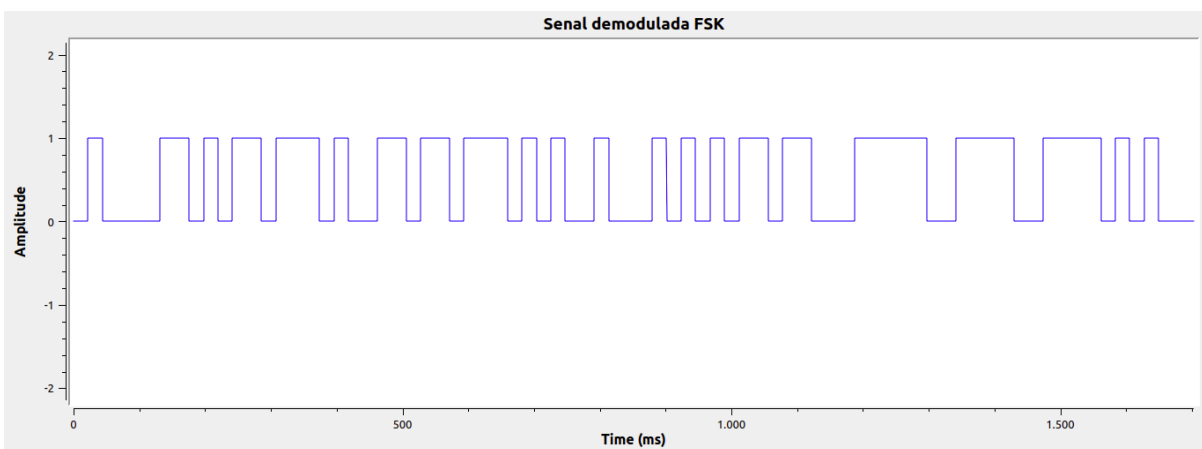


Figura 5.21: Señal demodulada FSK

5.4. Modulación digital de fase: PSK

5.4.1. Fundamentos de la modulación digital de fase

La modulación digital por desplazamiento de fase es una forma de modulación angular que se basa en hacer variar la fase de la señal portadora entre un número determinado de valores distintos; preservando así la información a transmitir en dichos cambios de fase.

La modulación PSK puede considerarse la versión digital de la modulación convencional PM analógica sobre una modulación PAM. La principal diferencia entre estas reside en que en PM las variaciones de fase son continuas mientras que en PSK, dado que la señal modulada es digital, el número de

estados será limitado.

La ecuación matemática a través de la cual se puede expresar la señal modulada es la siguiente 5.3.

$$S_m(t) = Ag(t) \cos(\omega_c t + \phi_m), \quad \phi_m = \phi_0 + \frac{2\pi}{M}(m-1), \quad 0 \leq t \leq T \quad m = 1, 2, \dots, M \quad (5.3)$$

En ella ϕ_0 es la fase inicial de la portadora y $g(t)$ un pulso de forma arbitraria cuyo periodo es el del símbolo.

Desarrollando la expresión anterior se llega a la siguiente:

$$S_m(t) = Ag(t) \cos(\phi_m) \cos(\omega_c t) - Ag(t) \sin(\phi_m) \sin(\omega_c t) \quad (5.4)$$

En este punto es posible separar la componente en fase $A_I = A \cos(\phi_m)$ de la componente en cuadratura $A_Q = A \sin(\phi_m)$.

La señal obtenida de este proceso junto con la señal de información se muestran en la figura 5.22. La señal moduladora se corresponde con una señal binaria con la información a transmitir con niveles de tensión entre 1 y -1 voltios. Para modularla se utiliza una señal portadora de alta frecuencia de la forma $\cos(\omega_c t)$. En la señal modulada se observa el proceso llevado a cabo en el modulador; cuando la entrada corresponde a un 0 lógico, la fase de la señal a la salida de este es de 180 grados mientras que si la entrada es un 1 lógico, la fase será de 0 grados [17]. La característica fundamental de esta modulación es que la fase de la portadora varía representando cada símbolo que se desee modular. Cada valor de la fase es elegido por el modulador entre un conjunto discreto de los posibles valores.

Dependiendo del número de símbolos y por consiguiente, del número de fases que se podrán tomar recibe diferentes denominaciones. Habitualmente se codifica un número entero de bits por cada símbolo por lo que el número de fases diferentes será una potencia de dos. Si se utilizan dos fases diferentes será un B-PSK mientras que siendo cuatro sería Q-PSK, si son 8 se denotará

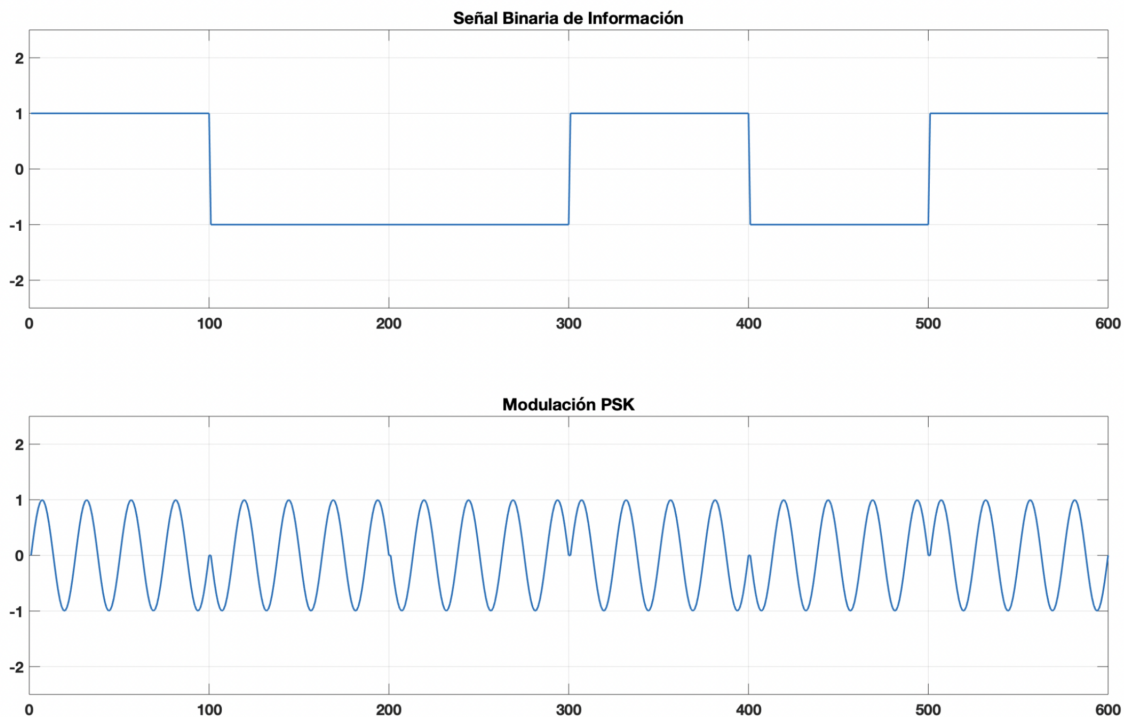


Figura 5.22: Señales en la modulación PSK

8-PSK y así sucesivamente.

Ampliando el número de símbolos se conseguirá transmitir la misma información empleando el mismo ancho de banda; por contra, mayor será la sensibilidad frente a ruidos e interferencias.

Una de las principales ventajas de esta modulación es que la potencia de todos sus símbolos es la misma. A continuación se muestra un ejemplo de su constelación en la figura 5.23.

5.4.2. Simulación de la modulación digital de fase en GNU Radio

Como en todas las secciones, se simulará la modulación PSK explicada anteriormente utilizando el software GNU Radio. El esquema utilizado para

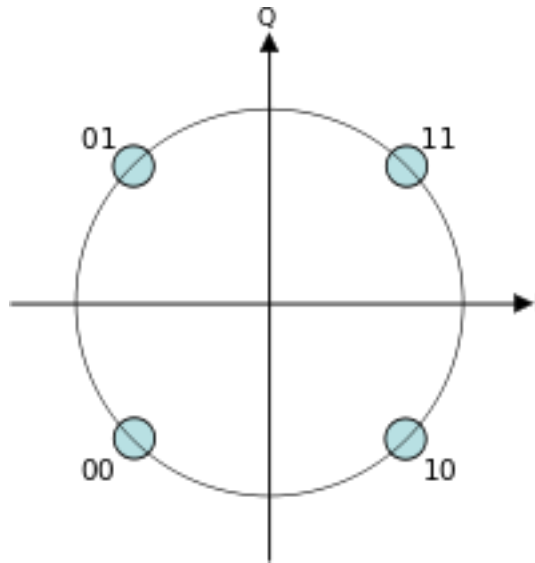


Figura 5.23: Constelación Q-PSK

ello es el mostrado en la figura 5.24.

En primer lugar se genera una secuencia de números en binario utilizando el bloque *Vector Source* en el que se introduce el siguiente vector: (0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2). A continuación se utiliza el bloque *Constellation Modulator* que permite la modulación de símbolos digitales a través de una constelación definida. En él se fijan sus parámetros:

- Muestras por símbolo (*Samples/symbol*): Se definen las muestras por símbolo que se utilizarán para muestrear la señal modulada; en este caso se fija a 4 muestras/símbolo.
- Codificación diferencial (*Differential Encoding*): La codificación diferencial se basa en transmitir la diferencia entre el símbolo actual y el símbolo anterior. En este parámetro se decide no utilizar este tipo de codificación ya que ya que al ser una simulación no será necesario sincronizar el receptor.
- *Excess Bandwidth*: Se utiliza para controlar el rolloff del filtro de raíz elevada utilizado para dar forma a la señal antes de ser transmitida. Su valor se fija comunmente a 0.35 para este tipo de constelaciones.

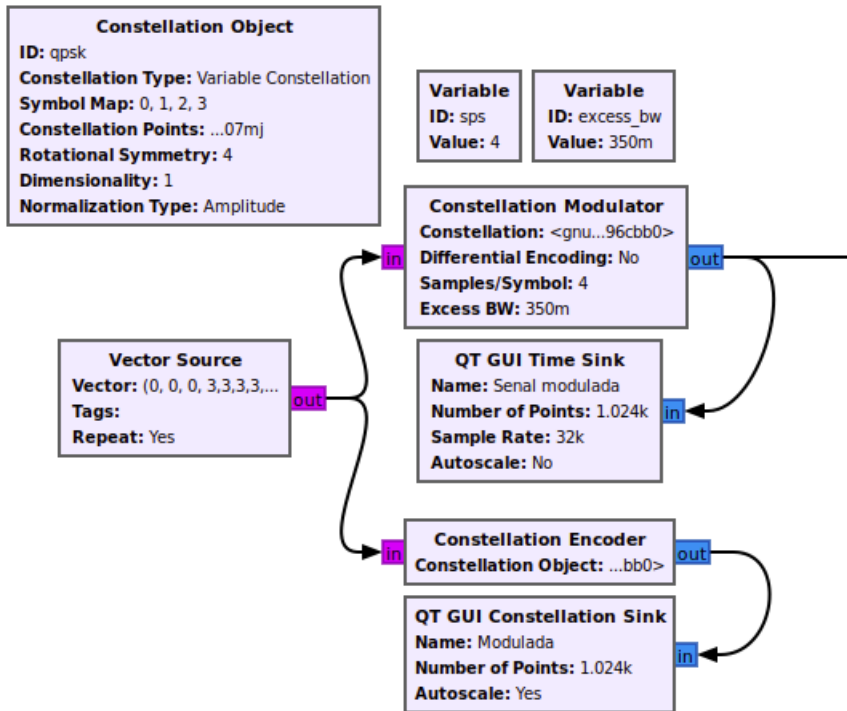


Figura 5.24: Esquema de modulación PSK en GNU Radio

- Puntos de la constelación (*Constellation Points*): Para definir los valores complejos de los símbolos de la constelación que se desea utilizar se define una variable del tipo *Constellation Rect. Object*. En ella se definen los puntos que son los valores complejos de los símbolos que conforman la constelación; en este caso se desean transmitir los símbolos 0, 1, 2 y 3 y sus valores serán $0,707 + 0,707j$, $-0,707 + 0,707j$, $-0,707 - 0,707j$ y $0,707 - 0,707j$ respectivamente.

La salida del bloque *Constellation Encoder* tendrá como resultado la señal modulada mostrada 5.25. En ella se observa como la información proveniente del vector ha sido introducida en los cambios de fase de la señal modulada.

Paralelamente, utilizando el bloque *Constellation Encoder* se logra convertir el flujo de bytes en una señal compleja modulada en banda base.

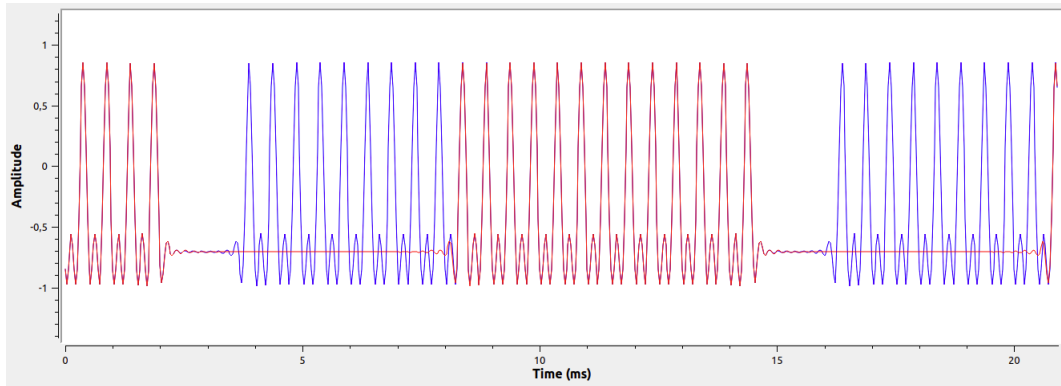


Figura 5.25: Señal modulada PSK

En este caso se deberá elegir la constelación definida anteriormente, dando lugar a la constelación recogida en la figura 5.26. En ella solo se aprecian los puntos correspondientes a los números 0, 2 y 3 ya que en el vector no ha sido introducido el número 1 con el fin de comprobar que se esté generando la constelación correctamente. Dado que la constelación elegida es de cuatro puntos se estará simulando una modulación 4-PSK.

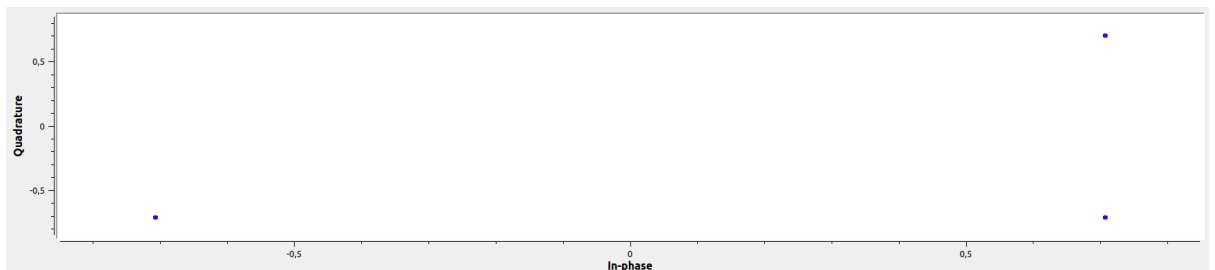


Figura 5.26: Constelación 4PSK codificada en GNU Radio

5.4.3. Fundamentos de la demodulación digital de fase

En esta sección se explicará el proceso de demodulación que será necesario llevar a cabo para recuperar la señal de información original tras haberla modulado.

A diferencia de las modulaciones ASK y FSK explicadas previamente, PSK

solo permite la demodulación utilizando el método de detección síncrona debido a que la señal presenta una envolvente constante; por ello es inútil pretender su demodulación empleando procedimientos que impliquen el uso del detector de envolvente. Además, estos procesos requieren de una señal portadora invariante así como de una señal binaria de alta frecuencia.

En la figura 5.27 se muestra el esquema del demodulador por detección síncrona. La señal de entrada al demodulador puede ser $\cos(\omega_c t)$ o $-\cos(\omega_c t)$. El circuito de recuperación de la portadora se encarga de detectar y regenerar la señal portadora que ha de ser coherente en frecuencia y fase con la portadora generada en el transmisor. La señal de entrada es multiplicada por la señal portadora generada utilizando el modulador balanceado que funciona como detector de producto.

La señal de salida del modulador balanceado pasa a través de un filtro paso bajo cuya misión es separa los datos binarios recuperados de la señal moduladora compleja. De esta forma se observará una señal de salida compuesta por un voltaje positivo y un coseno cuya frecuencia será el doble de la de la portadora, $2\omega_c$. Si la frecuencia de corte del filtro paso bajo utilizado es mucho más baja que la de la portadora solo logrará pasar la componente constante positiva; en caso contrario, solo pasaría la componente constante negativa [17].

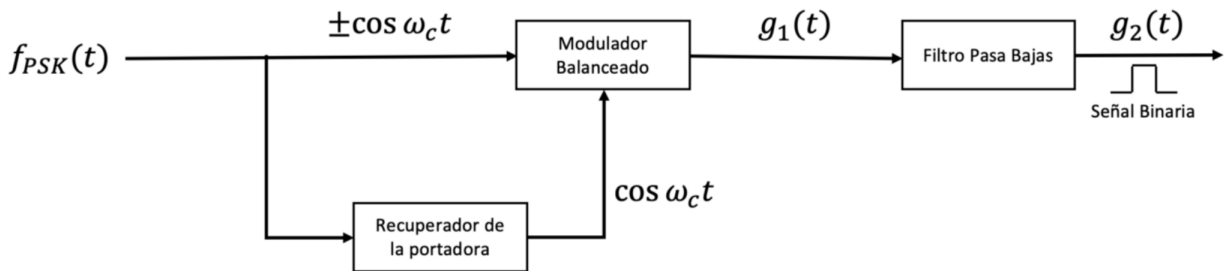


Figura 5.27: Esquema de la demodulación PSK mediante detección síncrona

Gracias al procedimiento descrito anteriormente se consigue recuperar la señal modulada anteriormente.

5.4.4. Simulación de la demodulación digital de fase en GNU Radio

A continuación se procede a simular en GNU Radio la demodulación digital de fase; para ello se utilizará el esquema mostrado en la figura 5.28.

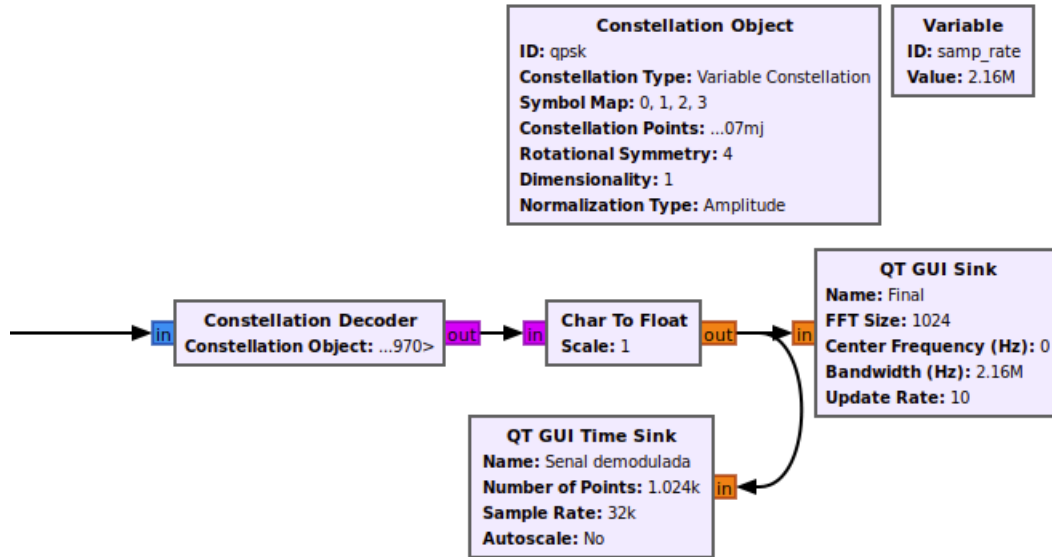


Figura 5.28: Esquema de demodulación PSK en GNU Radio

Para comenzar se introduce la señal modulada en el bloque *Constellation Decoder*; gracias a él se logran asociar los datos digitales a los símbolos de la constelación creada en el bloque *Constellation Object* que será la misma que en la modulación. Cabe destacar que en este caso no resulta necesario utilizar otros bloques ya que el modulador y el demodulador están en sincronía; como se observará en siguientes capítulos, no sucederá esto si la señal modulada es transmitida por un equipo y recibida por otro.

Gracias a esto se logra observar la constelación de la señal demodulada 5.29 que coincide con el vector utilizado para la modulación.

Además se podrá observar también la señal demodulada 5.30 cuyas amplitudes coinciden con las del vector.

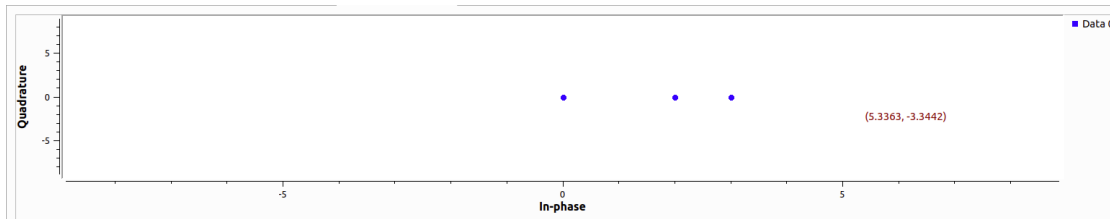


Figura 5.29: Constelación 4PSK decodificada en GNU Radio

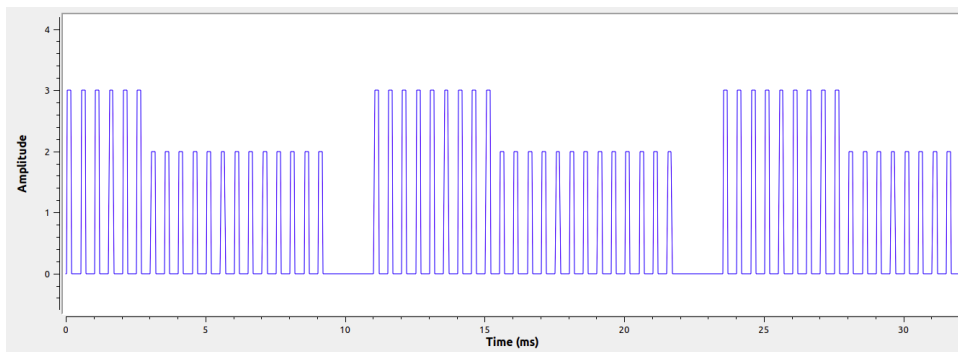


Figura 5.30: Señal demodulada

5.5. Modulación de amplitud en cuadratura: QAM

5.5.1. Fundamentos de la modulación de amplitud en cuadratura

La modulación de amplitud en cuadratura es una modulación digital híbrida que la información a transmitir se recoge tanto en la fase como con la amplitud de la señal modulada. Se basa en la transmisión de dos mensajes independientes utilizando un único camino; esto se consigue utilizando una misma portadora desfasada 90° entre uno y otro mensaje. Todo esto supone la formación de dos canales ortogonales en el mismo ancho de banda mejorando así la eficiencia respecto a la de las modulaciones explicadas anteriormente [8].

A continuación se mostrará el esquema de un modulador QAM básico 5.32. En él, los datos d_1 de entrada son generados a una velocidad de R_b bps y se agrupan mediante un conversor serie/paralelo formando palabras de J bits

que se introducen en el módulo de mapeo que se encargará de seleccionar un símbolo de entre los $M = 2^J$ posibles [18].

Dichos símbolos se pueden representar como símbolos complejos de la forma $c_k = a_k + jB_k$ por lo que el alfabeto estará formado por el conjunto completo de números complejos que representan los símbolos a transmitir.

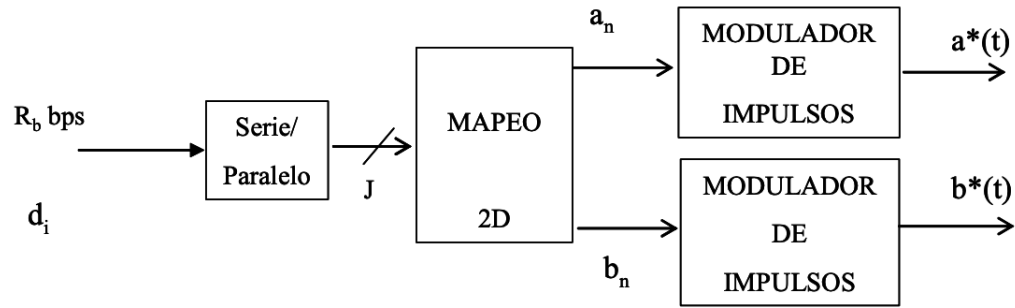


Figura 5.31: Modulador QAM

Otra forma de entender el esquema utilizado para modular una señal QAM es la mostrada en la figura. Una vez obtenidas las señales analógicas de información, se multiplican utilizando dos osciladores por dos señales portadoras idénticas cuya diferencia de fase es de 90 grados. Seguidamente se suman para ser procesadas como se desee en la posterior cadena de radio frecuencia. Debido a que la amplitud de la señal resultante es variante, los amplificadores RF utilizados deben ser lineales con el fin de preservar la señal ya que la no linealidad de estos podría modificar la diferencia de fase y haría irrecuperable la información modulada [19].

Como se ha explicado, esta modulación puede entenderse como dos señales N-PAM que modulan a una portadora, una en fase y otro en cuadratura. Por ello, la modulación M-QAM puede describirse matemáticamente de la siguiente forma:

$$S_m(t) = A_{mI}g(t) \cos(\omega_c t) - A_{mQ}g(t) \sin(\omega_c t), \quad m = 1, 2, \dots, M, \quad (5.5)$$

$$0 \leq t \leq T, A_{mI}, A_{mQ} \in \{A(2i - 1 - N), i = 1, \dots, N, N = \sqrt{M}\} \quad (5.6)$$

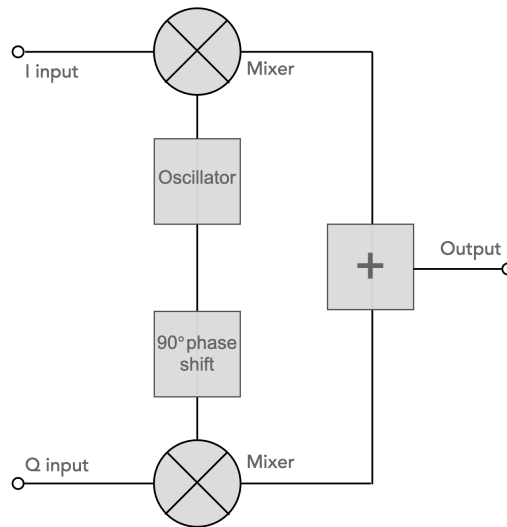


Figura 5.32: Esquema modulador QAM básico

En este caso, el espacio vectorial tendría dos dimensiones por lo que harían falta dos vectores ortonormales que formen la base.

La representación geométrica de las coordenadas se realiza en la constelación. Dependiendo del número de símbolos, M , utilizados se obtendrán diferentes constelaciones recogidas en la figura 5.33.

La inmunidad de la señal modulada en relación a las perturbaciones producidas en la línea y al ruido es mayor cuanto más separados estén los puntos de su constelación. El objetivo se encuentra por tanto en encontrar un diagrama cuyos puntos tengan la máxima separación posible entre ellos.

Finalmente, en la figura 5.34 se muestran las señales implicadas en la modulación. Las dos primeras corresponden a las dos portadoras cuya frecuencia es la misma desfasadas 90 grados. La tercera y la cuarta señal hacen referencia a las señales de información o moduladoras. Dichas señales son multiplicadas por las portadoras obteniéndose como resultado las señales 5 y 6 mostradas en la figura. Por último estas señales se suman para obtener la señal modulada correspondiente a la visualizada en último lugar.

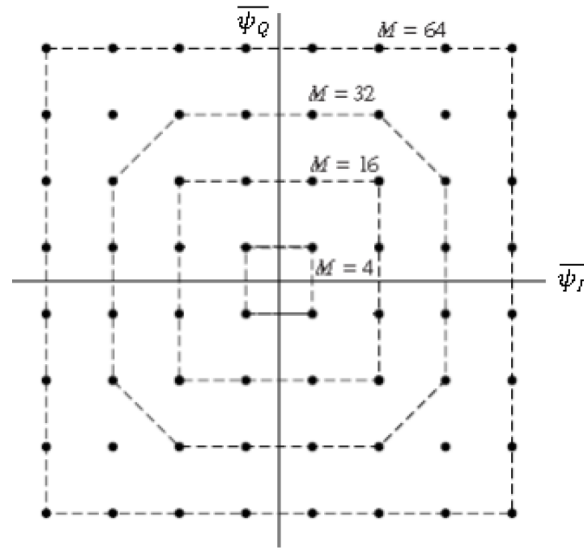


Figura 5.33: Constelaciones QAM

5.5.2. Simulación de la modulación de amplitud en cuadratura en GNU Radio

Para simular la modulación de amplitud en cuadratura, en este caso se emplearán 16 símbolos, es decir, será una modulación 16-QAM. El esquema utilizado en GNU radio es idéntico al implementado para la modulación PSK; solo variará la constelación utilizada. Se observa dicho esquema en la figura .

En primer lugar se genera una secuencia de números en binario utilizando el bloque *Random Source* gracias al cual se generan muestras aleatorias. Al igual que en PSK se utiliza el bloque *Constellation Modulator*, con los mismos parámetros, que permite la modulación de símbolos digitales a través de una constelación definida. La salida del bloque *Constellation Encoder* tendrá como resultado la señal modulada mostrada 5.36.

Paralelamente, utilizando el bloque *Constellation Encoder* se logra convertir el flujo de bytes en una señal compleja modulada en banda base.

Dado que la constelación elegida es de dieciseis puntos se estará simulando una modulación 16-QAM.

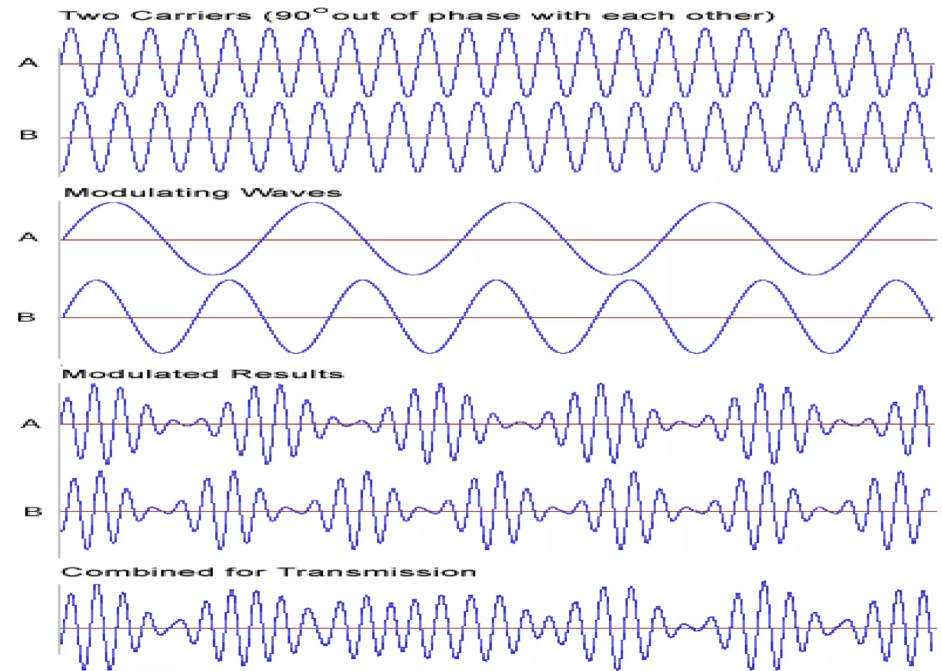


Figura 5.34: Señales en la modulación QAM

5.5.3. Fundamentos de la demodulación de amplitud en cuadratura

Tras conocer el proceso de modulación en cuadratura se procede a explicar el proceso de recuperación de la señal original.

El esquema del demodulador en cuadratura 5.38 se aprecia similar al de la modulación pero a la inversa. La señal de RF es multiplicada por la señal de un oscilador local en el canal I y por una señal igual desplazada 90 grados en el canal Q . El resultado tras un filtrado paso bajo de ambas señales será la forma de onda I y Q listas para su posterior procesamiento.

Los filtros paso bajo con necesarios ya que la multiplicación en cuadratura aplicada a la señal producirá un desplazamiento en frecuencia f_c , es decir, la frecuencia de la portadora. Con ellos se logrará suprimir el contenido de alta frecuencia asociado.

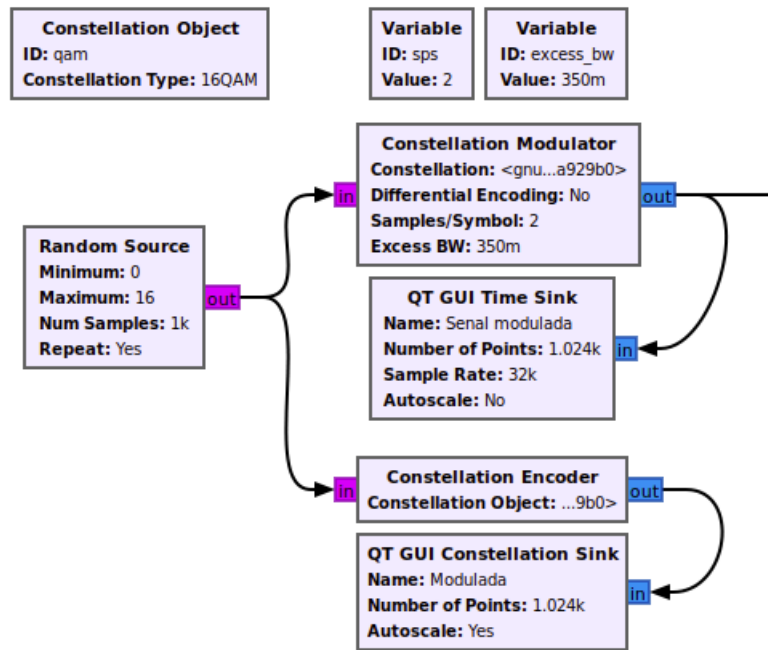


Figura 5.35: Esquema de modulación QAM en GNU Radio

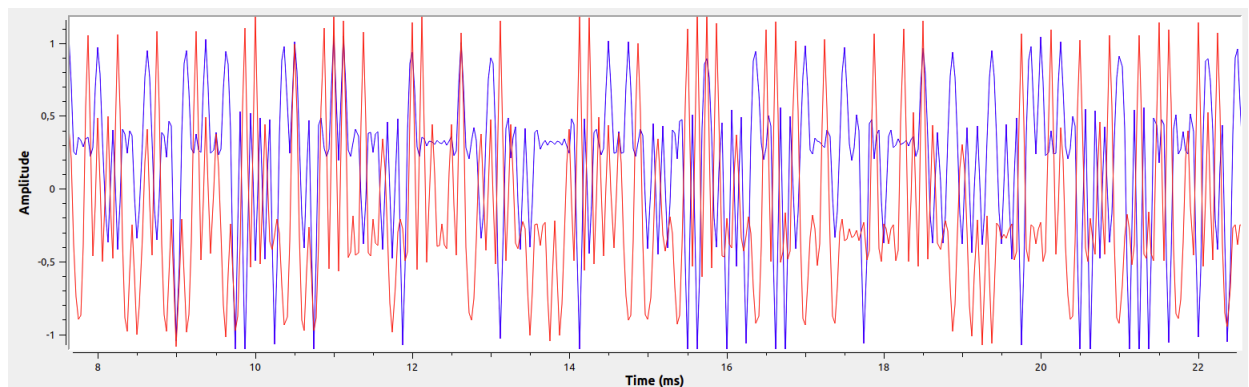


Figura 5.36: Señal modulada QAM

Observando el esquema se advierte que el demodulador de cuadratura está compuesto por dos demoduladores de amplitud. Al utilizar dos demoduladores de amplitud accionados por sinusoides a la frecuencia de la portadora con una diferencia de fase entre ellas de 90 grados se han generado dos señales de

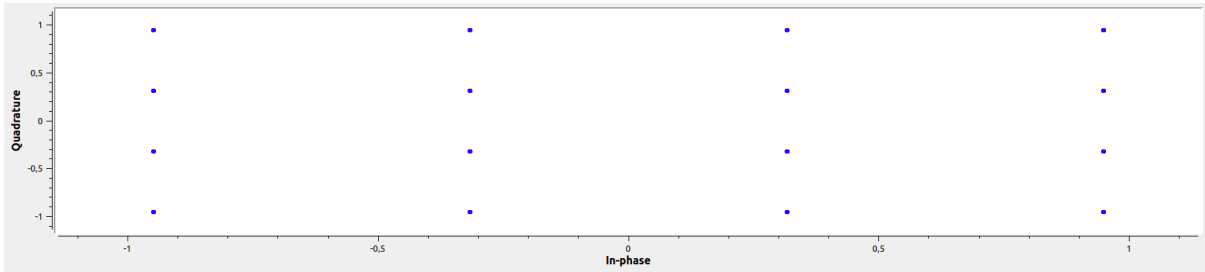


Figura 5.37: Constelación 16QAM codificada en GNU Radio

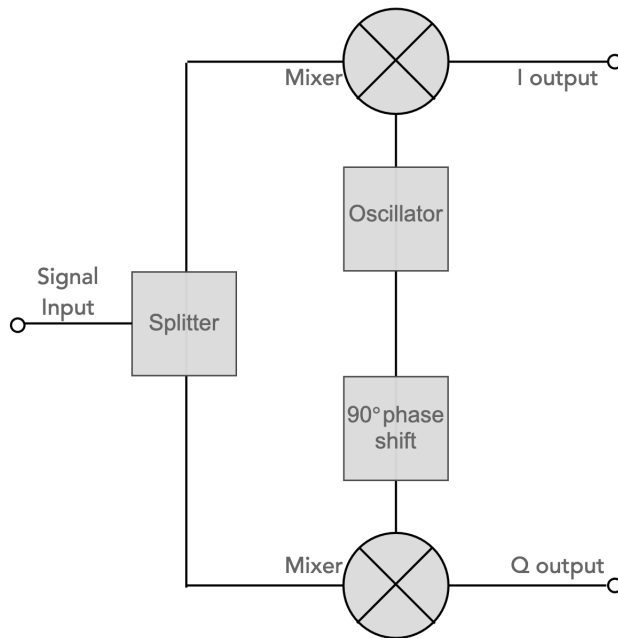


Figura 5.38: Esquema demodulador QAM básico

banda base capaces de transmitir información codificada utilizando cambios en su fase y su frecuencia [20].

Es importante utilizar una serie de circuitos adicionales que sincronicen la fase de la señal de referencia con la de la señal recibida.

Siguiendo el proceso detallado anteriormente se consigue recuperar las señales originales moduladas previamente.

5.5.4. Simulación de la demodulación de amplitud en cuadratura en GNU Radio

Al igual que en la modulación QAM, el esquema empleado para la demodulación también será idéntico al implementado para la demodulación PSK y se podrá observar en la figura 5.39.

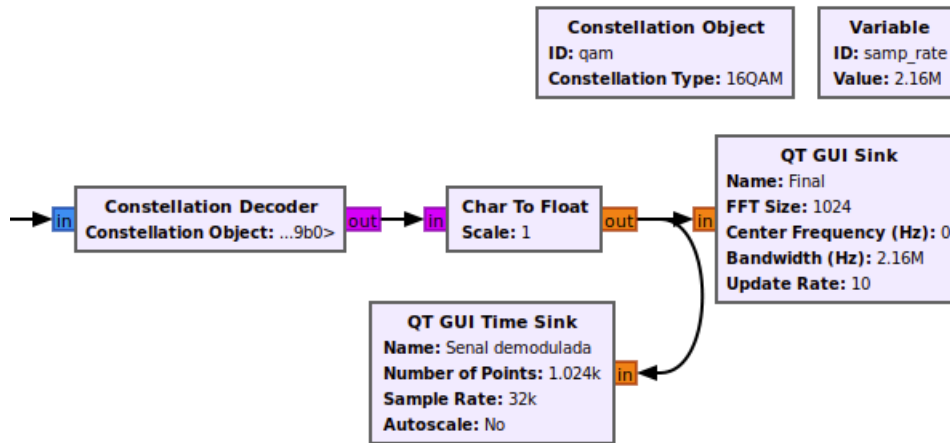


Figura 5.39: Esquema de demodulación QAM en GNU Radio

Gracias al bloque *Constellation Decoder* se logran asociar los datos digitales a los símbolos de la constelación creada en el bloque *Constellation Object* que será la misma que en la modulación.

Debido a esto se logra observar la constelación de la señal demodulada 5.40 que coincide con las muestras generadas para la modulación.

Además se podrá observar también la señal demodulada 5.41 cuyas amplitudes oscilan debido a la aleatoriedad de las muestras generadas.

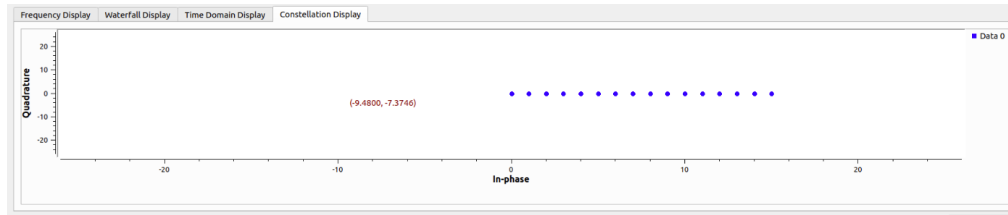


Figura 5.40: Constelación QAM decodificada en GNU Radio

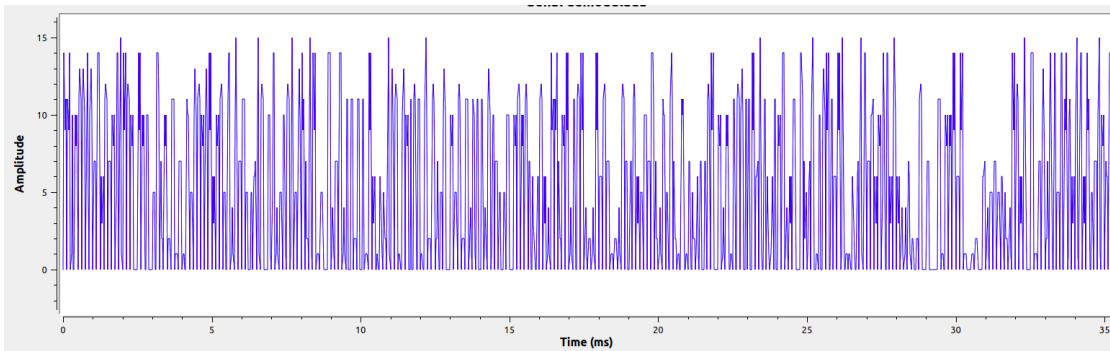


Figura 5.41: Señal demodulada

Capítulo 6

Resultados de la transmisión y recepción analógica

6.1. Introducción

Habiendo concluido las explicaciones teóricas de las modulaciones analógicas en amplitud y frecuencia se procede a poner en práctica los conocimientos adquiridos para lograr transmitir y recibir señales analógicas utilizando el software GNU Radio. Para la transmisión de dichas señales se utilizará el Hack RF cuya tecnología se ha expuesto previamente en el capítulo 3 mientras que para la recepción se utilizará un RTL-SDR común similar al explicado en el capítulo citado con anterioridad.

6.2. Modulación en amplitud

6.2.1. Resultados de la transmisión AM en GNU Radio

Para modular la señal en AM con el objetivo de transmitirla se utilizará como se ha explicado anteriormente el software GNU Radio. El esquema utilizado para la transmisión será el mostrado en la figura 6.1.

Dicho esquema se compone de diferentes bloques. En primer lugar se toma una señal de audio en formato .wav empleando el bloque *Wav File Source* ya utilizado anteriormente. Esta señal de audio corresponde a la moduladora y

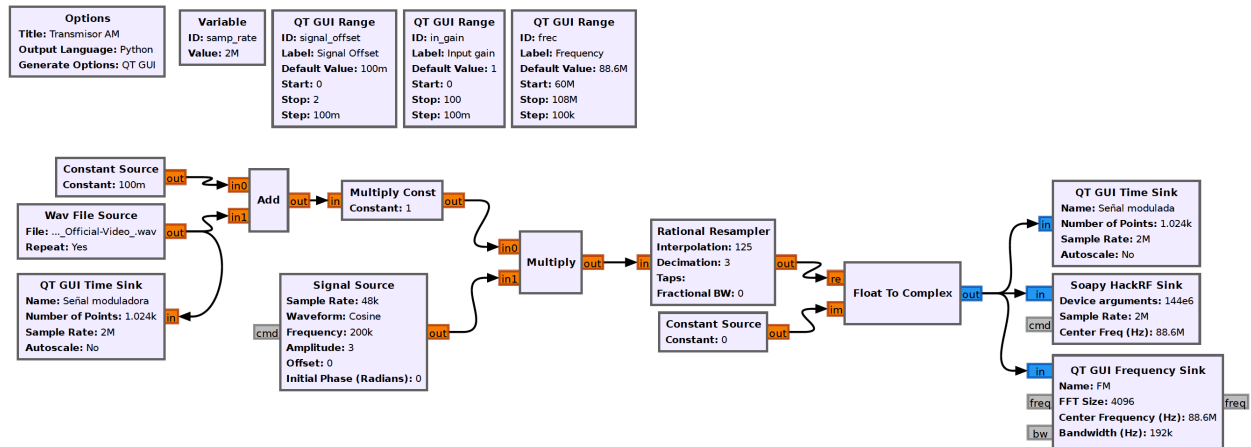


Figura 6.1: Esquema del transmisor AM en GNU Radio

se puede observar en la figura 6.2.

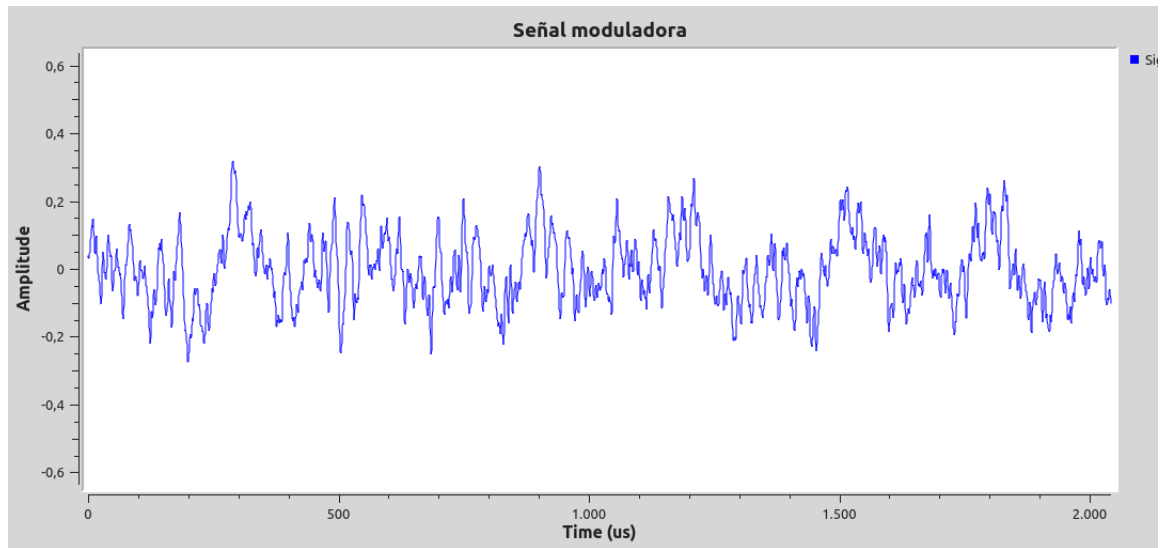


Figura 6.2: Señal moduladora

Una vez generada la señal, siguiendo el esquema de modulación AM explicado en capítulos anteriores, se le añade una constante utilizando el bloque *Add*.

A su salida se fija una ganancia mediante el bloque *Multiply Const*; de esta forma ya estaría preparada la señal de información para multiplicarla por la portadora, que será una señal de tipo coseno cuya frecuencia se fija a 200kHz. La tasa de muestreo de esta señal será de 48kHz ya que es conveniente que sea igual que la de la señal de información. En la figura 6.3 se muestra la señal que resulta de multiplicar la portadora por la señal de moduladora previamente tratada.

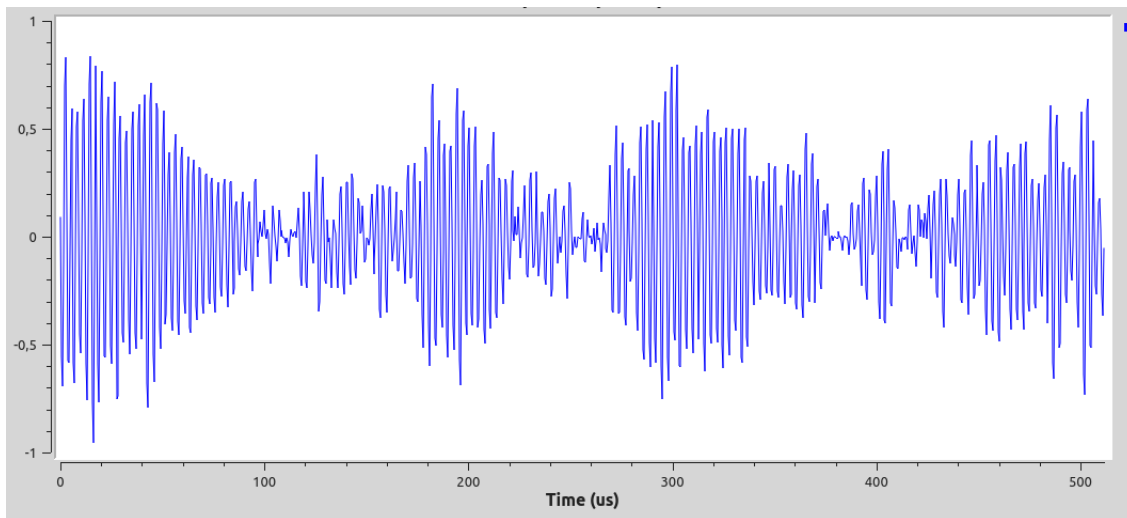


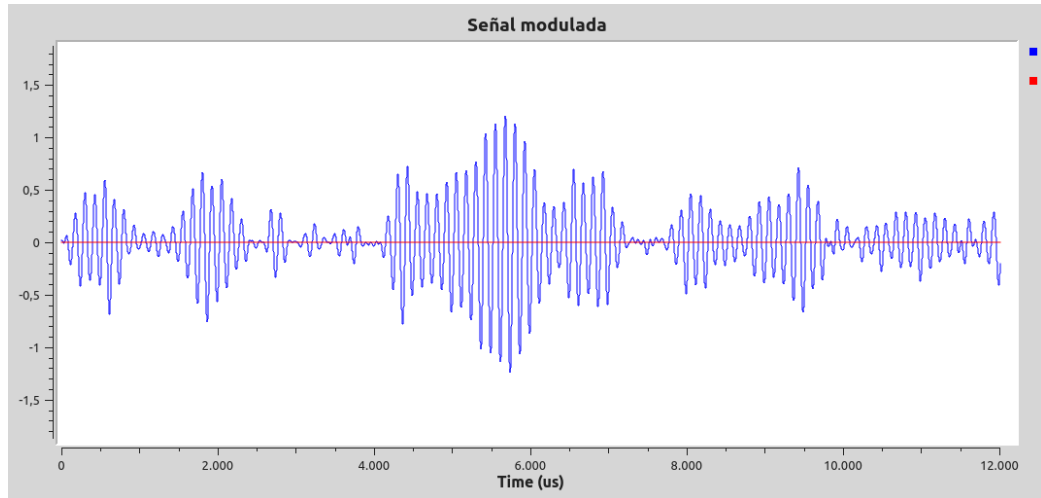
Figura 6.3: Señal resultante a la salida del multiplicador

Es necesario realizar un remuestreo a la salida del multiplicador, para ello se emplea el bloque *Rational Resampler*. Dado que la frecuencia de muestreo necesaria en transmisor es de 2MHz, se aplica a la señal muestreada a 48kHz una interpolación de 125 y un diezmado de 3 con el fin de obtener el resultado deseado.

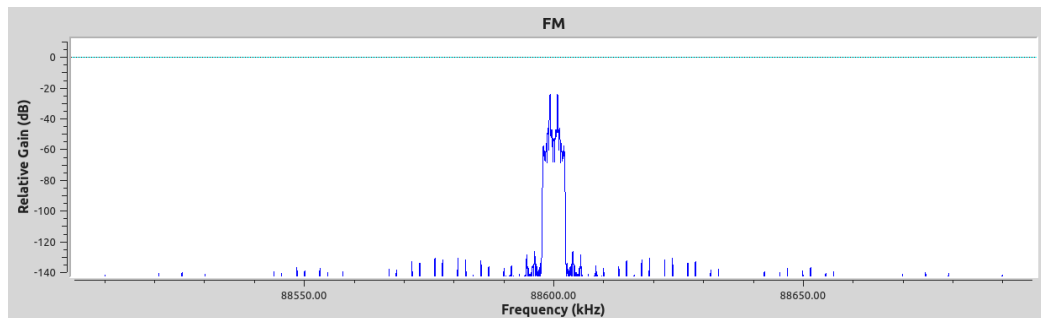
Dado que para poder transmitir una señal esta debe ser compleja, se le aplica una conversión a complejos mediante el uso del bloque *Float To Complex*.

Realizado todo lo anterior, la señal estará lista para su transmisión utilizando el dispositivo HackRF. La frecuencia de transmisión se recoge en una variable de tipo *QT GUI Range* con el objetivo de poder ajustarla en función de las condiciones del espectro. En la figura 6.2.1 se muestra dicha señal

modulada en el tiempo y en la 6.4(b) se puede observar su espectro.



(a) Señal modulada AM en tiempo



(b) Señal modulada AM en frecuencia

Figura 6.4: Señal modulada AM para su transmisión

Procedimiento alternativo para la transmisión AM

En la última etapa de esta modulación simulada en GNU radio para la transmisión, se multiplica la señal resultante de etapas anteriores por la portadora. Teniendo en cuenta el esquema del transmisor explicado en el Capítulo 2, este paso no tendría sentido ya que en el transmisor se efectúa esta multiplicación, en ese caso sería la frecuencia de transmisión la de la portadora. En efecto, la simulación realizada anteriormente para la transmisión sería una doble modulación AM.

Debido a esta conclusión resulta interesante plantear un esquema de modulación alternativo en el cual la multiplicación por la portadora sea realizada directamente por el transmisor utilizado, en este caso, el Hack RF.

El esquema utilizado para la transmisión es el mostrado en la figura 6.5.

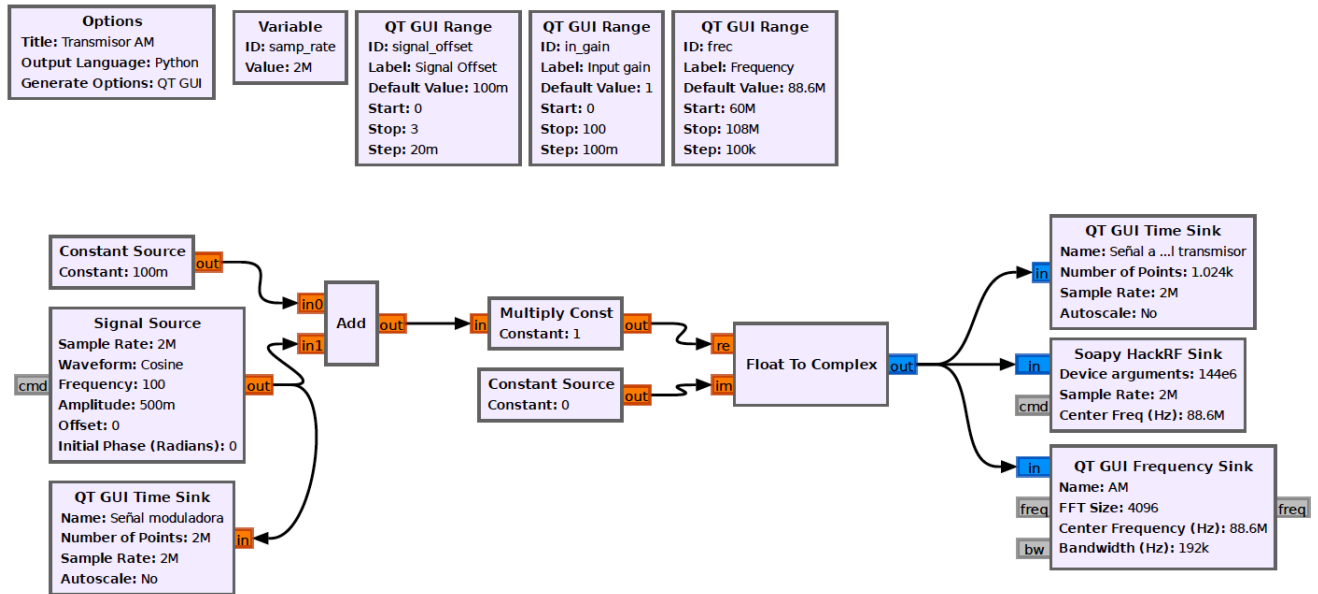


Figura 6.5: Esquema del transmisor AM en GNU Radio. Procedimiento alternativo

En dicho esquema se parte de una señal moduladora coseno cuya frecuencia es de 100Hz y muestreada a la frecuencia de muestreo a la que funciona el transmisor, 2MHz . Esta señal es generada con ayuda del bloque *Signal Source* y resulta la mostrada en la figura 6.6.

Al igual que en la expresión teórica de la modulación AM se le suma una constante cuyo valor se fija a 0.76. Tras dicha suma se coloca el bloque *Multiply Const* para modificar la amplitud de la señal según convenga.

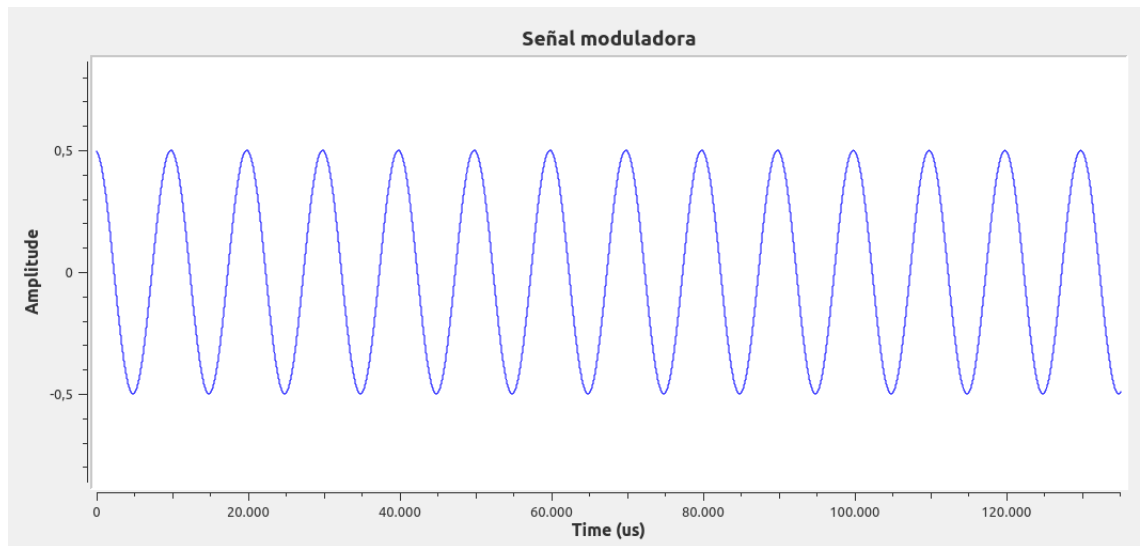


Figura 6.6: Señal moduladora. Procedimiento alternativo

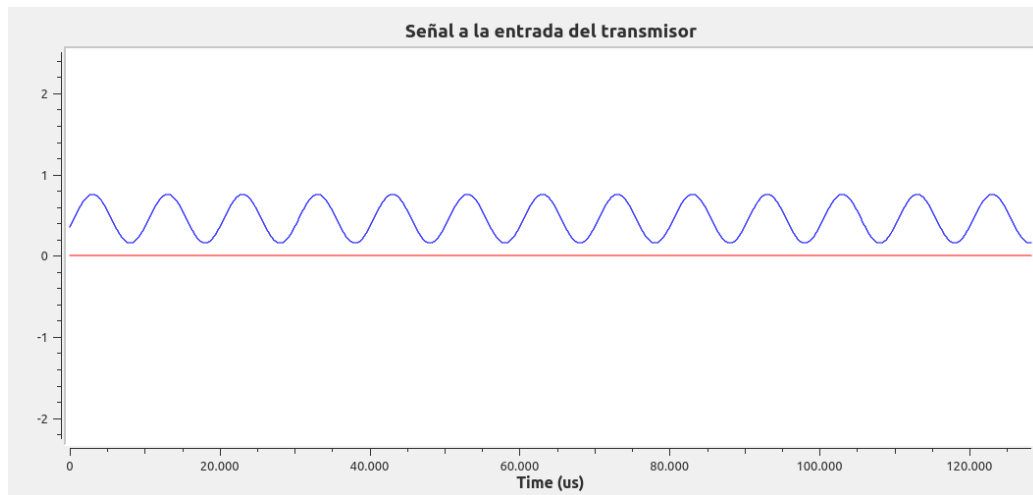
Mediante el bloque *Float To Complex* se transforma la señal en compleja para introducirla en el transmisor ya que este trabaja con dos flujos de muestras, uno complejo y otro imaginario.

Se puede observar el aspecto de la señal a la entrada del transmisor en la figura 6.2.1 y su espectro en la 6.7(b). Es importante destacar que para finalizar la modulación, esta señal deberá ser multiplicada por la portadora; esto no es posible visualizarlo debido a que esta operación es efectuada en el interior del HackRF.

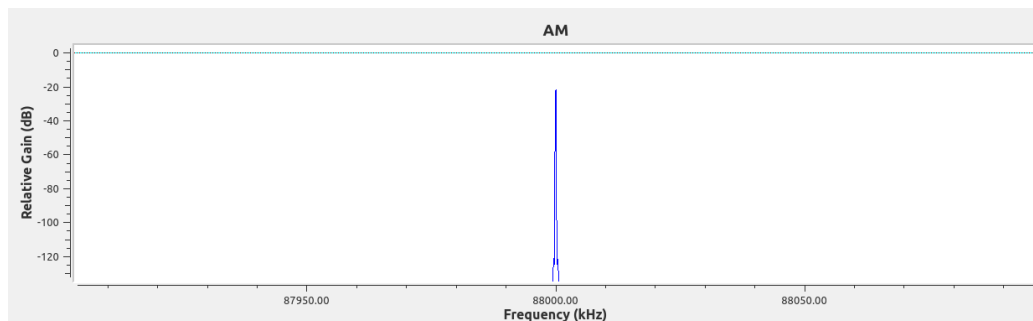
6.2.2. Resultados de la recepción AM en GNU Radio

Tras haber modulado la señal y haberse transmitido por el aire, se procede a recibirla y demodularla en otro equipo para lo cual se emplea el esquema mostrado en la figura 6.8.

Utilizando un receptor RTL-SDR se recibe la señal mostrada en la figura 6.2.2 cuyo espectro 6.2.2 se muestra en banda base. La frecuencia de muestreo utilizada en el receptor es 2,16MHz.



(a) Señal modulada AM en tiempo. Procedimiento alternativo



(b) Señal modulada AM en frecuencia. Procedimiento alternativo

Figura 6.7: Señal modulada AM para su transmisión. Procedimiento alternativo

En primer lugar será necesario realizar un filtrado de la señal, para ello se utilizará un filtro paso bajo, su bloque en GNU Radio es el *Low Pass Filter*. Se utilizará una ventana Hamming para el filtrado con una frecuencia de corte de 5kHz y una banda de transición de 56kHz . Se aplica una ganancia de 10 en el filtro con el fin de amplificar la señal filtrada. Se muestra la señal filtrada en el dominio del tiempo 6.2.2 y en el de la frecuencia 6.2.2.

Debido a que la modulación es en amplitud y que la señal se puede ver afectada en la transmisión por las condiciones del espectro, será necesario

110CAPÍTULO 6. RESULTADOS DE LA TRANSMISIÓN Y RECEPCIÓN ANALÓGICA

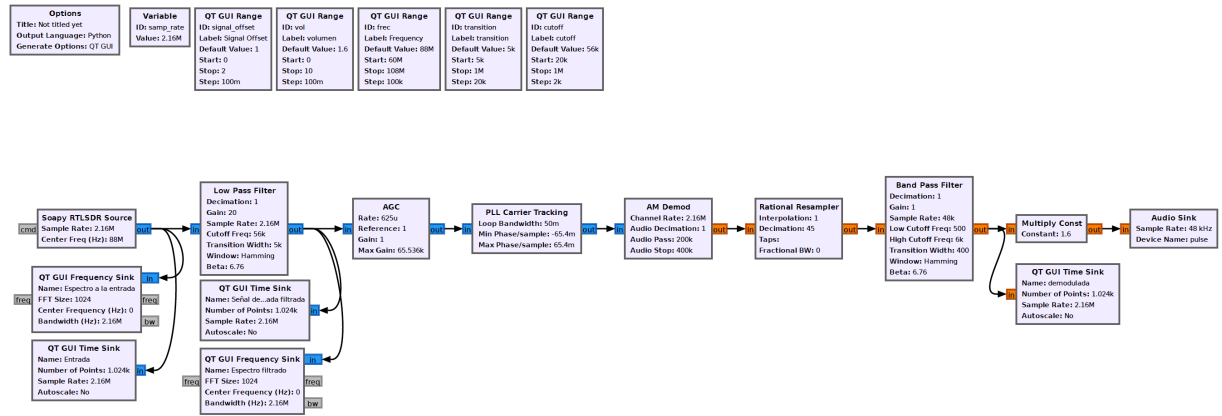


Figura 6.8: Esquema del receptor AM en GNU Radio

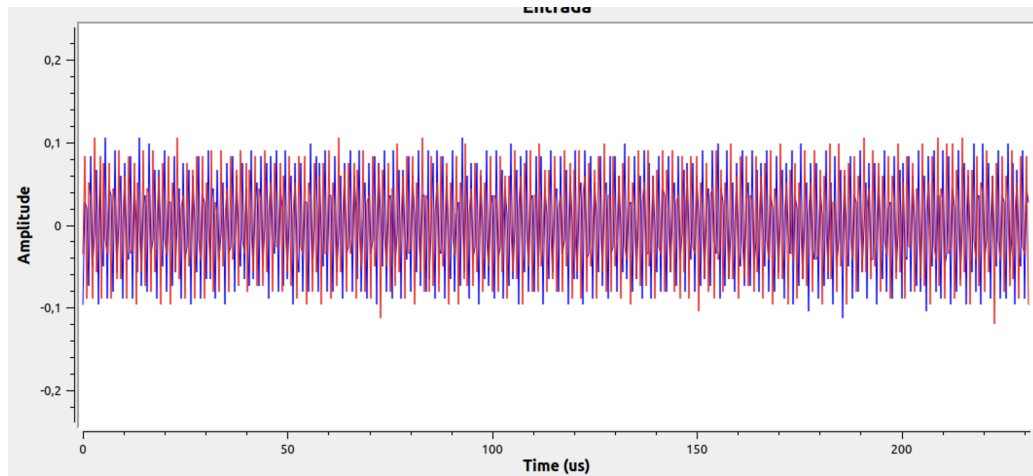
utilizar un bloque de control automático de ganancia *AGC*.

Tras él, es necesario colocar el bloque *PLL Carrier Tracking* que bloquea la portadora y emite la señal antes de haber sido mezclada con la portadora.

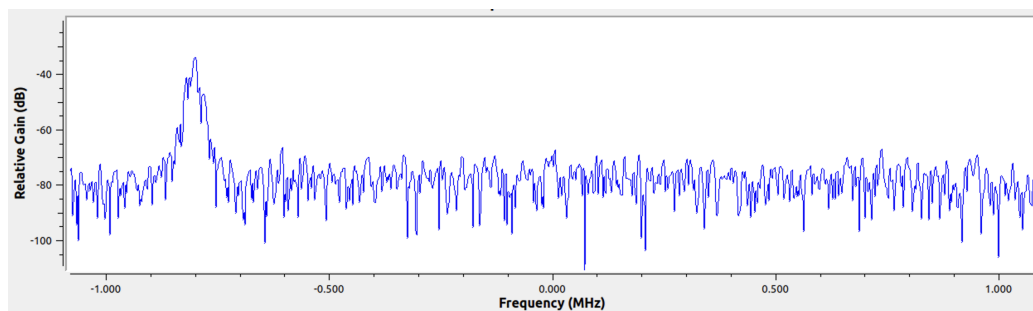
Se utiliza el *AM Demod* para demodular la señal AM aplicando un filtro paso bajo a la salida de audio. Para ello se introducen en él una serie de parámetros; la tasa de muestro *Channel Rate* será la utilizada hasta ahora, $2,16MHz$, se emplea una frecuencia para la banda de paso del filtrado paso bajo de $200kHz$ y una frecuencia de parada de $400kHz$.

A continuación, a consecuencia de que se pretende obtener una señal de audio, será necesario realizar un remuestreo de la señal con el fin de que la resultante sea de $48kHz$ para que pueda ser escuchada a través de un dispositivo de audio. Para ello se utiliza el bloque *Rational Resampler*. Dado que la señal introducida en dicho bloque tiene una frecuencia de muestreo de $2,16MHz$, será necesario realizar un diezmado de 45 para obtener una resultante de $48kHz$.

Tras haber remuestreado la señal se procede nuevamente a filtrarla, esta vez utilizando un filtro paso banda con ayuda del bloque *Band Pass Filter* aplicando una frecuencia de corte inferior de $500Hz$ y superior de $6kHz$ con una ventana de tipo Hamming.



(a) Señal recibida en tiempo

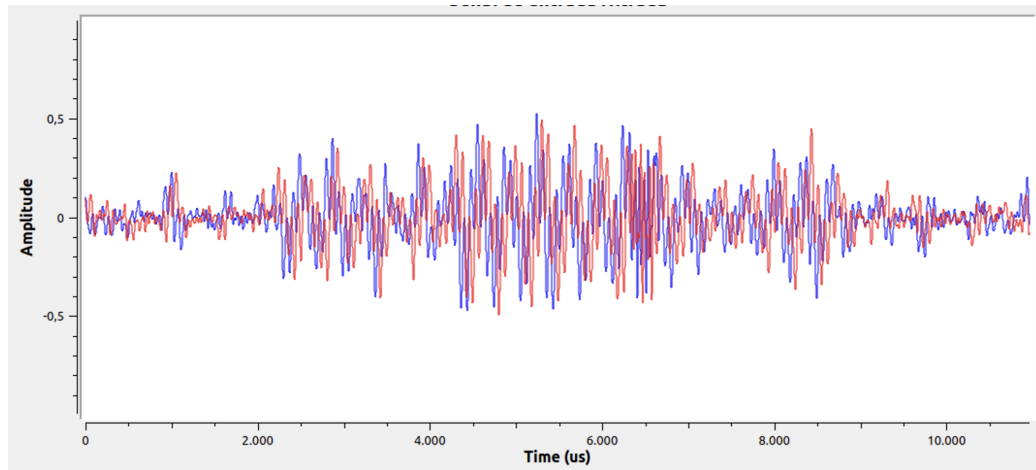


(b) Espectro recibido

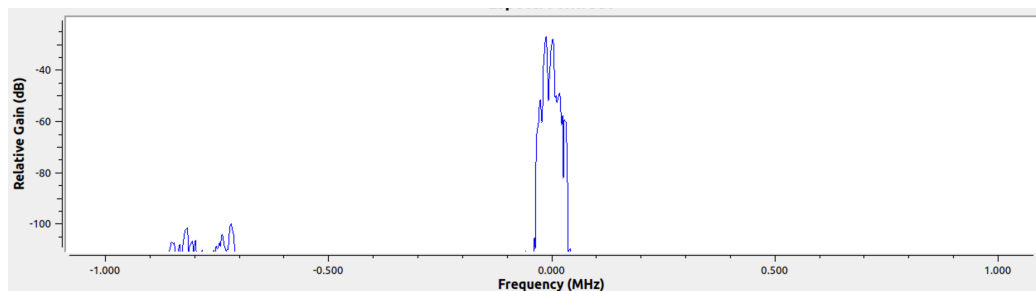
Figura 6.9: Entrada del receptor RTL-SDR

Una vez realizados todos los pasos descritos anteriormente se podrá observar la señal demodulada AM 6.11, muy similar a la moduladora inicial. Gracias a la salida de audio se observa cómo se ha transmitido correctamente la señal.

Además, se podrá emplear el bloque *Multiply Const* para regular la ganancia y por tanto el volumen como desee. Gracias al bloque *Audio Sink* se puede escuchar la señal final utilizando cualquier equipo de audio, observándose así como es prácticamente idéntica a la transmitida desde el equipo transmisor.



(a) Señal filtrada en tiempo



(b) Espectro de la señal filtrada

Figura 6.10: Señal filtrada mediante un filtro paso bajo

Procedimiento alternativo para la recepción AM

Se planteará ahora un esquema en GNU radio 6.12 para la recepción y demodulación de la señal transmitida en la sección 6.2.1

Como se comentó para la transmisión, el transmisor efectúa una multiplicación por una señal portadora cuya frecuencia es la fijada para la transmisión. De la misma forma, observando el esquema del receptor explicado en el Capítulo 2, cuando la señal llega al receptor se deshace dicha operación.

A continuación se explicarán los pasos a seguir para la demodulación de la señal recibida, la cuál se muestra en tiempo en la figura 6.2.2 y cuyo es-

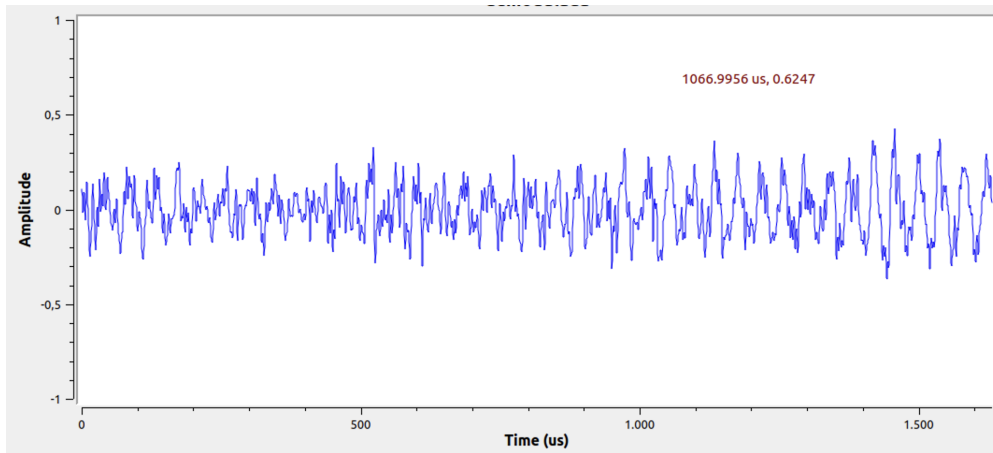


Figura 6.11: Señal demodulada AM

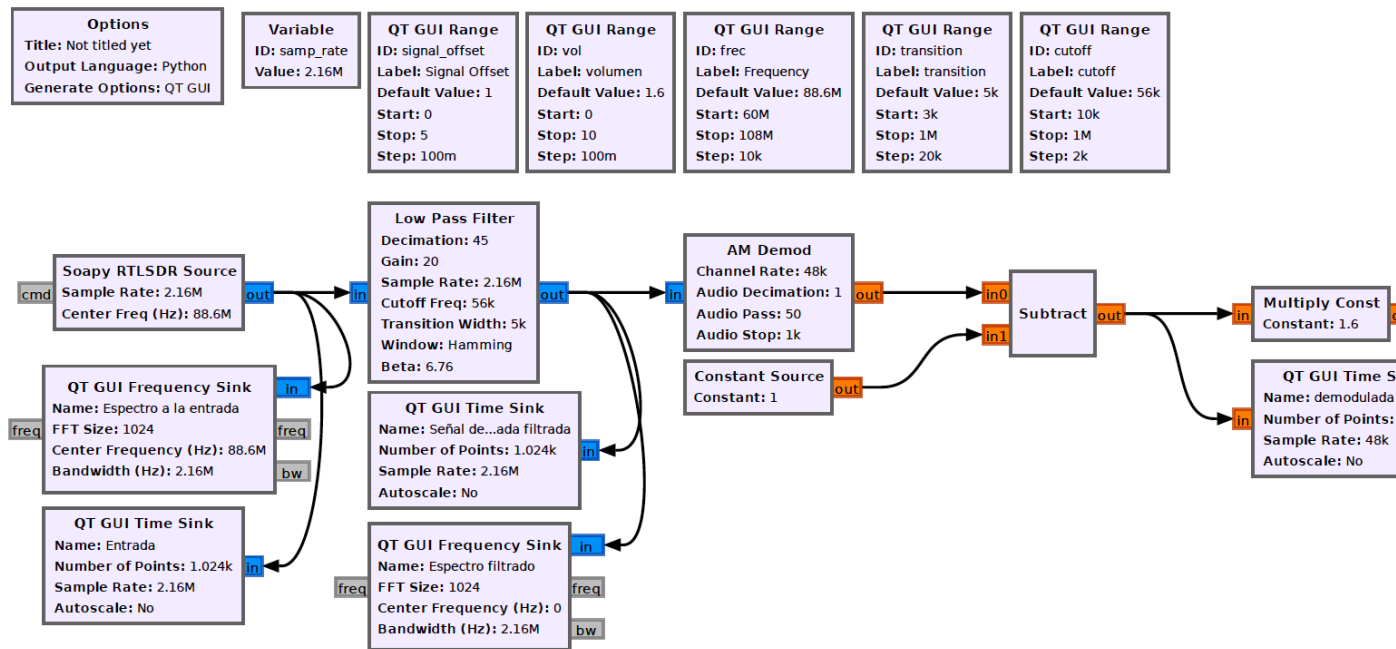
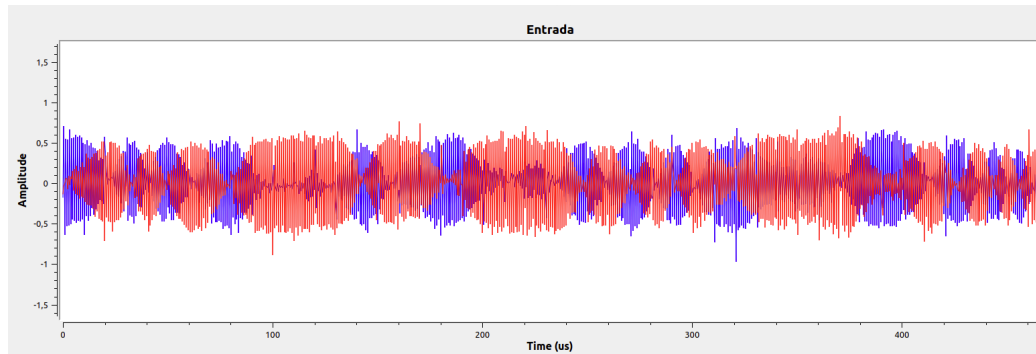


Figura 6.12: Esquema del receptor AM en GNU Radio. Procedimiento alternativo

pectro es 6.2.2.



(a) Señal recibida en tiempo. Procedimiento alternativo.

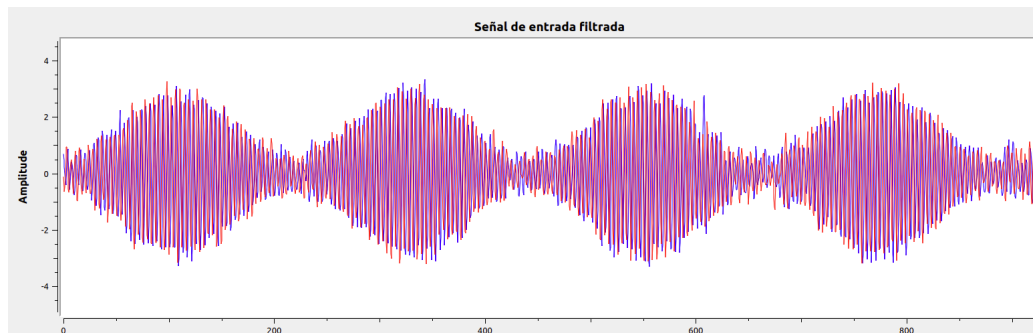


(b) Espectro recibido. Procedimiento alternativo

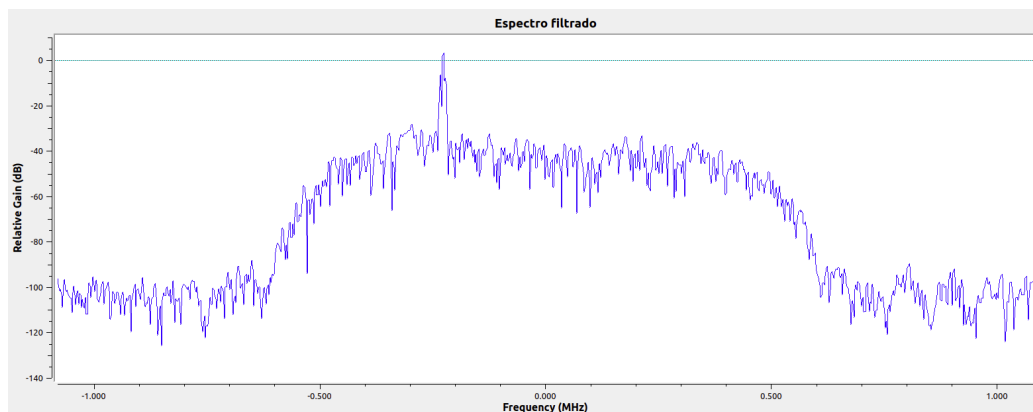
Figura 6.13: Entrada del receptor RTL-SDR. Procedimiento alternativo

En primer lugar resulta necesario filtrar la señal recibida con el fin de eliminar las componentes de alta frecuencia, para ello se hace uso de un filtro paso bajo con una frecuencia de corte de 10kHz . Puesto que la tasa de muestreo del receptor es de $2,16\text{MHz}$, se le aplica un diezmado de 45; de esta forma, la señal pasará a tener una frecuencia de muestreo de 48kHz . De esta forma, a la salida del filtro se obtiene la señal de la figura 6.2.2 cuyo espectro es el mostrado en la 6.2.2.

Tras el filtrado se observa claramente en la figura 6.2.2 cómo varía la señal su amplitud de forma sinusoidal, sin embargo, la componente de alta frecuencia aún no ha desaparecido. Prestando atención al espectro de la señal se percibe



(a) Señal filtrada en tiempo. Procedimiento alternativo



(b) Espectro de la señal filtrada. Procedimiento alternativo

Figura 6.14: Señal filtrada mediante un filtro paso bajo. Procedimiento alternativo

una desviación en frecuencia de unos 200kHz . Esta es debida a la escasa calidad del oscilador del receptor. Por ello, para conseguir demodular la señal, será necesario utilizar el bloque *AM Demod*, que actúa como un detector de envolvente consiguiendo extraer la señal demodulada. Tras él será necesario eliminar el nivel de offset introducido en la transmisión.

De esta forma se consigue recuperar finalmente la señal modulada en la transmisión, un coseno de frecuencia 100Hz que coincide con la señal mostrada en la figura 6.15.

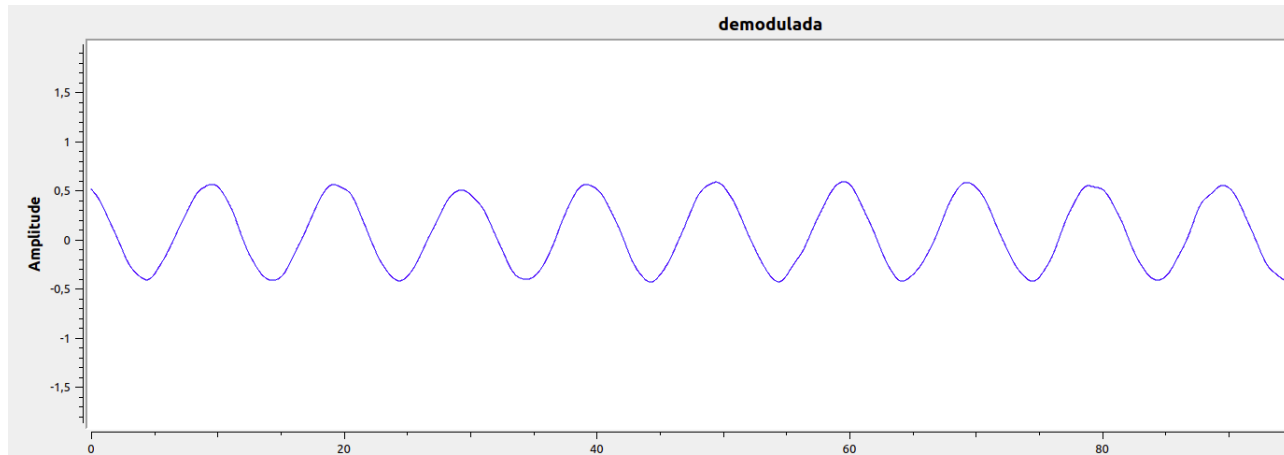


Figura 6.15: Señal demodulada AM. Procedimiento alternativo

6.3. Modulación en frecuencia

6.3.1. Resultados de la transmisión FM en GNU Radio

Con el objetivo de modular correctamente la señal para posteriormente transmitirla se utilizará el siguiente esquema en GNU Radio 6.16 el cual se explicará detenidamente a lo largo de esta sección.

Para comenzar se utilizará una señal de audio con forma de coseno, para generar esta señal se recurre al bloque *Signal Source* en el cual será necesario introducir la frecuencia de la señal, en este caso 100Hz . Se fija también la tasa de muestreo de la misma a 2MHz coincidiendo con la de funcionamiento del transmisor. En la figura se muestra la señal generada que se utilizará como moduladora.

Posteriormente, gracias al bloque *Multiply Const* se logra amplificar esta señal lo deseado. En este caso, se ha utilizado una variable de tipo *QT Gui Range* que permitirá modificar este valor en el transcurso de la ejecución del programa.

El *WBFM Transmit* es un bloque modulador FM de banda estrecha que utiliza un único flujo de muestras de tipo *float* produciendo una única salida de banda base compleja modulada en FM. En él se introducen una serie de

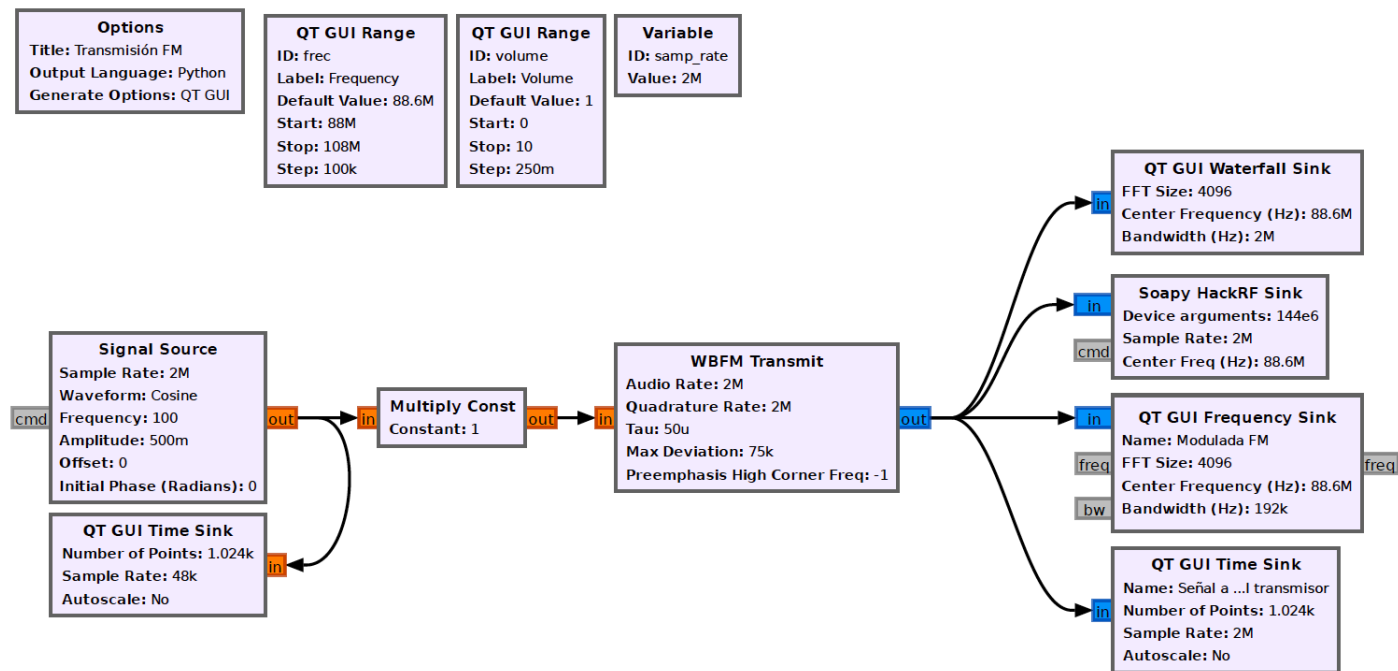


Figura 6.16: Esquema del transmisor FM en GNU radio

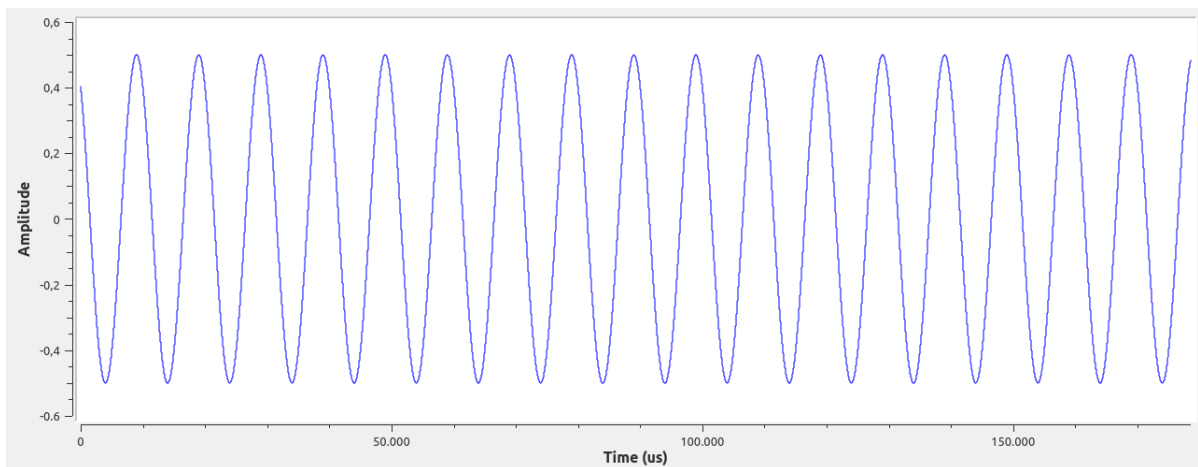


Figura 6.17: Señal moduladora

variables; el *Audio Rate* hace referencia a la tasa de muestreo de la señal de entrada, por eso en este caso se fija a 2MHz . *Quadrature Rate* es la tasa de

muestreo del flujo de salida; debe ser múltiplo del *Audio Rate* por lo que se establece también a $2MHz$ haciéndola coincidir con la del transmisor.

La constante de tiempo de preénfasis (o énfasis previo) es un parámetro utilizado en el procesamiento de señales de audio para mejorar la calidad de la señal y reducir el ruido. La preénfasis es una operación que aumenta la amplitud de las frecuencias altas en una señal de audio antes de su transmisión o grabación y luego, en el receptor o durante la reproducción, se aplica una operación inversa llamada desénfasis para restaurar la señal a su forma original. La constante de tiempo de preénfasis es un factor que controla la velocidad a la que se aplican estas operaciones. *Tau* es la constante de tiempo de preénfasis y se establece en $50\mu s$. Respecto a esto se elegirá también la frecuencia a la que se utilizará dicho preénfasis, *Preemphasis High Corner Freq*.

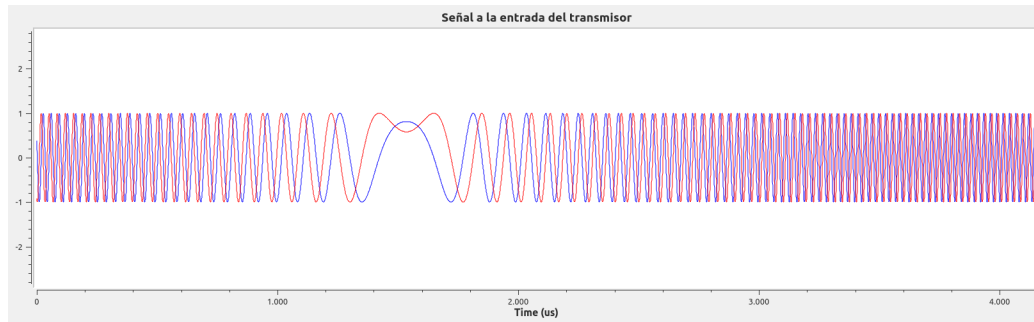
El parámetro *Max Deviation* hace referencia a la desviación máxima en Hercios que se utilizará en la modulación, en este caso se deja su valor por defecto, $75kHz$. Esto controla la cantidad de ancho de banda ocupado por la señal FM. Cuanto mayor sea la desviación de frecuencia, más ancho de banda ocupará la señal.

Tras haber concluido el proceso de modulación de la señal, se observa en la figura la forma de la señal modulada en el dominio del tiempo. Se aprecian las variaciones de frecuencia de la portadora elegida 6.3.1 en las que queda registrada la información para ser transmitida y posteriormente demodulada.

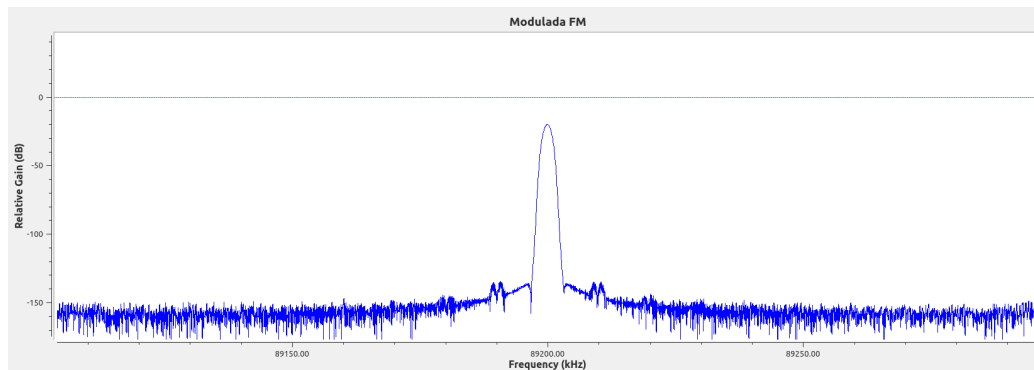
Hasta el momento no se ha hecho referencia al dominio frecuencial ya que resulta más comprensible explicarlo desde el punto de vista de la transmisión y la recepción de señales.

Se observa cómo el espectro de la señal FM mostrado en la figura 6.18(b) se encuentra centrado en la frecuencia elegida para la transmisión; esta debe coincidir con la que se determinará posteriormente en el bloque transmisor. Cabe destacar también que el espectro de la señal FM será más ancho que el de AM a consecuencia de que es en esa variación frecuencial donde se consigue recoger la información modulada como se ha explicado previamente.

En el caso de que la frecuencia de muestreo de la señal a la salida del modu-



(a) Señal modulada FM en tiempo



(b) Señal modulada FM en frecuencia

Figura 6.18: Señal modulada FM para su transmisión

lador no coincidiese con la de trabajo del HackRF ($2MHz$) sería necesario emplear el bloque *Rational Resampler* cuya función, como ya se ha explicado, es realizar un submuestreo con el objetivo de preparar la señal para su transmisión. Su funcionamiento se basa en la comprensión del proceso de interpolación que consiste en introducir valores intermedios en la secuencia de muestras de la que se dispone; por el contrario, diezmar se fundamenta en tomar uno de cada x valores de la secuencia.

Una vez construido el esquema detallado anteriormente ya se está en disposición de transmitir la señal, para ello se utilizará el bloque *Soapy HackRF Sink* que será el encargado de transmitir la señal mediante el HackRF. Existen otros bloques similares como el *Osmocom Sink* pero se empleará este ya que es más específico para este dispositivo. Se encuentra disponible a partir

de la versión 3.9.4 de GNU Radio.

Este bloque consta de una serie de parámetros de entrada a determinar. La frecuencia central se determinará en la frecuencia de transmisión deseada, en este caso se emplea un tipo de variable ya explicada previamente, *QT Gui Range*, para poder modificar dicha frecuencia según los requerimientos del espectro en el que se transmitirá. Resulta conveniente también fijar el control automático de ganancia a unos 16dB debido a que la señal utilizada. La tasa de muestreo adecuada para el dispositivo HackRF es de 2MHz por lo que se debe fijar a este valor.

Gracias a este esquema se conseguirá propagar la señal modulada a través del aire para posteriormente ser recibida en otra máquina a través de un dispositivo RTL-SDR para su recuperación.

6.3.2. Resultados de la recepción FM en GNU Radio

Tras haber transmitido la señal se recibe en otra máquina utilizando el dispositivo RTL-SDR. El objetivo es recuperar la señal original, para ello se plantea el esquema 6.19 de recepción y demodulación en GNU Radio.

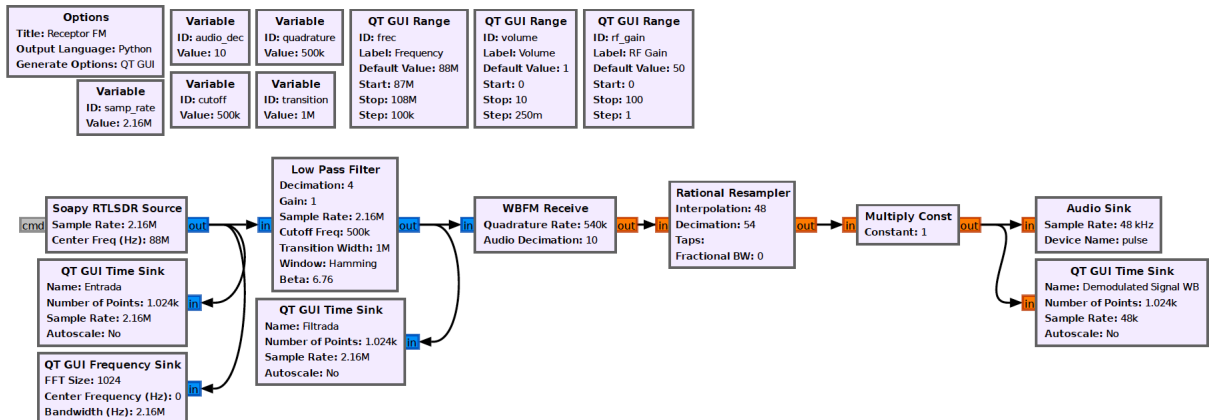
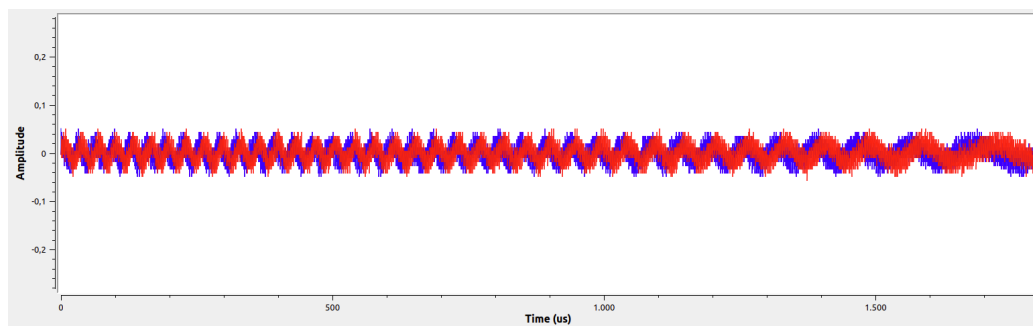


Figura 6.19: Esquema del receptor FM en GNU Radio

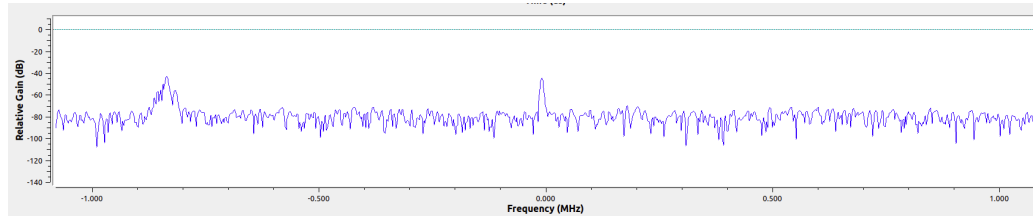
En primer lugar se sitúa el bloque *Soapy RTL-SDR Source* que será el encargado de recibir la señal a través del dispositivo RTL-SDR. En él se elige

la frecuencia de recepción deseada que deberá ser la misma que la elegida en transmisión. También se fija una tasa de muestreo *Sample Rate*; la adecuada para dicho dispositivo será de 2.16MHz.

Tras recibir la señal se visualiza la señal en tiempo 6.21 y en frecuencia 6.3.2. Si se observa el espectro de la señal se percibe como oscila como máximo unos $75kHz$, coincidiendo con la desviación máxima de frecuencia elegida en la modulación.



(a) Señal recibida en tiempo



(b) Espectro recibido

Figura 6.20: Entrada del receptor RTL-SDR

Como se observa en la señal recibida, existe una componente de ruido que es necesario limpiar previamente a la demodulación de la señal. Esto se conseguirá con el bloque *Low Pass Filter* que funciona como un filtro paso bajo. A dicho filtro se le asigna una frecuencia de corte de 500KHz y 1MHz de ancho de banda de transmisión. La señal de entrada tiene potencia suficiente por lo que no se considera necesario añadirle ganancia en el filtro. Se precisa utilizar una ventana *Hamming*, con su valor *beta* por defecto, por el decaimiento de sus lóbulos secundarios, esto hará que se desprecien las frecuencias que no

son de interés obteniendo así un mejor filtrado. En este bloque también se le aplica a la señal un diezmado de 4, es decir, se toma una de cada cuatro muestras generadas en el receptor.

Se observa en la figura 6.21 cómo ha desaparecido la mayor parte de las componentes de ruido que se observaba en la señal recibida gracias al filtrado de la señal.

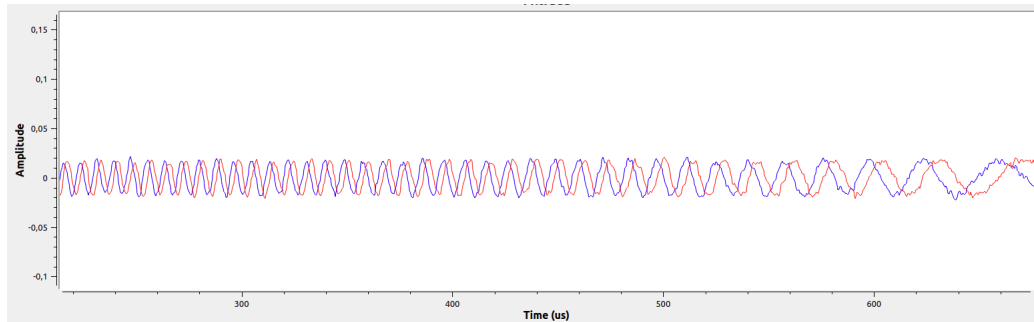


Figura 6.21: Señal filtrada mediante un filtro paso bajo

A continuación se utiliza el bloque *WBFM Receive* preparado para demodular una señal FM. Como se observa su entrada es compleja para obtener en su salida un flujo de muestras de tipo *float* que se corresponde con la señal demodulada. En él se definen una serie de parámetros de funcionamiento; *Quadrature Rate* será la tasa de muestreo de la señal y se fija a 540kHz y se aplica un diezmado de 10 para que la frecuencia de muestreo de la señal a la salida del demodulador sea de 54kHz . Se establece el diezmado, *Decimation*, a 10 debido a

Debido a que la tasa de muestreo a la salida del bloque explicado es de 500kHz es necesario modificarla para transformarla en 48kHz , la tasa necesaria para reproducir audio. Para ello se utiliza nuevamente el bloque *Rational Resampler* con un factor de interpolación de 48 y de diezmado de 50.

se desea conectar una salida de audio, por lo tanto la tasa de muestreo final de la señal deberá ser de 48kHz , para conseguirlo se utiliza el bloque *Rational Resampler* en el cuál se aplica un diezmado de 54 a la señal y interpola por 48.

Finalmente se añade un bloque *Multiply Const* cuyo parámetro es variable con el objetivo de poder modificar la ganancia durante el transcurso de la ejecución.

En la figura 6.22 se visualiza con éxito la señal senoidal de frecuencia $100Hz$ que se transmitió.

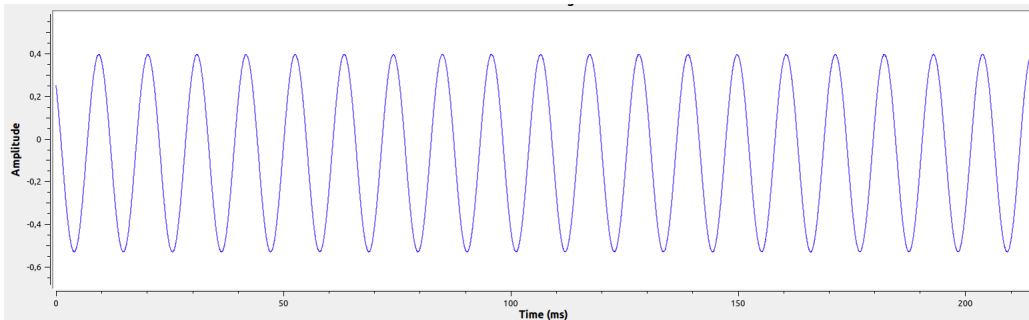


Figura 6.22: Señal demodulada FM

Cabe destacar que este esquema, puesto que sus tasas de muestreo se han enfocado a las compatibles con el audio, podría utilizarse para visualizar y escuchar señales FM comerciales. Basta con sintonizar una emisora cualquiera. A continuación, en la figura 6.23, se muestra un ejemplo de una señal proveniente de una emisora común.

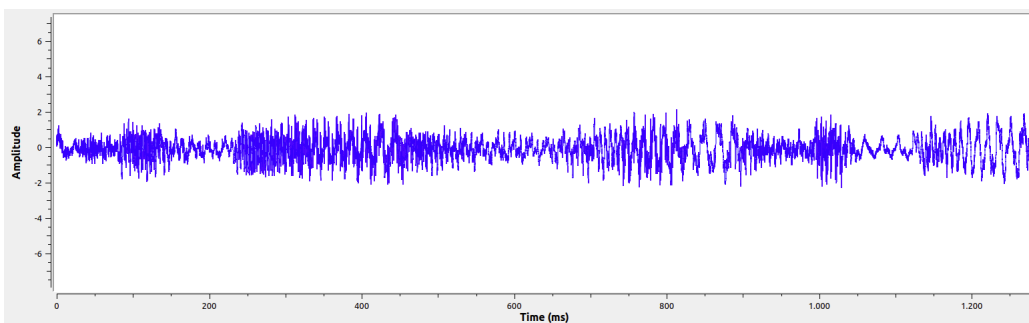


Figura 6.23: Señal demodulada FM proveniente de una emisora

Capítulo 7

Resultados de la transmisión y recepción digital

7.1. Introducción

Habiendo concluido las explicaciones teóricas de las modulaciones digitales en amplitud, frecuencia, fase y amplitud en cuadratura se procede a poner en práctica los conocimientos adquiridos para lograr transmitir y recibir señales analógicas utilizando el software GNU Radio. Al igual que en el capítulo anterior, para la transmisión de dichas señales se utilizará el Hack RF y para su recepción, el RTL-SDR.

7.2. Modulación digital en amplitud

7.2.1. Resultados de la transmisión ASK en GNU Radio

Para modular la señal en ASK con el objetivo de transmitirla se utilizará cómo se ha explicado anteriormente el software GNU Radio. El esquema utilizado para la transmisión será el mostrado en la figura 7.1.

En el esquema 7.1 se muestra un diseño sencillo partiendo de una señal cuadrada de frecuencia $1kHz$, amplitud uno y offset y fase inicial cero. Se muestra la señal generada en tiempo en la figura 7.2.

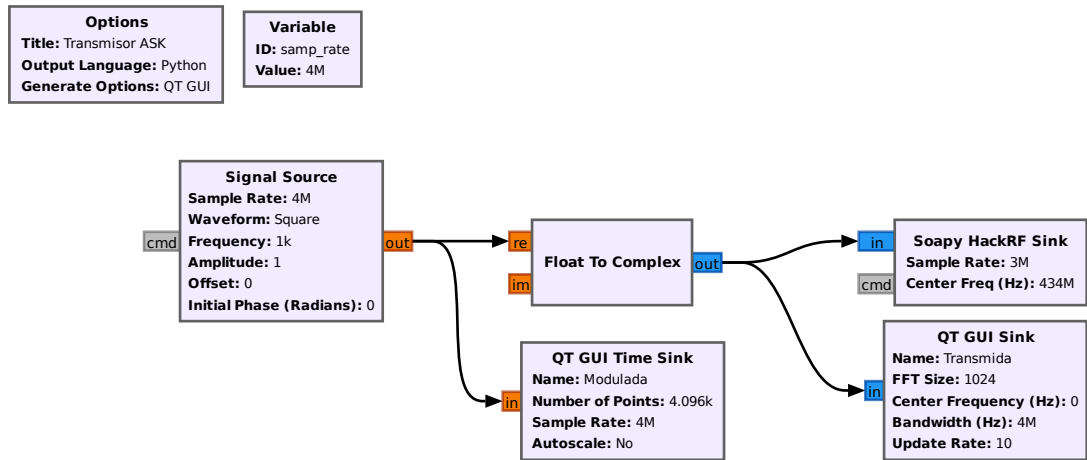


Figura 7.1: Esquema del transmisor ASK en GNU Radio

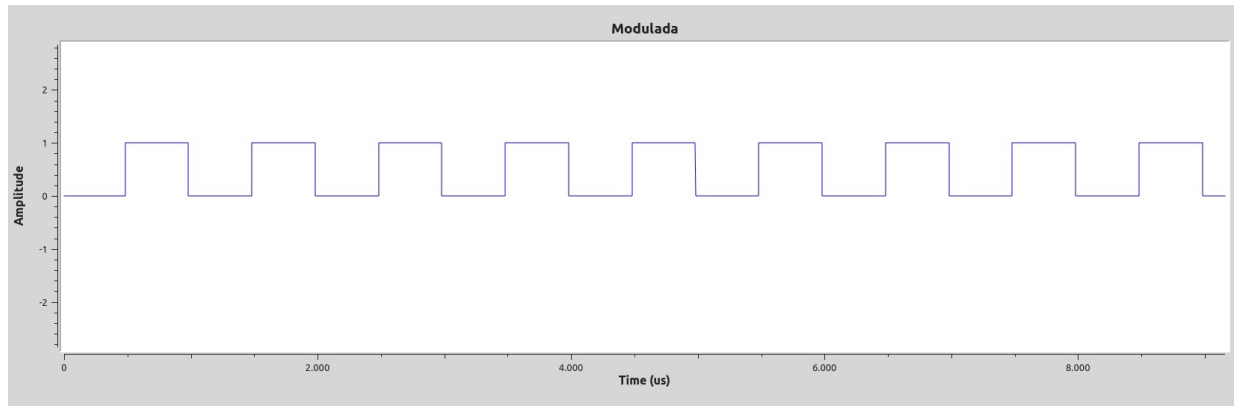


Figura 7.2: Señal generada en tiempo

Posteriormente, es necesario transformar la señal generada a complejos; para ello se hace uso del bloque *Float To Complex* en el cuál se inserta en la parte real la señal y se deja libre la parte imaginaria, lo que equivale a una secuencia de ceros. En la figura 7.3 se muestra el diagrama de constelación de dicha señal en el que se observa que la amplitud de esta es de 0 o 1 en función del símbolo a transmitir.

Una vez convertida la señal a complejos se estará en disposición de transmitirla mediante el HackRF; para ello se hace uso del bloque *Soapy HackRF*

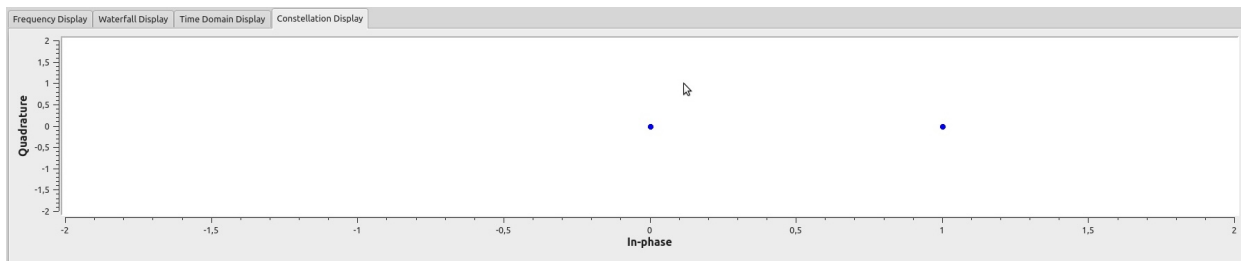


Figura 7.3: Constelación de la señal a transmitir ASK

Sink en el cuál se fija la frecuencia de muestreo en la variable *Sample Rate* a $3MHz$ y se elige una frecuencia de transmisión, *Center Freq*, libre que se fijará posteriormente para su recepción.

Resulta interesante destacar que a diferencia de la simulación en la que era necesario multiplicar esta señal por una portadora, ahora no será necesario ya que se utilizará la portadora derivada de la frecuencia de transmisión. Esta multiplicación es realizada de forma interna por el dispositivo HackRF.

7.2.2. Resultados de la recepción ASK en GNU Radio

Una vez transmitida la señal se procede a su recepción utilizando el dispositivo RTL-SDR. El bloque empleado para ello es el *Soapy RTLSDR Source*; en él se elige una tasa de muestreo soportada, $3,2MHz$ y la frecuencia de recepción elegida previamente en transmisión. Internamente, en el RTL-SDR se produce el proceso inverso al llevado a cabo en el HackRF, gracias a él, la señal portadora de $434MHz$ desaparece y en recepción se obtiene en consecuencia la señal mostrada en la figura 7.2.2.

La constelación asociada a esta señal es variable; se muestra un ejemplo de ella en la figura 7.5. Cabe destacar que en la transmisión solamente se han enviado dos símbolos por lo que el objetivo final será recuperar esos dos símbolos llevando a cabo una serie de pasos para demodular la señal.

El esquema empleado en GNU Radio para la demodulación de esta señal será el mostrado en la figura 7.6.

En él se observa que en primer lugar se emplea el bloque *Multiply Const*

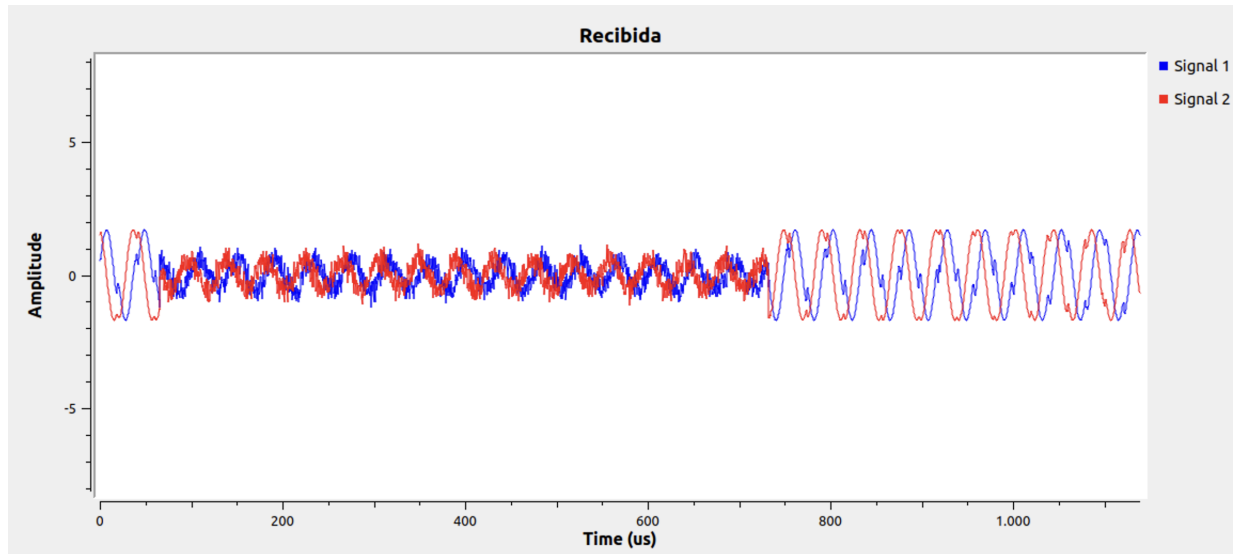


Figura 7.4: Señal recibida en tiempo

con una constante 4 para amplificar la señal una vez recibida. Posteriormente, gracias al bloque *Complex to Mag* se logra obtener una señal real que será el resultado de realizar el módulo de la señal compleja.

Habiendo obtenido una señal real, el paso siguiente es someterla a un umbral para que de esta forma se pueda distinguir entre que amplitudes de la señal corresponden al símbolo 1 y cuales al 0. Para fijar este umbral resulta conveniente fijarse en la señal recibida para encontrar un valor intermedio que de la mayor veracidad posible a los símbolos elegidos. En ella se puede observar como cuando el símbolo recibido corresponde a un 0, los valores de la señal no superan una amplitud de 1.2. Por medio del bloque denominado *Threshold* en GNU Radio, se fija un umbral inferior, *Low*, de amplitud 1 y otro superior, *High*, cuya amplitud será 1.4. Será este umbral el que actúe de decisor y fije si la señal corresponde a un 1 o un 0 lógicos.

Gracias a lo explicado anteriormente se obtienen las siguientes constelaciones, mostradas en la figura , tomadas en instantes de tiempo diferentes. En la primera de ellas se observa cómo el símbolo recibido ha sido un 0 lógico y en la segunda un 1 lógico.

Finalmente se muestra la señal demodulada 7.8 y se observa cómo coin-

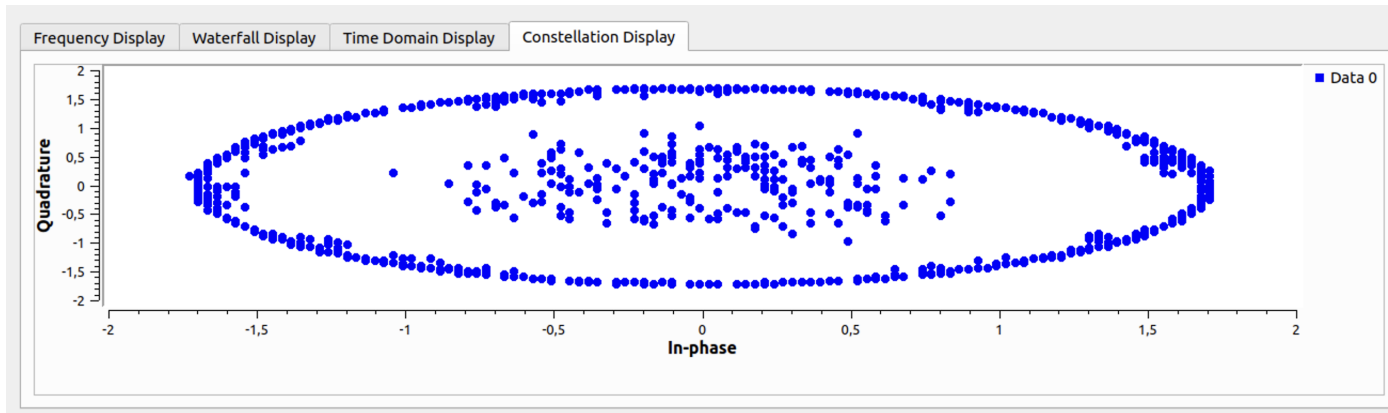


Figura 7.5: Constelación de la señal recibida

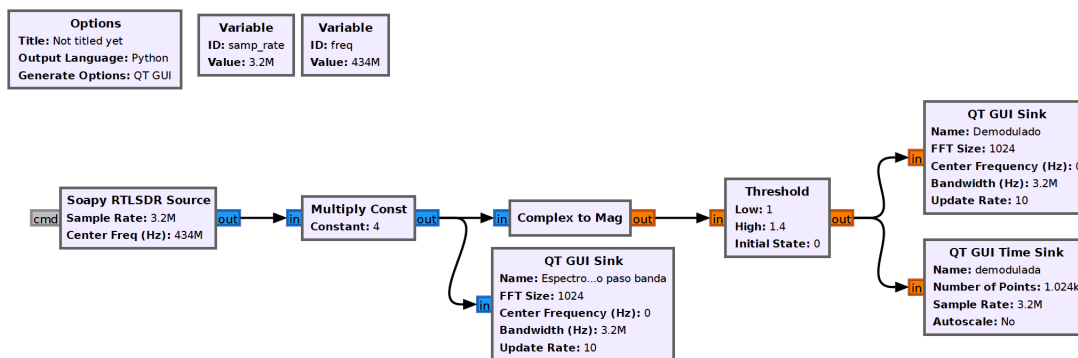
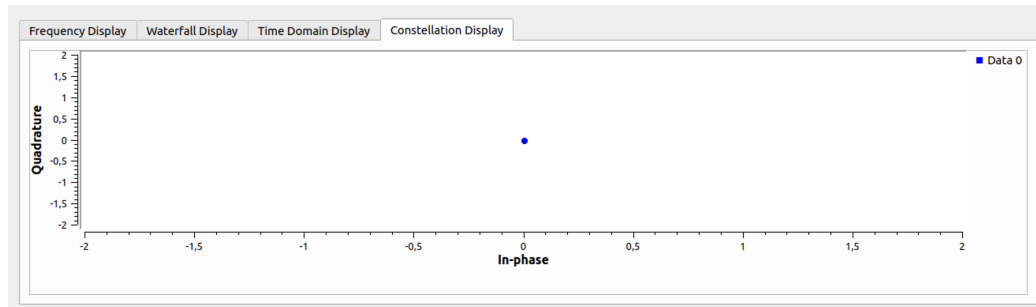


Figura 7.6: Esquema del receptor ASK en GNU Radio

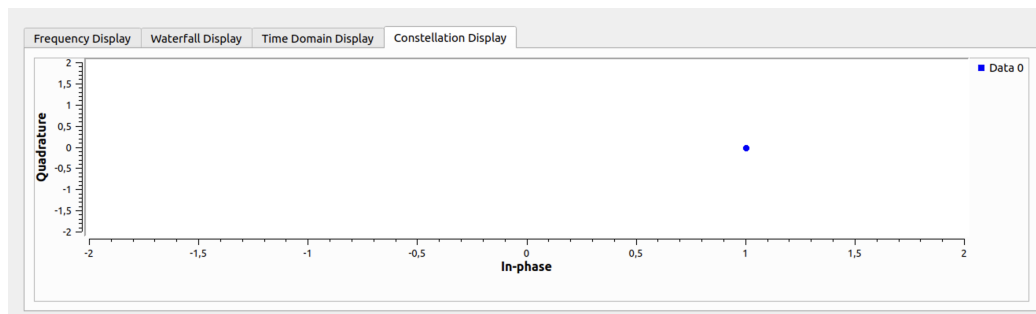
cide con la transmitida como se deseaba.

Una vez explicada la transmisión y recepción ASK caben diferentes dudas al respecto. Observando la señal recibida y habiendo mencionado que el receptor eliminaba la portadora añadida previamente en el transmisor, resulta difícil comprender el origen de las oscilaciones observadas en la figura 7.2.2 ya que esta señal teóricamente debería ser la existente en el transmisor antes de multiplicarse por la portadora.

Como se explicó en el capítulo 2, al transmisor, en este caso el HackRF, llegan dos flujos de muestras, uno real y otro imaginario; haciendo uso de



(a) Constelación correspondiente al símbolo 0



(b) Constelación correspondiente al símbolo 1

Figura 7.7: Constelaciones obtenidas en la recepción ASK.

mezcladores, el flujo real es multiplicado por $\cos(2\pi * f_c * t)$ mientras que el imaginario se multiplica por $\sin(2\pi * f_c * t)$, siendo f_c la frecuencia de la portadora elegida. Si en el receptor es llevado a cabo el proceso contrario, las oscilaciones mencionadas anteriormente solo pueden entenderse como señales espúreas, espaciadas en este caso alrededor de $26kHz$ de la frecuencia de la portadora. El origen de estas señales es en su mayoría el ruido interno que se produce en el receptor.

Para evitar la aparición de las señales espúreas se prueba a disminuir ligeramente (en torno a los $26kHz$ ya mencionados) la frecuencia de recepción; gracias a lo cuál se logra captar la señal mostrada en la figura 7.9.

De esta manera, bastaría simplemente con tomar la parte real de la señal para pasarla por el detector de umbral que seguiría siendo necesario para poder limpiarla de ruidos y conseguir una señal cuadrada perfecta.

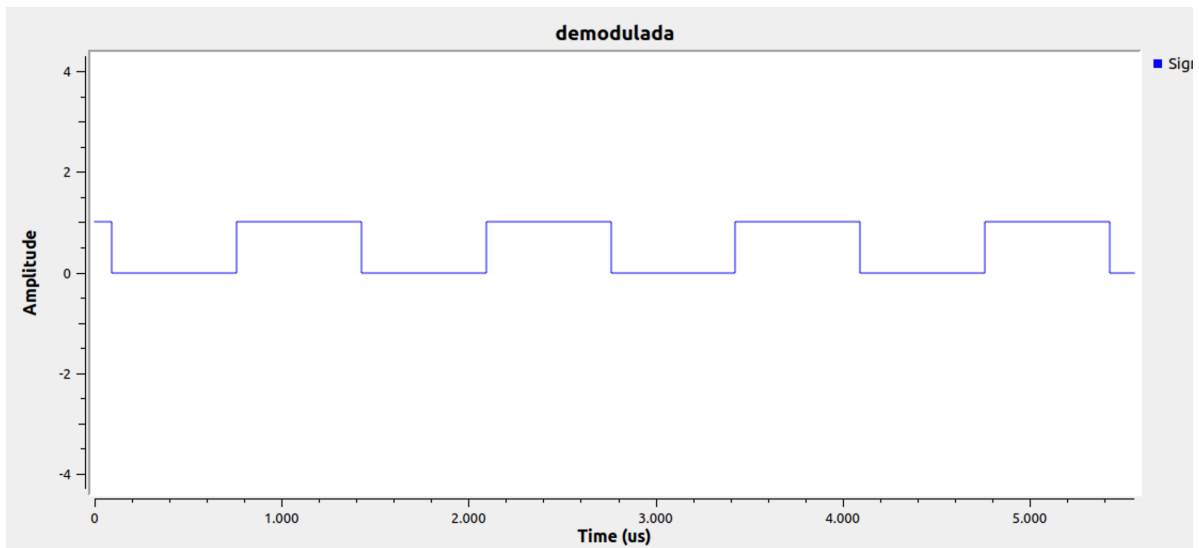


Figura 7.8: Señal demodulada ASK en recepción

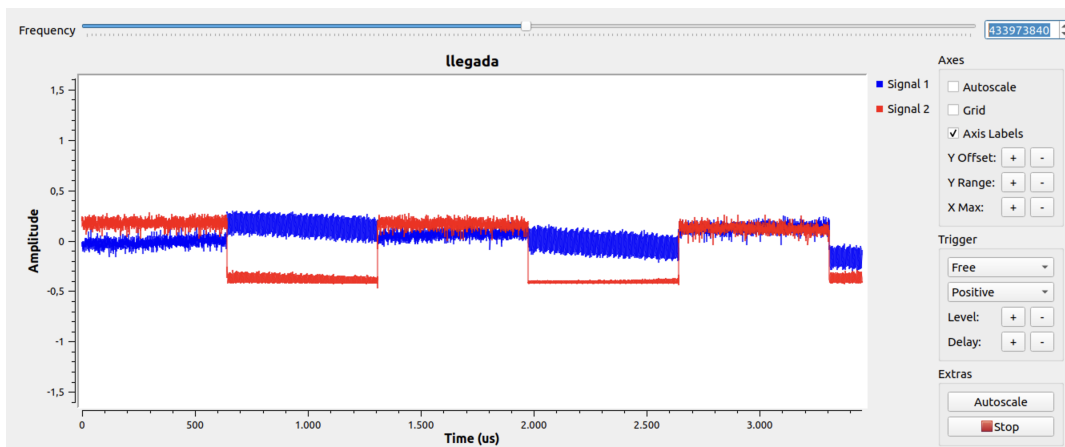


Figura 7.9: Señal recibida en tiempo disminuyendo la frecuencia del receptor

Por otra parte, resulta interesante profundizar en el por qué del uso del bloque *Complex to Mag* cuando sintonizamos el receptor a la frecuencia del transmisor. Volviendo a la figura de nuevo, las oscilaciones producidas por el receptor son la causa. Si en vez de tomar el módulo de la señal se tomase únicamente la parte real o la imaginaria, el resultado de la señal demodulada

sería el mostrado en la figura 7.10.

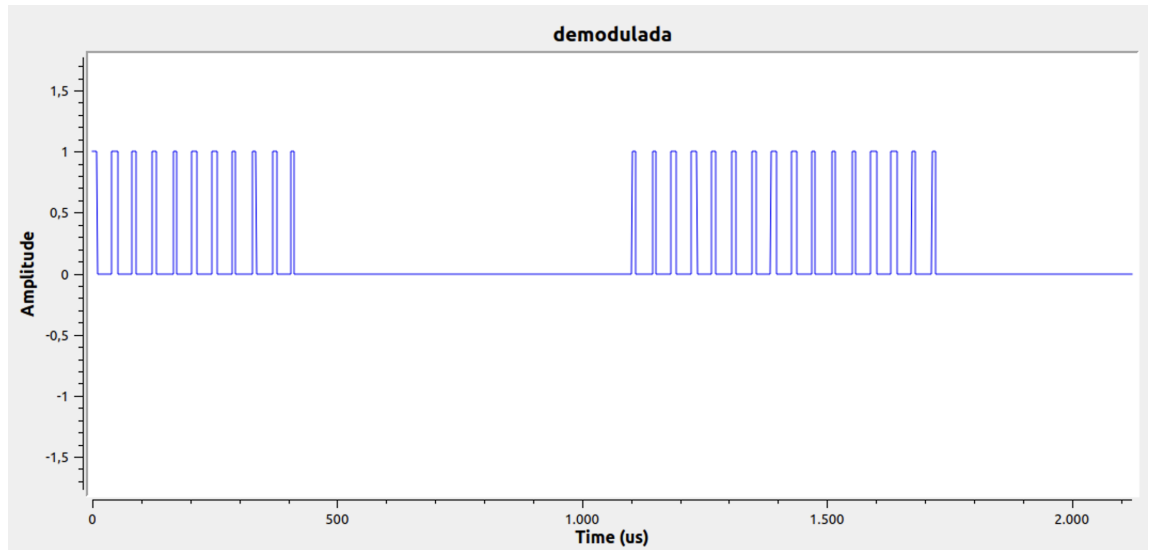


Figura 7.10: Señal demodulada ASK tomando la parte real

El cálculo del umbral se realizaría para cada una de las oscilaciones de la señal recibida y en consecuencia, la señal demodulada no se correspondería con la señal cuadrada de $10kHz$ que se transmitió. Se justifica así la necesidad del cálculo de su módulo.

7.3. Modulación digital en fase

7.3.1. Resultados de la transmisión PSK en GNU Radio

Para modular la señal en PSK con el objetivo de transmitirla se utilizará cómo se ha explicado anteriormente el software GNU Radio. El esquema utilizado para la transmisión será el mostrado en la figura 7.11. En el esquema 7.11 se muestra un diseño de un transmisor PSK. Como se observa, se parte de un vector cuyo valor será una secuencia compuesta por los números 0, 1, 2 y 3. A continuación se utiliza el bloque *Constellation Modulator* que permite

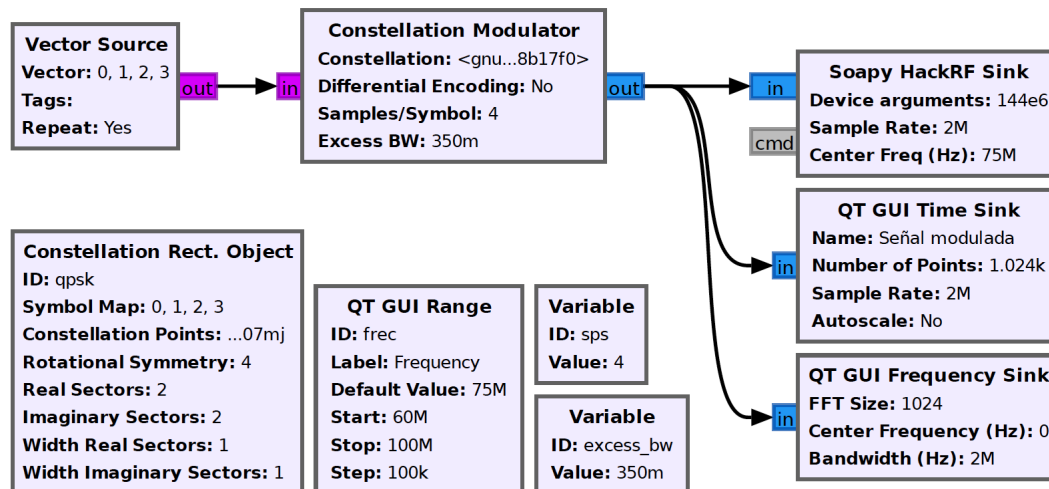
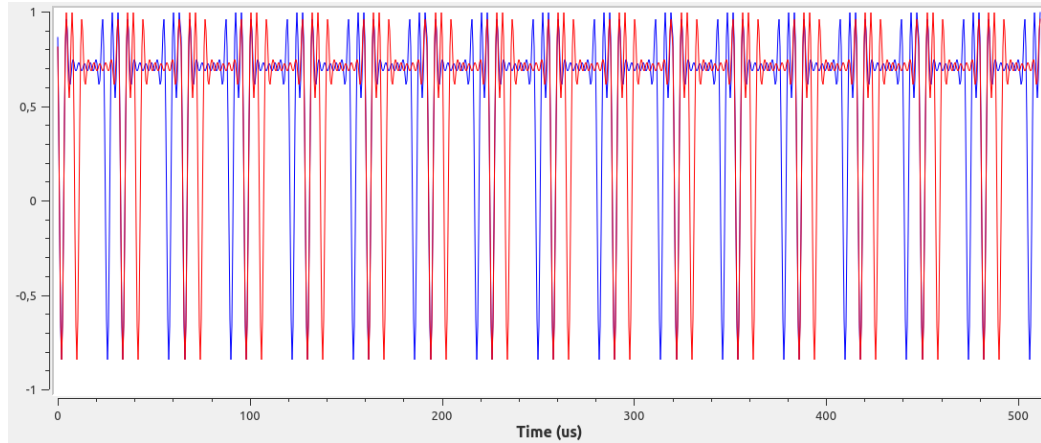


Figura 7.11: Esquema del transmisor PSK en GNU Radio

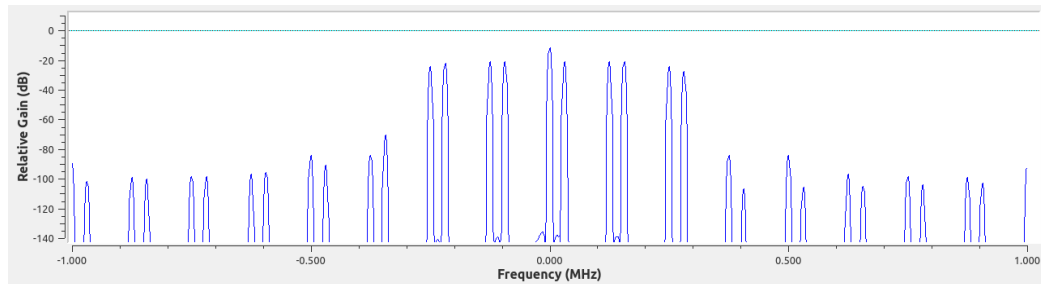
la modulación de símbolos digitales a través de una constelación definida [12]. En él se fijan sus parámetros:

- Muestras por símbolo (*Samples/symbol*): Define cuántas muestras por símbolo se utilizan para muestrear la señal modulada; en este caso se fija a 4 muestras/símbolo.
- Codificación diferencial (*Differential Encoding*): La codificación diferencial se basa en transmitir la diferencia entre el símbolo actual y el símbolo anterior. En este caso se decide no utilizar este tipo de codificación para simplificar el proceso.
- *Excess Bandwidth*: Se utiliza para controlar el rolloff del filtro de raíz elevada utilizado para dar forma a la señal antes de ser transmitida. Su valor se fija comúnmente a 0.35 para este tipo de constelaciones.
- Puntos de la constelación (*Constellation Points*): Para definir los valores complejos de los símbolos de la constelación que se desea utilizar se define una variable del tipo *Constellation Rect. Object*. En ella se definen los puntos que son los valores complejos de los símbolos que conforman la constelación; en este caso se desean transmitir los símbolos 0, 1, 2 y 3 y sus valores serán $0, 707 + 0, 707j$, $-0, 707 + 0, 707j$, $-0, 707 - 0, 707j$ y $0, 707 - 0, 707j$ respectivamente.

Como se observa, el proceso es idéntico al utilizado en la simulación y gracias a él se obtiene la siguiente señal a transmitir 7.12.



(a) Señal modulada PSK en tiempo



(b) Señal modulada PSK en frecuencia

Figura 7.12: Señal transmitida en la modulación PSK.

7.3.2. Resultados de la recepción PSK en GNU Radio

El proceso de recepción y demodulación de una señal modulada PSK es complejo debido a diversos factores que se describirán a lo largo de esta sección. Para favorecer la explicación se dividirá en varias fases.

En primer lugar será necesario recibir la señal utilizando nuevamente el dispositivo RTL-SDR y realizar un filtrado de la misma como se muestra en el esquema 7.13.

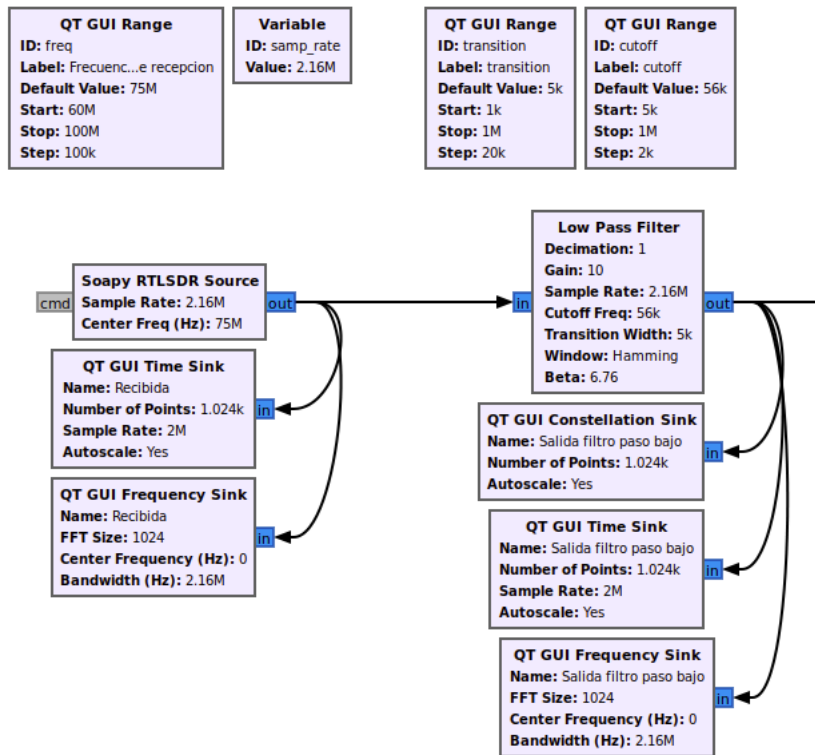
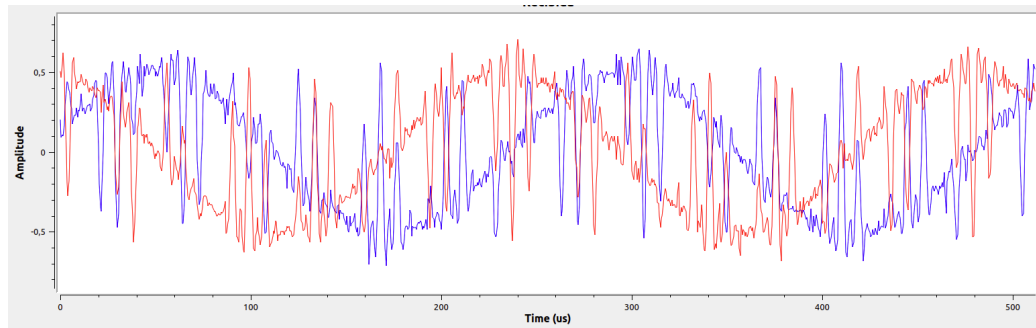


Figura 7.13: Esquema PSK en GNU Radio: Recepción y filtrado

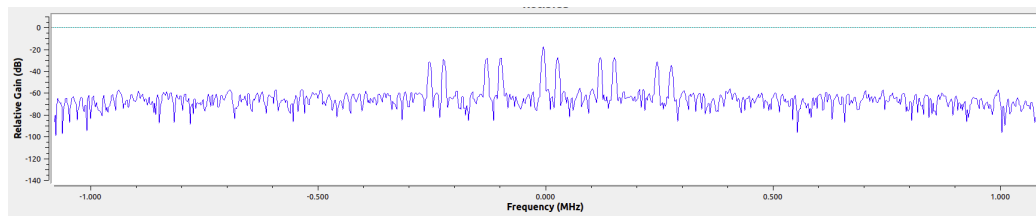
Para ello se utiliza el bloque *Soapy RTLSDR Source* en el que se fija la tasa de muestreo a la requerida por el dispositivo, 2.16MHz . La señal recibida se observa en la figura 7.14.

Como se visualiza, la señal recibida sufre distorsión, por ello es necesario filtrarla utilizando un filtro paso bajo para lo cual se utiliza el bloque *Low Pass Filter*. Teniendo en cuenta el espectro de la señal en recepción 7.14 se fija la frecuencia de corte del filtro en este caso a 280kHz . De esta forma se obtiene la señal filtrada 7.15.

Una vez filtrada la señal, será necesario realizar una sincronización de los tiempos de los símbolos en el demodulador, en este caso se empleará una técnica de recuperación de reloj basada en un banco de filtros polifásicos



(a) Señal recibida PSK en tiempo



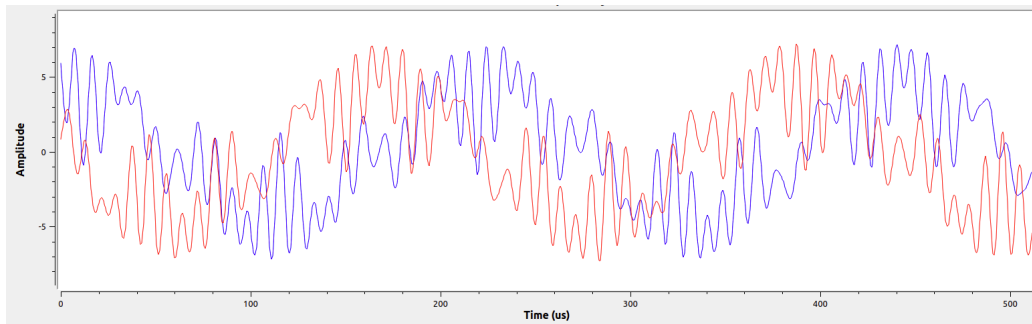
(b) Señal recibida PSK en frecuencia

Figura 7.14: Señal recibida PSK.

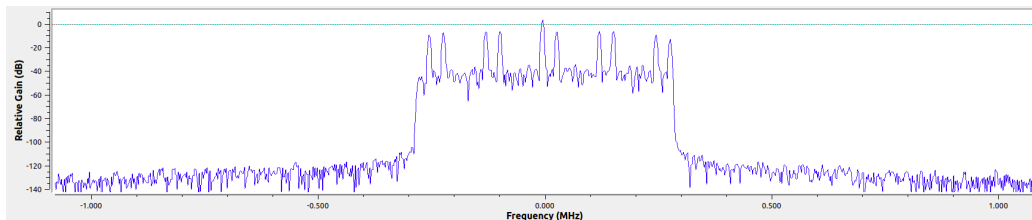
[21]; para ello se utiliza el bloque de GNU Radio *Polyphase Clock Sync* como se muestra en la figura 7.16.

Este bloque realiza en primer lugar una recuperación del reloj para posteriormente aplicar un filtro que permitirá resolver el problema de la interferencia entre símbolos. En él se pueden ajustar las siguientes variables que determinarán el funcionamiento del bloque.

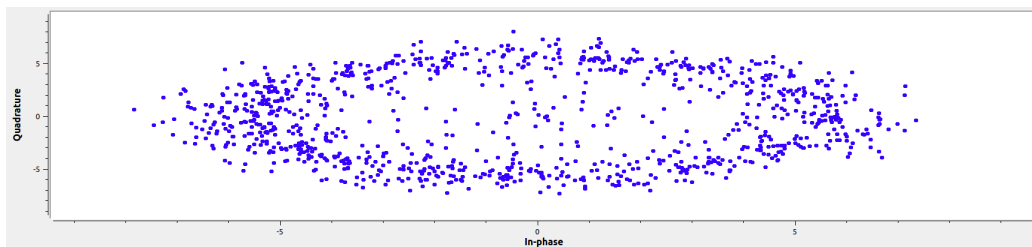
- *Samples/Symbol*: Número de muestras por símbolo a la entrada. En este caso 4 al igual que en transmisión.
- *Loop Bandwidth*: El ancho de banda de realimentación. Se utiliza para establecer la ganancia del lazo interno de realimentación. Debe de ser un valor pequeño de alrededor de $2\pi/100$, el cual corresponde con una subdivisión de la circunferencia unitaria en 100 trozos (en radianes) y determina el tamaño del paso para ajustarse al error.
- *Taps*: Uno de los parámetros más importantes de este bloque son los coeficientes (taps) del filtro. Uno de los beneficios de implementación



(a) Señal a la salida del filtro paso bajo en tiempo



(b) Señal a la salida del filtro paso bajo en frecuencia



(c) Constelación a la salida del filtro paso bajo

Figura 7.15: Señal a la salida del filtro paso bajo.

del bloque es que se puede definir el filtro acoplado directamente en este bloque, lo que permite tener el filtro acoplado y la corrección de tiempo de muestreo de manera simultánea. Los taps corresponden al filtro acoplado, que para una modulación digital típica es del tipo raíz de coseno alzado. El número de taps de este filtro está basado en la longitud del enlace o canal canal que genera la dispersión temporal, esto se traduce en cuántos símbolos se quieren combinar para obtener la energía de los símbolos de vuelta. Una vez diseñado el filtro acoplado es necesario tener en cuenta que ese filtro tendrá un desfase diferente en

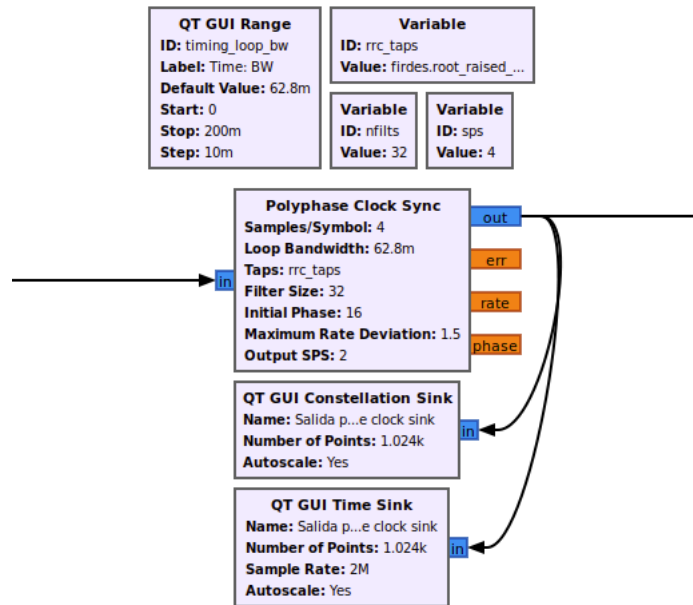


Figura 7.16: Esquema PSK en GNU Radio: Polyphase Clock Sync

cada filtro del banco, por lo que ese número debe de ser multiplicado por el número de filtros del banco (variable n_{filts}). El producto de estos dos valores tendrá como resultado el número de taps del filtro acoplado a utilizar. Al utilizar el banco de filtros, este segmenta estos taps dentro del banco de forma que cada filtro tendrá una fase diferente equiespaciadas $2\pi/N$, donde N es el número de filtros del banco.

- *Filter size*: Este parámetro hace referencia al número de filtros y su valor por defecto es 32. Con este valor se obtiene una buena resolución en la fase tal que el ISI producido es prácticamente imperceptible. Utilizar 64 filtros puede reducir más el ISI, pero incrementa significativamente la complejidad del bloque.
- *Initial Phase*: Gracias a este parámetro es posible fijar la fase inicial, en este caso será el número de filtros impuesto anteriormente entre 2.
- *Maximum Rate Deviation*: Su valor por defecto es 1,5. Este parámetro define la desviación máxima permitida de la frecuencia con la que se

viaja por los filtros para recoger los errores. Limitar esta tasa puede ayudar a mantener el algoritmo dando pasos desde muy lejos hasta el enganche durante periodos en que no haya señal.

- *Output SPS*: Se utiliza para fijar el número de muestras por símbolo a la salida del bloque. Se fija en este caso a 2 muestras por símbolo debido a que los ecualizadores modelan mejor el canal con este valor.

La constelación obtenida a la salida de este bloque se puede ver en la figura 7.17.

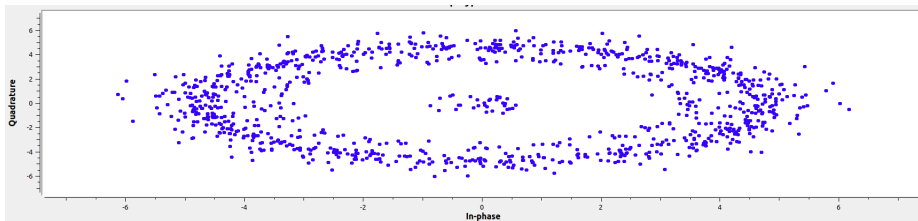


Figura 7.17: Constelación a la salida del bloque Polyphase Clock Sync

A continuación se emplea el bloque *Linear Equalizer* para mitigar los efectos de la distorsión y la interferencia de la señal recibida [21] y se puede observar en el esquema de la figura .

Su función principal es actuar como un ecualizador lineal adaptativo para corregir las distorsiones introducidas en la señal transmitida durante su paso a través de un canal de comunicación.

En él se fijan sus principales parámetros:

- *Num. Taps*: Especifica la cantidad de taps o coeficientes del filtro que se utilizan en el ecualizador. Más taps pueden proporcionar una corrección más precisa, pero también pueden requerir más recursos computacionales. En este caso se fija a 15.
- *Input Samples per Symbol*: Se utiliza para fijar el número de muestras por símbolo a la entrada del bloque. Se fija en este caso a 2 muestras por símbolo debido a que es el valor que introducido en el *Polyphase Clock Sync* .

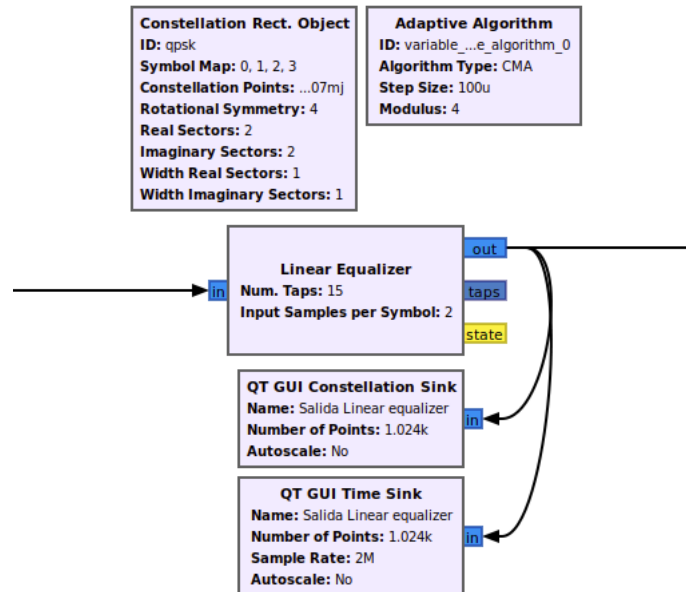


Figura 7.18: Esquema PSK en GNU Radio: Linear Equalizer

- *Adaptative Algorithm Object*: En él se define el algoritmo a utilizar. Para ello se define una variable de tipo *Adaptative Algorithm* en la que se selecciona el algoritmo CMA y la constelación diseñada para la transmisión. El algoritmo CMA utiliza el criterio de módulo constante como base para su funcionamiento. Esto significa que asume que la señal transmitida tiene un módulo constante bajo ciertas condiciones. Esto es aplicable a muchas señales moduladas digitalmente y es muy útil para este caso ya que todos los símbolos de la modulación PSK tienen igual módulo.

De esta forma el equalizador utiliza un filtro adaptativo que se ajusta continuamente en función de la señal de entrada y la señal de salida observada. Los coeficientes del filtro se actualizan iterativamente para minimizar la diferencia entre la señal de salida del filtro y la señal de referencia (la señal original sin distorsión).

La constelación resultante a la salida de este bloque se observa en la fi-

gura 7.19. En ella se perciben como los símbolos han recuperado su módulo original aunque aún no han recuperado su posición.

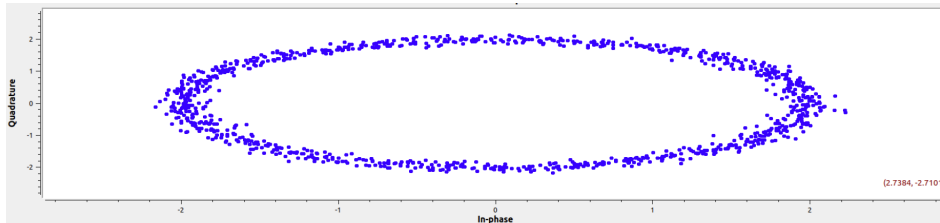


Figura 7.19: Constelación a la salida del bloque Linear Equalizer

El bloque *Linear Equalizer*, aunque es muy útil para disminuir la distorsión, provoca una rotación de fase, por ello es necesario utilizar el bloque *Costas loop* ya que ayudará en la sincronización de fase y frecuencia. En la figura se observa el esquema correspondiente en GNU Radio.

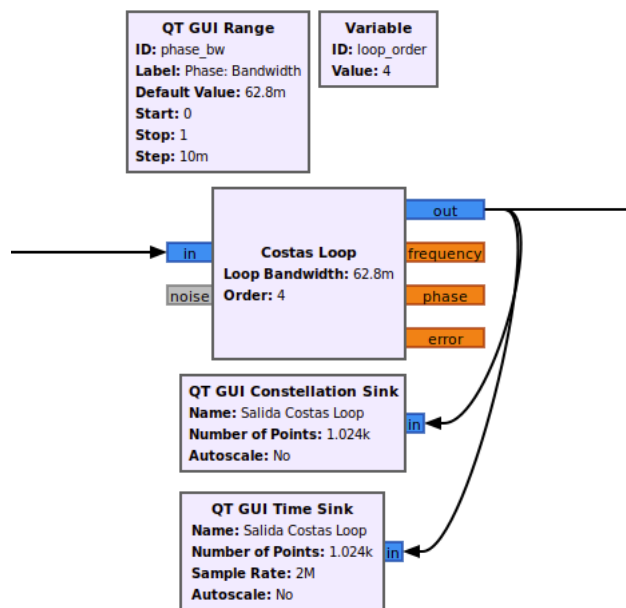


Figura 7.20: Esquema PSK en GNU Radio: Costas loop

Este bloque realiza un seguimiento continuo de la fase y la frecuencia de la señal de entrada con el fin de recuperar la información modulada en fase de manera precisa. Para demodular la señal, el bloque genera una señal de referencia conocida como 'lazo de seguimiento de fase' o 'lazo de bucle de Costas'. Esta señal de referencia tiene la misma frecuencia que la señal modulada, pero su fase se ajusta continuamente para rastrear la fase de la señal de entrada. Posteriormente compara la fase de la señal de entrada con la fase de la señal de referencia. Si hay una diferencia de fase, el bloque ajusta la fase de la señal de referencia para reducir esa diferencia. Este proceso de corrección de fase es esencial para mantener la coherencia con la señal de entrada.

A medida que la fase de la señal de referencia se ajusta para seguir la fase de la señal de entrada, se obtiene una señal demodulada en la que la información de datos original se ha recuperado. Esta señal se procesará posteriormente para recuperar los datos originales.

En él se fijan los siguientes parámetros:

- *Loop Bandwidth*: En esta variable se fija el ancho de banda del filtro utilizado en el lazo de seguimiento de fase. Su valor se configura variable para que sea posible modificarlo durante la ejecución del programa.
- *Order*: Se utiliza para fijar el orden del bucle, en este caso 4.

La constelación resultante a la salida de este bloque tras haber ajustado sus parámetros durante la modulación se muestra en la figura 7.21. En ella se observa cómo se ha conseguido recuperar la fase original de la señal ya que los símbolos se aproximan a su valor original.

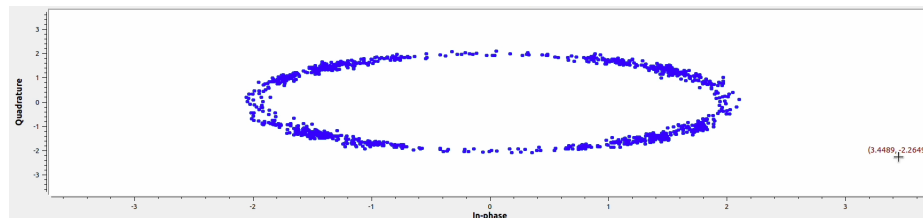


Figura 7.21: Constelación a la salida del bloque Costas loop

En la última fase del diseño de este demodulador PSK, se utilizarán los bloques empleados en la simulación ya que en este momento se dispone de una señal sincronizada en tiempo, frecuencia y fase con la transmitida. El esquema empleado para recuperar los símbolos originales será el de la figura 7.22.

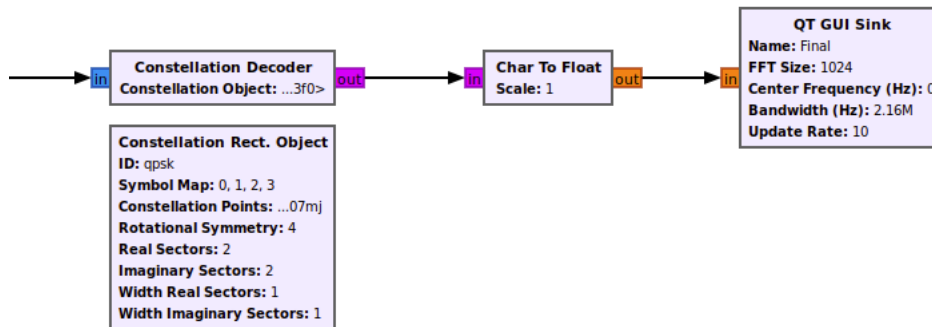


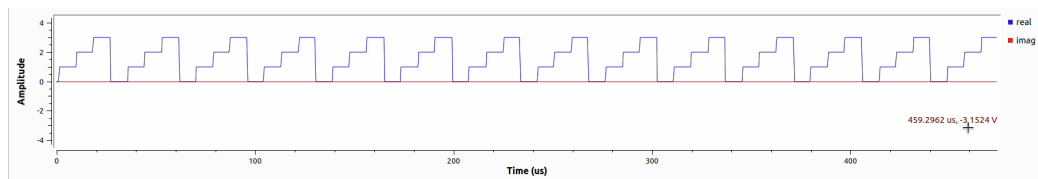
Figura 7.22: Esquema PSK en GNU Radio: Decodificación de la señal

Al igual que en la simulación se utiliza el bloque *Constellation Decoder* que se encarga de decodificar los símbolos complejos de la constelación en su forma original. En él se introduce la misma constelación que se diseñó para la transmisión y que ya ha sido utilizada en el diseño del algoritmo CMA.

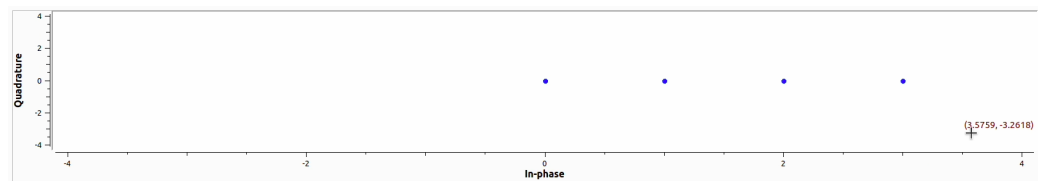
Gracias al bloque *Char To Float* se convierte el flujo de bytes en una secuencia de floats obteniéndose finalmente la señal demodulada que se muestra en la figura 7.23.

Como se observa, la señal demodulada es idéntica al vector utilizado en transmisión. Resulta muy difícil la recuperación de la señal ya que receptor y transmisor deben estar completamente sincronizados en tiempo, fase y frecuencia. La complejidad de la sincronización aumenta debido a la inestabilidad en frecuencia tanto del transmisor como del receptor.

En la figura 7.24 se muestra el esquema completo utilizado para la demodulación.



(a) Señal demodulada PSK en tiempo



(b) Constelación PSK demodulada

Figura 7.23: Señal demodulada PSK.

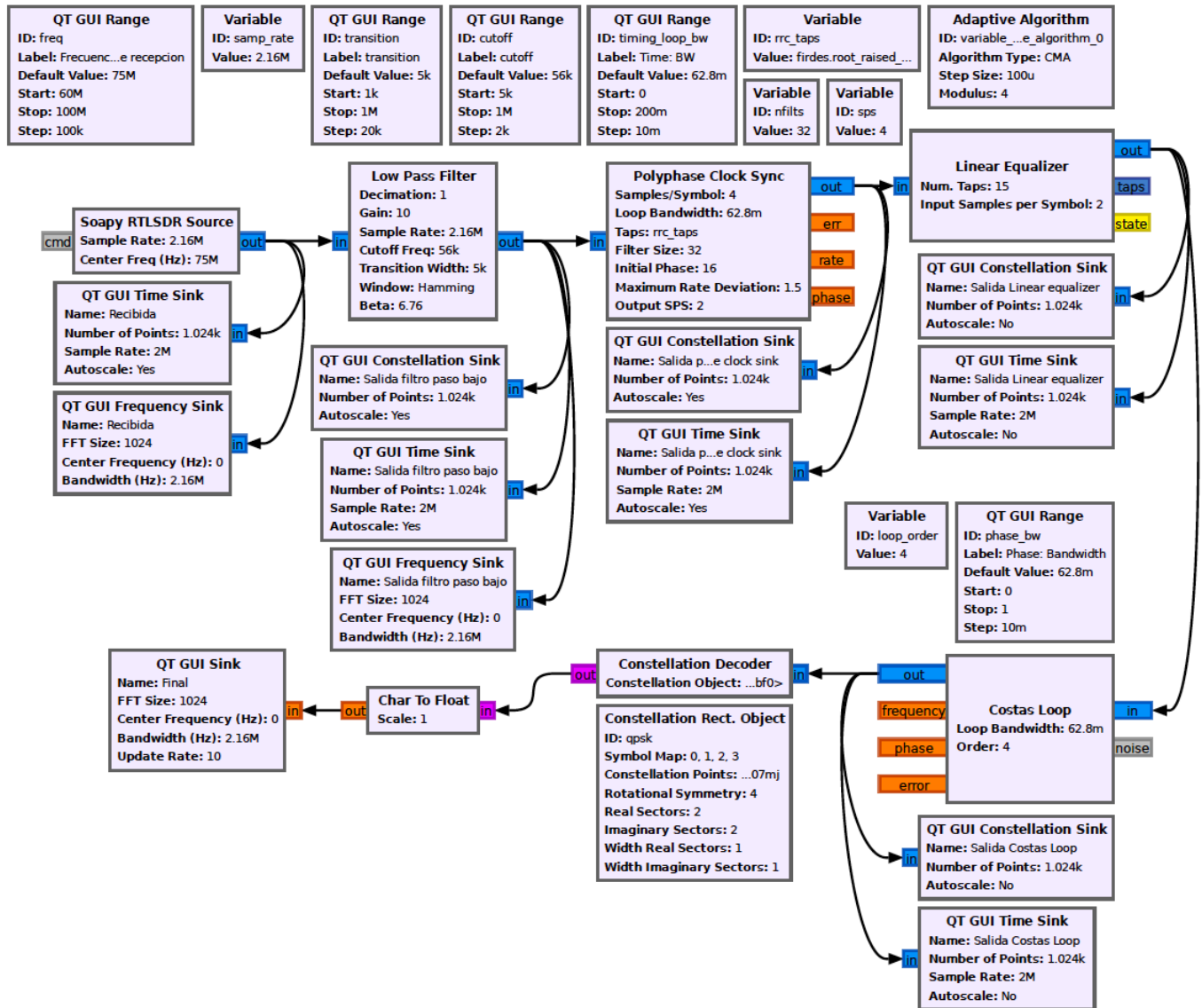


Figura 7.24: Esquema completo para la demodulación PSK en GNU Radio

Capítulo 8

Conclusiones y líneas futuras

8.1. Conclusiones

Durante el desarrollo de este proyecto de radiocomunicaciones con radio definida por software (SDR) y GNU Radio, se ha llegado a importantes conclusiones que resaltan la versatilidad y el potencial de esta tecnología.

A lo largo del trabajo se han explorado algunas de las múltiples opciones y configuraciones de las que dispone GNU Radio. Gracias a su extraordinaria configurabilidad resulta factible implementar infinitos sistemas de comunicaciones lo que resulta muy útil a la hora de realizar experimentos en tiempo real y de desarrollar prototipos de sistemas de comunicaciones ya que ha permitido personalizar cada aspecto de los sistemas de comunicación, desde la modulación y el filtrado hasta la demodulación y el procesamiento de señales. Esto ha permitido adaptar las soluciones a los requisitos específicos. Su capacidad para modificar parámetros en tiempo real ha sido esencial para optimizar el rendimiento en diferentes entornos.

También la capacidad de ajustar el ancho de banda en tiempo real ha sido crucial para lograr una mayor eficiencia espectral. Esto es fundamental en aplicaciones donde el espectro radioeléctrico es un recurso limitado y valioso.

Los mayores impedimentos a lo largo de la realización de este trabajo han tenido su origen en la calidad del hardware utilizado; esto ha afectado principalmente en los experimentos llevados a cabo para la transmisión y recepción

PSK ya que para lograr recuperar la señal original es necesario que receptor y transmisor se encuentren sincronizados en tiempo, frecuencia y fase.

Resulta muy complicado realizar una sincronización en frecuencia debido a que tanto el hardware utilizado para la transmisión, Hack RF, como el utilizado para recepción, RTL-SDR, son muy inestables en frecuencia y esto supone problemas cuanto más compleja y sensible es la modulación. Todo ello sumado al ruido ambiental, de cuantificación, atmosférico y a las interferencias han hecho difícil el camino para obtener los resultados esperados.

Los experimentos que se muestran a lo largo de esta memoria son los que han sido satisfactorios, pero para llegar a ellos se han afrontado varios desafíos técnicos como se ha explicado, la sincronización de múltiples SDR, la mitigación del ruido y la corrección de errores en las transmisiones. Estos desafíos subrayan la importancia de un profundo conocimiento técnico y la necesidad de soluciones ingeniosas en el campo de la radio definida por software.

Todos los experimentos llevados a cabo han permitido adquirir habilidades valiosas en la implementación de sistemas de comunicación y procesamiento de señales en tiempo real.

Este trabajo ha ayudado a apreciar la importancia de la investigación continua en este campo. Las tecnologías de radio definida por software están en constante evolución, y el conocimiento actualizado es esencial para mantenerse al día con los avances y las oportunidades emergentes.

Por último, cabe destacar el valor de la comunidad de SDR y GNU Radio. La colaboración de investigadores y entusiastas en foros en línea, grupos de usuarios y conferencias ha sido muy enriquecedora.

8.2. Líneas futuras

A lo largo del trabajo, y así se ha plasmado en la memoria, la secuencia llevada a cabo para la realización de los diversos experimentos que aquí se recogen ha sido siempre la misma.

En primer lugar se ha procedido al estudio teórico de una serie de modulaciones escogidas para posteriormente efectuar la simulación de cada una de ellas, empezando por las más sencillas para progresivamente ir aumentando la dificultad.

Las modulaciones analógicas escogidas han sido simuladas y se han conseguido obtener los resultados esperados tanto en AM como en FM. Esto no sucede para las digitales ya que por cuestiones de tiempo y de extensión se han focalizado los esfuerzos en la transmisión y recepción ASK y PSK.

Sería interesante la ampliación de este trabajo utilizando los conocimientos que aquí se presentan para realizar experimentos utilizando también las modulaciones FSK y QAM.

Observando los resultados obtenidos para la transmisión y recepción PSK, cabe intuir que la modulación QAM será muy similar a esta ya que será necesario realizar una sincronización en tiempo, frecuencia y fase previa a la decodificación de la constelación.

Otra de las vías interesantes para la ampliación de este trabajo podría ser la investigación para lograr implementaciones que permitan el aumento de la distancia entre transmisor y receptor. Como se observa en la imagen 8.1, la distancia entre ambos dispositivos debe ser lo más reducida posible lo que imposibilita el sistema para su uso en aplicaciones reales. También sería de gran utilidad conseguir mejorar la comunicación aumentando su sincronización.

Una de las posibles soluciones que se podrían adoptar y que probablemente corregirían los problemas anteriores sería el uso de hardware más potente y estable, como el USRP cuyas características se explican en el capítulo 2. Además este dispositivo permite la creación de sistemas MIMO (Entrada múltiple, Salida Múltiple) lo que en la práctica posibilitaría la creación de protocolos de comunicación, gracias a los cuales se podría confirmar si la información recibida es correcta para poder detectar errores en la comunicación y solventarlos.

Gracias a ello se podría implementar un sistema capaz de transmitir correctamente un fichero utilizando modulaciones digitales.



Figura 8.1: Transmisor y receptor

Además de su impacto en la investigación y la industria, se ha identificado un potencial significativo de SDR y GNU Radio en la educación. Estas herramientas permiten una enseñanza más práctica y dinámica de la teoría de la comunicación, la electrónica y el procesamiento de señales, preparando a los estudiantes para desafíos tecnológicos del mundo real.

Bibliografía

- [1] Ian Poole. *Newnes Guide to Radio and Communications Technology*. Elsevier Ltd., Linacre House, Jordan Hill, Oxford OX2 8DP 200 Wheeler Road, Burlington, MA 01803, 2003.
- [2] Joseph Mitola III. *Cognitive Radio Architecture: The Engineering Foundations of Radio XML (English Edition)*. 1992.
- [3] RTL-SDR. Review: Airspy vs. sdrplay vs. hackrf, . URL <https://www.rtl-sdr.com/review-airspy-vs-sdrplay-rsp-vs-hackrf/>.
- [4] Di Pu Travis F. Collins, Robin Getz and Alexander M. Wyglinski. *Software-Defined Radio for Engineers*. Mobile Communications Series, United States of America, 2018.
- [5] Fernando Fernández de Villegas. Sdr: Equipos de radio definidos por software. 2008. URL <https://www.ea1uro.com/sdr1/sdr.htm>.
- [6] RTL-SDR. Gnu radio 3.9 ¿cómo instalar gnu radio en ubuntu y derivados?, . URL <https://ubunlog.com/gnu-radio-3-9-llega-con-mejoras-correcciones-de-errores-y-mas/>.
- [7] V. Azúa M. Murillo L. Parrales G. Camposano Y. Pin Regalado, J. Romero. *Redes de Computadora*. Área de Innovación y desarrollo, S.L., Abril 2018.
- [8] María García Gadañón Jesús Poza Crespo, Juan Pablo de Castro Fernandez. *Tema 2. Modulaciones analógicas*. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad de Valladolid, 2019-2020.
- [9] Leandro GG. Radio am. URL <https://elcajondelelectronico.com/tag/modulacion-am/>.

- [10] GRAF Grup de Radio-Freqüència. Tema 3. moduladores y demoduladores. URL http://diposit.ub.edu/dspace/bitstream/2445/173862/4/SC_Tema3%28a%29.pdf.
- [11] Zhandg Ke. *Implementación de un demodulador de radio FM definido por Software*. Universidad Politécnica de Valencia, 2016.
- [12] Juan José Murillo Fuertes Iván Pinar Dominguez. *Laboratorio de Comunicaciones Digitales Radio Definida por Software*. Dep. Teoría de la Señal y Comunicaciones. Universidad de Sevilla, 2011.
- [13] Juan Pablo de Castro Fernandez. *Tema 4. Transmisión digital. Modulación de pulsos*. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad de Valladolid, 2019-2020.
- [14] Juan Pablo de Castro Fernandez. *Tema 5. Transmisión digital*. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad de Valladolid, 2019-2020.
- [15] Miguel S. C. Fsk (frequency shift keying), . URL <https://medium.com/modulaciones-digitales-ask-psk-fsk/fsk-frequency-shift-keying-e64a849dfd42>.
- [16] Mundo Teleco. Modulación digital fsk (frequency shift keying), Octubre 2014. URL <https://mundotelecomunicaciones1.blogspot.com/2014/10/modulacion-digital-fsk-frecuency-shift.html>.
- [17] Miguel S. C. Psk (phase shift keying), . URL <https://medium.com/modulaciones-digitales-ask-psk-fsk/psk-phase-shift-keying-37a85398729f>.
- [18] Inmaculada Hernáez Rioja. *Tema 3. Modulación QAM*. Aholab, Laboratorio de procesamiento de señales. Universidad del País Vasco. URL <https://aholab.ehu.eus/users/inma/psc/tema3.pdf>.
- [19] Fmuser. Modulador y demodulador qam, . URL <https://es.fmuser.net/content/?7551.html>.
- [20] Fmuser. Comprender la demodulación en cuadratura, . URL <https://es.fmuser.net/content/?7002.html>.

- [21] Wiki GNU Radio. Guided tutorial psk demodulation. URL https://wiki.gnuradio.org/index.php/Guided_Tutorial_PSK_Demodulation.