

**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

Implementación de un sistema receptor 5G usando radio definida  
por software.

**PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**INGENIERO EN TELECOMUNICACIONES**

Autores:

Danny Paul Cali Gusqui

Daniel Fernando Zumba Briones

GUAYAQUIL – ECUADOR

AÑO 2022

## **DEDICATORIA**

Este proyecto de titulación está dedicado a mi madre Carmen del Rocío Cali Gusqui quien ha sido mi guía y mi apoyo incondicional a lo largo de mi vida estudiantil, forjó mi carácter y ayudó a cumplir cada una de mis metas y sueños. A mis abuelitos Alberto y Cristina que me enseñaron la importancia del trabajo honesto y los buenos modales que debe llevar un caballero, y que desde el cielo me cuidan y envían las mejores de las bendiciones. A mis tíos y hermanos que me han llenado de amor y me motivaron a seguir adelante.

Finalmente, una dedicatoria especial a mi pareja Ericka Noemí que nunca dejo de creer en mí y estuvo a mi lado siempre que más lo necesité y ahora es partícipe de este logro alcanzado. A mis amigos José Luis, Jorge, Bryan, César, Anthony, Juan Carlos, Ariel y Claudia que hicieron de mi etapa de colegio y universidad más amena y son testigos de mi crecimiento personal y profesional.

Danny Paul Cali Gusqui

Mi trabajo va dedicado a mi mamá Marisol Briones Briones, quien siempre me ha apoyado en todo momento, quien ha sido mi principal motivación; a mi papá que Paul Zumba Córdova quien siempre ha estado presto para brindarme ayuda; mi hermano Paul Zumba Briones con quien he compartido todo este tiempo y mi novia Valeska Villegas Sánchez quien siempre ha confiado en mí incondicionalmente.

También quisiera dedicar este trabajo a mis familiares y amigos que han estado conmigo a lo largo de mi vida universitaria, con quienes he compartido muchos momentos agradables y son pilar fundamental en mi desarrollo personal y profesional.

Daniel Fernando Zumba Briones

## **AGRADECIMIENTO**

Doy gracias Dios por haberme bendecido y acompañado a lo largo de toda mi carrera universitaria, pese a las adversidades que tuve que enfrentar su luz y palabra me llenaron de fuerzas para culminar esta etapa de mi vida.

Gracias a mi profesor Ph.D. Francisco Novillo, a mi tutor MSc. Alfredo Nuñez, y al MSc. Eduardo Chancay por todo el apoyo brindado en la elaboración de este proyecto integrador y que son parte fundamental del mismo. Al Ph.D. Boris Ramos y a la MSc. Verónica Ojeda por haber resuelto las dudas y problemas que se presentaron en este trabajo. Finalmente, un agradecimiento especial a mi compañero Daniel Zumba por su esfuerzo, dedicación, y responsabilidad para culminar con éxito este proyecto.

Danny Paul Cali Gusqui

En primer lugar, doy gracias a Dios por haberme guiado en todo momento y por estar presente en cada paso que doy en la vida y bendecirme enormemente.

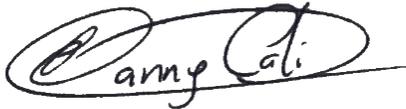
Seguidamente, agradezco a mis maestros que se han esforzado mucho por incentivar me a seguir aprendiendo en muchos ámbitos, no solo lo académico; y también a doy gracias a mis compañeros con quienes he compartido aula de clases presencial y/o virtual.

Un agradecimiento especial a mis profesores: Ph.D. Francisco Novillo, mi tutor MSc. Alfredo Nuñez, y MSc. Eduardo Chancay; quienes me han guiado a lo largo de este proyecto; también gracias a la MSc. Verónica Ojeda quien me ayudó a resolver inconvenientes oportunamente. También le agradezco mucho a mi amigo y profesor, Ing. Efrén Herrera con quien compartí gratos momentos en el salón de clases y en la cancha de fútbol. Para finalizar, a mi compañero Danny Cali quien ha sido un líder y sin su ayuda no hubiera sido posible poder realizar este trabajo.

Daniel Fernando Zumba Briones

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Danny Paul Cali Gusqui y Daniel Fernando Zumba Briones damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



Danny Paul Cali Gusqui



Daniel Fernando Zumba Briones

## EVALUADORES

.....

PhD. Francisco Vicente Novillo Parales

.....

MSc. Alfredo José Nuñez Unda

## RESUMEN

El programa de maestría en Telecomunicaciones de la ESPOL, pretende incursionar en la enseñanza de la quinta generación de redes móviles o 5G a través de prácticas de laboratorio que permitan a los estudiantes simular y analizar tales sistemas de comunicación. Sin embargo, se conoce que los equipos apropiados y la infraestructura para este tipo de sistemas son muy costosos, por lo cual resulta complicado levantar un sistema 5G real.

Como solución a esta problemática se ha adoptado el uso de dispositivos de radio definida por software (SDR) que son más accesibles y adaptables con los softwares Matlab y GNU radio reduciendo significativamente los recursos económicos.

Se tiene como objetivo desarrollar seis formatos de prácticas de laboratorio donde se van a tratar los siguientes temas: Introducción al ADALM-PLUTO, modulaciones digitales (BPSK, QPSK y 16QAM), filtro de acoplamiento del receptor, recuperación de tiempo de símbolo, sincronización de frecuencia portadora, y sincronización de trama con codificación de canal. Los estudiantes tendrán que programar las principales etapas de un sistema receptor 5G para luego convertirlos en respectivos bloques de Simulink y construir dicho sistema adaptable con el SDR ADALM-PLUTO.

Con esto, los estudiantes entenderán el funcionamiento de un sistema receptor 5G en las diferentes etapas que lo constituye, además podrán variar ciertos parámetros de simulación que invite al análisis crítico y adquisición del conocimiento de forma didáctica mediante las diferentes herramientas que ofrece Simulink como es el diagrama de ojo, potencia de la señal, figura de constelaciones y BER.

## ABSTRACT

ESPOL's master's program in Telecommunications aims to venture into the teaching of the fifth generation of mobile networks or 5G through laboratory practices that allow students to simulate and analyze such communication systems. However, it is known that the appropriate equipment and infrastructure for this type of system are very expensive, which is why it is difficult to build a real 5G system.

As a solution to this problem, the use of software-defined radio (SDR) devices has been adopted, which are more accessible and adaptable with Matlab and GNU radio software, significantly reducing economic resources.

The objective is to develop six formats of laboratory practices where the following topics will be developed: Introduction to ADALM-PLUTO, digital modulations (BPSK, QPSK and 16QAM), receiver coupling filter, symbol time recovery, synchronization of carrier frequency, and frame synchronization with channel coding. Students will have to program the main stages of a 5G receiver system to then convert them into respective Simulink blocks and build such an adaptable system with the SDR ADALM-PLUTO.

With this, students will understand the operation of a 5G receiver system in the different stages that constitute it, in addition they will be able to vary certain simulation parameters that invite critical analysis and acquisition of knowledge in a didactic way through the different tools that Simulink offers, such as the eye diagram, signal strength, constellation plot and BER.

## ÍNDICE GENERAL

DEDICATORIA	1
AGRADECIMIENTO	3
DECLARACIÓN EXPRESA	5
EVALUADORES	2
RESUMEN	2
ABSTRACT	3
ÍNDICE GENERAL	4
ÍNDICE DE FIGURAS	6
CAPÍTULO 1	7
1.1    Introducción	7
1.2    Definición del problema	8
1.3    Justificación del problema	9
1.4    Objetivos	9
1.5    Metodología	10
1.6    Resultados esperados	11
CAPÍTULO 2	12
2    Marco teórico	12
2.1    Radio definida por software	12
2.1.1    Aplicaciones prácticas de SDR	13
2.2    Tecnología 5G	13
2.2.1    Espectro y frecuencia para el 5G	14
2.2.2    Alcance e Impacto de la tecnología 5G en el mundo real	15
2.3    Arquitectura de la red 5G	16

2.4	Arquitecturas de las RAN 5G	17
2.4.1	RAN Centralizado	17
2.4.2	RAN en cloud	17
CAPÍTULO 3		18
3.1	Descripción del escenario	18
3.2	Planteamiento de la solución	19
3.3	Diseño del sistema receptor 5G	19
3.4	Diagramas de funcionamiento	20
CAPÍTULO 4		22
4	Análisis de resultados	22
4.1	Configuración del dispositivo SDR ADALM-PLUTO	22
4.2	Parámetros de simulación	23
4.2.1	Modulaciones digitales	23
4.2.2	Filtro de acoplamiento del receptor (Matched Filter)	24
4.2.3	Recuperación de tiempo de símbolo	27
4.2.4	Sincronización de la frecuencia de portadora	28
4.2.5	Sincronización de trama y codificación de canal	30
4.3	Descripción de pruebas	31
CAPÍTULO 5		34
5	Conclusiones y recomendaciones	34
5.1	Conclusiones	34
5.2	Recomendaciones	35
BIBLIOGRAFÍA		36
ANEXOS		38

## ÍNDICE DE FIGURAS

<i>Figura 2.1 Arquitectura de la red 5G</i> .....	16
<i>Figura 3.1 Laboratorio de radiocomunicaciones en donde se llevarán a cabo las prácticas los estudiantes de la maestría en Telecomunicaciones</i> .....	19
<i>Figura 3.2 Diagrama de bloques del sistema receptor 5G usando SDR ADALM-PLUTO radio</i> .....	21
<i>Figura 4.1 Comando iio_info-s en la consola de Windows para verificar la instalación de los drivers</i> .....	22
<i>Figura 4.2 Instalación de complementos ADALM-Pluto Radio</i> .....	23
<i>Figura 4.3 Figuras obtenidas después de las modulaciones: BPSK, QPSK Y 16QAM</i> .....	24
<i>Figura 4.4 Parámetros del bloque del filtro de acoplamiento</i> .....	25
<i>Figura 4.5 Espectro de la señal recibida y ancho de banda para un valor de roll off de <math>\alpha=0</math> (a), <math>\alpha=0.5</math> (b), y <math>\alpha=1</math> (c)</i> .26	
<i>Figura 4.6 Constelación de la señal recibida para un valor de roll de <math>\alpha=0</math> (a), <math>\alpha=0.5</math> (b), y <math>\alpha=1</math> (c)</i> .....	26
<i>Figura 4.7 Parámetros del bloque de recuperación de tiempo de símbolo</i> .....	27
<i>Figura 4.8 Constelación de la señal recibida después de la sincronización de símbolos</i> .....	28
<i>Figura 4.9 Parámetros del bloque sincronizador de frecuencia portadora</i> .....	29
<i>Figura 4.10 Constelación de la señal recibida después de la sincronización de portadora</i> .....	29
<i>Figura 4.11 Parámetros del bloque sincronizador de trama</i> .....	30
<i>Figura 4.12 BER obtenido</i> .....	30
<i>Figura 4.13 Pruebas realizadas en el laboratorio de radiocomunicaciones de la FIEC (a) pruebas realizadas con dos dispositivos SDR ADALM-PLUTO mediante antenas (b)</i> .....	31
<i>Figura 4.14 Estructura qpsktx para el transmisor (a) y estructura qpskrx para el receptor (b)</i> .....	32
<i>Figura 4.15 Resultados obtenidos de la simulación del sistema receptor 5G en la pantalla de Simulink</i> .....	33
<i>Figura 4.16 Diagrama de bloques del sistema receptor 5G con modulación QPSK</i> .....	33

# CAPÍTULO 1

## 1.1 Introducción

Las telecomunicaciones sin lugar a duda experimentan un crecimiento y desarrollo constante a la par de la tecnología, provocando que el ser humano las aproveche de una manera más eficiente, confiable y sofisticada en el ámbito laboral, educación, salud, entretenimiento o cualquier otra situación que representa comunicación [1]. Dentro del amplio estudio de las telecomunicaciones encontramos diferentes equipos y sistemas de comunicación al servicio de la sociedad; actualmente una de las tecnologías que avanza a gran escala son las comunicaciones inalámbricas, acelerando la obsolescencia de los dispositivos de comunicaciones al poco tiempo de su producción [1].

Los principales obstáculos a resolver se focalizan en rediseñar los sistemas de comunicaciones para maximizar la incorporación de nuevas tecnologías y trabajar en manera conjunta con el resto de los dispositivos [2]. En los últimos años el auge de la tecnología de Radio Definida por Software (SDR) conduce a la necesidad de su estudio e implementación por parte de los estudiantes de ingeniería en telecomunicaciones, ya que gracias a sus aportaciones permitirán la evolución de las comunicaciones inalámbricas, como por ejemplo el 5G [2].

La tecnología 5G ya es una realidad en algunas regiones del mundo, y se refiere a la quinta generación de redes móviles. Sin lugar a duda es el avance más significativo de los últimos años ya que proporcionará una gran velocidad en la transferencia de datos, diez veces más rápido de lo que es posible actualmente, y una reducción de hasta cinco milisegundos en la latencia, con lo cual estaremos prácticamente conectados en tiempo real [3].

SDR o Radio Definida por Software es lo nuevo en los sistemas de radiocomunicaciones, gracias a su combinación de hardware y software lo convierte en una tecnología flexible, y de gran utilidad tanto en servicios de telefonía celular como en el sector militar y aeroespacial [5]. SDR promete a los usuarios capturar, demodular, y acceder a las señales de radiofrecuencia (RF) a través de una sola pieza de hardware y vasta gama de frecuencias de radio [4].

Generalmente cuando se hace uso de los instrumentos y equipos de laboratorio para alguna práctica en específico, se requiere de un manual detallado del procedimiento a seguir, configuración de los equipos o calibración de ciertos instrumentos [2]. Es por esta razón que en el presente proyecto integrador se va a elaborar un manual de prácticas para la implementación de un sistema receptor 5G basado en radio definida por software en la plataforma de programación de MATLAB; dicho manual mejorará las competencias y habilidades de los estudiantes que se encuentre cursando el programa de maestría en telecomunicaciones (MET) de la Escuela Superior Politécnica del Litoral (ESPOL), ya que podrán acceder a una información más didáctica y orientada al diseño de este tipo de sistemas.

## **1.2 Definición del problema**

Con la llegada de nuevas tecnologías, también han evolucionado los sistemas de radiocomunicaciones, trayendo consigo diversas innovaciones como la introducción de procesadores, el desarrollo de procesadores digitales de señales (DSP), hasta llegar al nuevo concepto de equipos de radio definido por software o SDR [2].

La quinta generación de tecnologías móviles es uno de los principales tópicos a dictarse dentro del programa de maestría en Telecomunicaciones de la ESPOL; sin embargo, levantar un sistema 5G real involucra un enorme gasto económico debido al precio de los materiales y equipos

que se necesitan. Por esta razón los estudiantes del MET presentan dificultades al momento de realizar las prácticas de laboratorio con sistemas 5G, ya que no cuentan con un material especializado en este tipo de sistemas [7].

### **1.3 Justificación del problema**

Los estudiantes que se encuentran cursando una maestría en Telecomunicaciones de la ESPOL necesitan profundizar en el estudio de la tecnología 5G, y sean capaces de definir las características, estándares, tecnologías de acceso y todo lo relacionado a esta nueva generación de tecnologías móviles [5].

La radio definida por software es de gran utilidad en múltiples servicios, desde la tecnología celular hasta el ámbito militar, y largo plazo se prevé que el SDR sea la tecnología dominante en las radiocomunicaciones, y la vía para llegar a la radio cognitiva [5]. Es por esta razón, que el manual a elaborarse permitirá a los estudiantes que se encuentren cursando una maestría en Telecomunicaciones en la ESPOL acceder a una información enfocada en los sistemas receptores 5G basados en SDR.

### **1.4 Objetivos**

El siguiente proyecto integrador tiene como objetivo principal “Elaborar una guía de laboratorio de un sistema receptor 5G usando radio definida por software para el programa de maestría en Telecomunicaciones de la Escuela Superior Politécnica del Litoral.” Para alcanzar esta meta, se plantean los siguientes objetivos específicos:

- Identificar los sistemas de radiocomunicaciones actuales, y definir aquellos elementos que serán implementados por el software de Matlab.

- Programar cada uno de los bloques que conforman el sistema SDR en la función del Simulink de MATLAB.
- Implementar en diagrama de bloques el sistema de SDR, y comprobar su funcionamiento con ayuda del módulo ADALM-PLUTO.

## **1.5 Metodología**

En este proyecto integrador se empleará el método experimental, debido a que se manipularán variables como la frecuencia, ancho de banda y ganancia del módulo de aprendizaje activo ADALM-PLUTO, y determinar cómo ciertos cambios afectan en el resultado del sistema receptor 5G basado en SDR. También, se va a recurrir a la metodología “Waterfall” (cascada) ya que el proyecto se va a desarrollar de manera secuencial, comenzando con la fase análisis y diseño, hasta llegar con el testeo y puesta en producción [6].

En primer lugar, se procede a instalar los drivers del módulo ADALM-PLUTO tanto en el software de Matlab como en el ordenador de tal manera que se puedan comunicar entre sí [6]. Después de haber establecido la comunicación entre los dispositivos, se procede a diseñar el entorno del sistema receptor 5G basado en radio definido por software e identificar aquellos elementos que se van a programar mediante el software en la plataforma de MATLAB. De tal manera, que luego se proceda a implementar los bloques que conforman nuestro sistema de comunicación en la función SIMULINK de Matlab.

Finalmente, se realizan las pruebas pertinentes del sistema 5G a través del módulo ADALM-PLUTO [7]. Con el análisis de las medidas obtenidas se procederá a elaborar el manual de prácticas que serán usados por los estudiantes de postgrado de la Escuela Superior Politécnica del Litoral en el programa de maestría en Telecomunicaciones.

## **1.6 Resultados esperados**

El siguiente proyecto integrador permitirá a los estudiantes de postgrado de la ESPOL profundizar en el estudio de los sistemas 5G basados en radio definida por software, mediante una guía de laboratorio enfocada en la simulación de estos sistemas desarrollada en MATLAB; además, serán capaces de descubrir las principales características como: formas de ondas, técnicas de modulación, estándares, tecnologías de acceso, etc. de la nueva generación de los sistemas de comunicación 5G.

## CAPÍTULO 2

### 2 Marco teórico

Las redes definidas por software (SDN) [8] y radio definida por software (SDR) [5] son dos tecnologías totalmente diferentes; sin embargo, comparten una característica en cuanto a su función “definidas por software”, es decir, la capacidad de reconfiguración. Los administradores ahora cuentan con la posibilidad de recopilar señales o cambiar parámetros dentro del paquete con gran facilidad, y encontrar rápidamente una banda de frecuencia o ruta adecuada para los datos recopilados, de esta manera se estaría avanzando hacia un entorno auto adaptativo [8].

#### 2.1 Radio definida por software

El concepto de radio definida por software o SDR puede tener variaciones, esto porque su enfoque se define de acuerdo al campo donde se esté aplicando ya que se trata de un tema que forma parte de diversas disciplinas [2].

Uno de los primeros organismos en darle una definición al SDR fue Wireless Innovation Forum, donde definieron el SDR a través de su portal web como ‘una radio cuyas funciones que forman parte de la capa física se las define mediante el uso de software [2].

Lee Pucker, colaborador de esta compañía mencionada anteriormente, comparte este concepto y además agrega: ‘Esta tecnología tiene múltiples aplicaciones en cuanto a industrias inalámbricas se refiere; tiene diversas capacidades como entregar soluciones de manera eficiente, rápida y con un precio más asequible en cuanto al mercado; pudiendo así solucionar problemas que presentan diversas arquitecturas de radio tradicionales’. Dando así un concepto más sofisticado y entendible para el público, haciendo hincapié en la importancia y facilidad de su uso [2].

Otro concepto de SDR fue aportado por William A. Sethares, donde da una descripción de la palabra como tal, mencionando que: ‘Definido por software hace referencia a que se toman los elementos de vital importancia del radio para poder abordarlos en un software’, y además recalca: ‘Tener en cuenta lo dicho anteriormente implica una visión de diseños actuables, donde se está buscando que los sistemas ya no posean un software fijo, sino que este pueda ser más versátil; esto a su vez, permitirá tener un mejor entendimiento de conceptos o de bloques que hay detrás de cada aplicación de transmisión’. Haciendo referencia a la importancia de conocer un equipo en su función y manejo para poder darle diferentes enfoques de trabajo y poder hacerlo más útil incluso comercialmente [2].

### ***2.1.1 Aplicaciones prácticas de SDR***

Las oportunidades que brinda la radio definida por software son amplias, y permite la realización de una infinidad de proyectos, y es que esta tecnología puede adaptarse en un dispositivo para emitir y recibir todo tipo de señales [9].

Entre las aplicaciones de los SDR se destaca el uso de equipos que son mas competitivos en cuanto a que flexibilidad, mas económicos y fácil de adquirir. [9]. Las tecnologías inalámbricas tanto de telefonía como de internet se permite detectar si la señal se está enviando correctamente; también sirve como analizador de espectro y para crear filtros personalizados como: controles de ganancia, supresores de ruido, y procesador de señales digitales (DSP). Además, un SDR actúa como receptor de señales de satélites para detectar estaciones meteorológicas de un aeropuerto, y para leer y emitir señales GPS [8].

## **2.2 Tecnología 5G**

La red 5G tuvo sus inicios en el 2015 [10], conocida como tecnología OWA (Open Wireless Architecture). Respecto a las capas que posee: La capa física y de enlace son quienes

soportan la tecnología inalámbrica, la capa de red tiene dos características (red superior soporte móvil y la red menor está para dar soporte a la interfaz), la capa de transporte y de sesión son quienes dan soporte al protocolo OTT (Protocolo de Transporte Abierta), que es justamente quien se encarga de las pérdidas de bits, y la capa de aplicación se encarga de validar la calidad de los servicios entregados mediante muchas redes [11].

El enrutamiento está basado en direccionamiento IP, posee una alta velocidad en el orden de los Gigas (de 1 a 10 Gbps). Además, presenta un gran rendimiento en tiempo real (debido a su rapidez, baja latencia y retardo), haciendo uso de la radio definida por software, sistemas de costes bajos y con poca interferencia ya que acepta 100 veces más de dispositivos para que se puedan conectar a la red, también de garantizar la cobertura; otra característica de gran importancia es el ahorro energético que es de aproximadamente el 90% [11].

### ***2.2.1 Espectro y frecuencia para el 5G***

La tecnología perteneciente a 5G se concentra en 3 grandes rangos de cobertura, como los son: Frecuencias menores a 1 GHz, entre 1 GHz y 6 GHz, y mayores a 6 GHz [12].

Para frecuencias menores a 1 GHz se tendría contemplado hacerse principalmente en zonas urbanas, suburbanas y rurales, donde tendría un rol fundamental en servicios de Internet de las cosas. Para frecuencias entre 1 GHz y 6 GHz, se entrega una buena relación entre cobertura y capacidad, donde se manejan frecuencias entre 3.3 GHz y 3.8 GHz y se prevé que sea la base inicial para bastantes servicios de 5G; aunque también se puede mencionar otras bandas que pueden ser muy usadas como el de 1.8 GHz, 2.3 GHz y 2.6 GHz, principalmente. También, existen frecuencias mayores a 6 GHz que pueden hacer muy buen trabajo para esta tecnología 5G, principalmente para ofrecer una gran velocidad; de momento, las bandas de 26 GHz y de 28 GHz poseen mayor apoyo internacional [12].

### ***2.2.2 Alcance e Impacto de la tecnología 5G en el mundo real***

La tecnología 5G se está expandiendo a un ritmo acelerado en los últimos años. A nivel mundial, el uso de redes de telecomunicaciones de quinta generación empezó por el 2019 y tal como mencionó la empresa llamada GSA en su informe, se dio a conocer el alcance comercial de la tecnología 5G, la cual se extiende alrededor del mundo hasta en 58 países (información validada en Junio del año pasado) [13]. Algunos de los países que ya gozan de esta tecnología son Corea del Sur, China, Tailandia, Reino Unido, España, Italia, Suecia, Estados Unidos, Canadá, Suiza, Alemania, Bélgica, Emiratos Árabes Unidos, Austria, entre otros [13].

Sin embargo, para Latinoamérica el despliegue de 5G aún se encuentra en una etapa inicial debido a sus limitaciones por factores económicos y políticas públicas y privadas [3]. A esto debemos agregar que los operadores se han enfocado más en la migración de los clientes de 2G y 3G hacia las redes 4G, y es que en la región todavía no se llega a una cobertura 4G del 80%, exceptuando a Uruguay y Chile, según lo afirma José Otero, vicepresidente para América Latina y el Caribe de 5G Américas, organización de proveedores y fabricantes de servicios de telecomunicaciones [12]. Según el experto alemán en datos de mercado, Statista, se espera que para el año 2025 el promedio global del acceso a las redes móviles 5G sea del 14%; mientras que para América Latina la cifra no superaría del 7% [3].

En cuanto a los servicios que presenta, son muchos, pero destacan: El mundo podrá tener acceso a Wi-Fi, dirección IP que posea va a ir de acuerdo a la ubicación, mayor señal de radio, diagnóstico remoto, telemedicina, monitoreo a cualquier zona para organizaciones gubernamentales e investigaciones en general, predecir y comunicar más ágilmente los desastres naturales, entre otras aplicaciones muy innovadoras [13].

### 2.3 Arquitectura de la red 5G

El mundo se encuentra cada vez más conectado y en constante cambio. 5G consta de una arquitectura mejorada, donde se tiene como principal objetivo ofrecer una gran variedad de servicios inalámbricos tanto en el ámbito industrial como en el consumidor final. [14]

En cuanto a las anteriores generaciones de redes móviles se basaban principalmente en brindar acceso de datos móviles de gran velocidad y confiabilidad. La arquitectura del 5G es más inteligente en comparación a las arquitecturas anteriores, en la figura 2.1 se puede apreciar que cuenta con redes RAN, las cuales ofrecen muchas ventajas, entre las cuales destaca que ya no se ve limitada por su infraestructura [14] [15].

La tecnología 5G debe satisfacer los requerimientos que demanden sus diversas aplicaciones, ya sea viajar largas distancias, transportar mucha información. Y en búsqueda de esta mejoras es que se ha hecho uso de diversas bandas para poder aprovechar el mejor rendimiento posible para cada aplicación como sea de conveniencia, por ello hace uso de espectros en todas sus bandas (baja, media y alta), de cualquier tipo de fuente sea esta licenciada o no. [15].

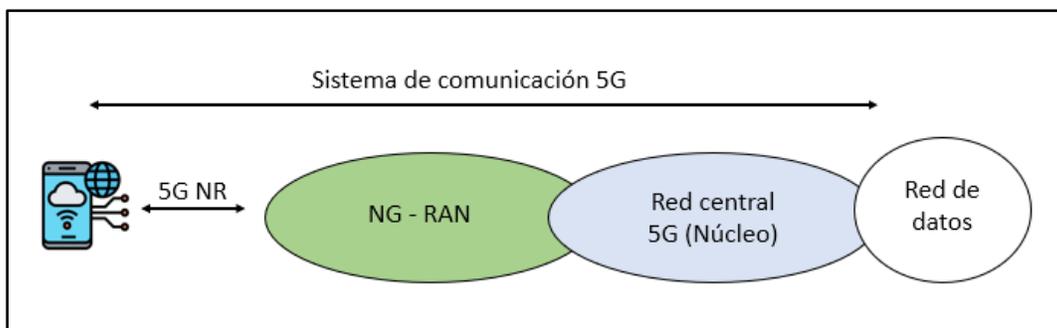


Figura 2.1 Arquitectura de la red 5G

## **2.4 Arquitecturas de las RAN 5G**

En las redes RAN (Radio Access Network) tradicionales, cada estación base dispone de su propia BBU (Base Band Unit) responsable de la modulación de la señal, RRU (Remote Radio Unit) encargado del procesamiento de radiofrecuencia, un alimentador que conecta la RRU y la antena, esta última debe convertir las ondas viajeras guiadas en cables en ondas espaciales en el aire. De esta manera, se plantean dos posibles evoluciones para las redes RAN 5G [16]:

### **2.4.1 RAN Centralizado**

Este concepto involucra a varias unidades de radio remotas (RRUs) conectadas a una única BBU que puede ser local o remota. De esta manera se comparten los recursos de manera flexible y una reducción en los costos de energía [16].

### **2.4.2 RAN en cloud**

También conocida como cloud RAN (o en español RAN en la nube), es un tipo de tecnología de acceso por radio (RAN) de vital importancia para poder desplegar las redes 5G ya que funciona mediante computación en la nube [16].

Esta tecnología se encarga de implementar funciones de red mediante la virtualización como NFV (Network Functions Virtualization), adicionalmente proporciona una plataforma para más servicios de aplicaciones basados en cloud como lo es el servidor MEC, el cual actúa en la capa física también [10].

El CLOUD RAN ayuda a que la red inalámbrica 5G sea más escalable y flexible, además de que son más simples de operar (en cuanto a administración y mantenimiento) y más económico para un propietario, ya que cuando se centralizan los equipos BBU en el 'hotel BBU' se va a disminuir la demanda energética y de enfriamiento [10].

## CAPÍTULO 3

### 3.1 Descripción del escenario

Para la implementación del sistema receptor 5G usando SDR se tomó en consideración al laboratorio de Radiocomunicaciones ubicado en el bloque 11D de la ESPOL. Este bloque acoge a la gran mayoría de los laboratorios de la Facultad de Ingeniería en Electricidad y Computación (FIEC) lo que lo convierte en el espacio ideal para probar el diseño y evaluar el desempeño de nuestro sistema receptor 5G.

El laboratorio de radiocomunicaciones que se puede observar en la figura 3.1 se encuentra altamente equipado tanto en computadoras con el software de Matlab instalado como en los dispositivos SDR, ADALM-PLUTO para realizar las pruebas pertinentes y obtener los mejores resultados posibles. De tal manera que, los estudiantes de MET podrán llevar a cabo las prácticas de laboratorio correspondientes a la materia de comunicaciones móviles.

Para la implementación de este sistema receptor 5G debemos considerar a los factores que inciden directamente en la propagación de las señales, como por ejemplo ciertas obstrucciones, o interferencias con otros equipos. Dado a que no se utilizarán muchos equipos, al momento de realizar las pruebas en exteriores se debe procurar la integridad física de los mismos; para ello se seleccionarán puntos específicos en donde se eviten golpes o caídas.



*Figura 3.1 Laboratorio de radiocomunicaciones en donde se llevarán a cabo las prácticas los estudiantes de la maestría en Telecomunicaciones*

### **3.2 Planteamiento de la solución**

Para desarrollar este proyecto se hará uso del entorno de Simulink; en donde se simulará un sistema receptor de quinta generación basado en radio definida por software, más específicamente con el módulo de ADALM-PLUTO, el cual va a transmitir y recibir la información.

Los bloques que conforman el sistema 5G serán creados mediante código en la plataforma de programación de Matlab; no obstante, también se tomará en consideración bloques propios del software que se encuentran en las distintas librerías de Mathworks [4].

### **3.3 Diseño del sistema receptor 5G**

Una vez que se tenga listo los bloques del sistema receptor 5G se los ubicará en el orden correcto, y se procederá a configurar ciertos parámetros para que trabaje en conjunto con el dispositivo SDR ADALM-PLUTO; el cual es muy adaptable con Matlab dado que ofrece un módulo para la transmisión y recepción de la señal.

El sistema receptor 5G estará en funcionamiento para un medio inalámbrico, y deberá ser robusto frente a los principales problemas que se obtiene en la recepción de señal para este tipo de ambientes. La señal que recibe el dispositivo ADALM-PLUTO será muy débil, por lo que necesita ser intensificada, al mismo tiempo que los símbolos se encuentran dispersos y desfasados de su constelación de referencia, entonces se deberá corregir la sincronización del tiempo de símbolo, de portadora y de trama, con la finalidad de recuperar la mejor calidad de la señal que se ha enviado.

### **3.4 Diagramas de funcionamiento**

En la figura 3.2 podemos apreciar el diagrama de bloques de cada una de las etapas que conforman el sistema receptor 5G. El dispositivo SDR, ADALM-PLUTO debe estar conectado a una PC que cuente con el programa de MATLAB, y una vez dentro del entorno del Simulink se procede a configurar los módulos de ADALM-PLUTO transmitter y receiver para la transmisión y recepción del mensaje; dado que esta señal es muy débil se debe implementar un controlador automático de ganancia para que la señal sea lo suficientemente fuerte y así acoplarla o adaptarla a un filtro de acoplamiento o matched filter, que de preferencia es uno de tipo coseno elevado.

Una vez que la señal atraviese el filtro, el siguiente paso es establecer un sincronizador para el tiempo de símbolo debido al retraso (delay) [17] que se introduce por la interferencia entre símbolos (ISI) [17] al muestrear un punto incorrecto, después tenemos un sincronizador de portadora para ‘enganchar’ la portadora de llegada con la portadora generada en fase internamente; luego se establece un sincronizador de trama con el propósito de resolver los retrasos que son múltiplos del período de símbolo, y así ubicar el inicio y fin del paquete de trama.

En los sistemas de comunicación inalámbricos, la señal encuentra efectos de propagación como reflexiones, dispersión, difracción, y pérdidas de paso, que crean a su vez múltiples caminos

de propagación entre el transmisor y receptor, introduciendo así la ISI en el sistema. La ISI puede ser eliminado utilizando un buen detector o sincronizador de símbolo o por ecualización.

Finalmente, la señal debe pasar por un bloque demodulador QPSK [17] para recuperar la información que fue enviada inicialmente y recuperar el mensaje en bits. Cabe mencionar que la tecnología 5G sigue usando los esquemas de modulación BPSK, QPSK, y M-QAM; aunque se espera que 5G incorpore modulaciones más altas que 4G como 512 QAM y 1024 QAM, para poder alcanzar mayores velocidades de datos y eficiencias espectrales [17].



*Figura 3.2 Diagrama de bloques del sistema receptor 5G usando SDR ADALM-PLUTO radio*

## CAPÍTULO 4

### 4 Análisis de resultados

Para comprobar la eficiencia y desempeño del sistema receptor 5G se realizaron varias pruebas ubicando los dispositivos SDR ADALM-PLUTO sobre una superficie plana en donde la formación de componentes multipasos era inminente debido a las paredes, objetos y equipos alrededor.

Durante las pruebas realizadas se obtuvieron diferentes resultados en cada ejecución del programa. Los mismos que nos permitió observar los problemas a los que se enfrenta los sistemas de comunicación inalámbricos, y como se iban corrigiendo en cada una de las etapas que se han establecido.

#### 4.1 Configuración del dispositivo SDR ADALM-PLUTO

Para trabajar con los equipos SDR ADALM-PLUTO se necesita instalar ciertos componentes para que la computadora en la cual se va a conectar lo reconozca, estos son los drivers libiiio y PlutoSDR-M2k-USB. A través del comando `iio_info -s` en la consola de windows se puede apreciar en la figura 4.1 que la computadora ya reconoce al dispositivo cuando está conectado a través de un puerto USB; además, nos muestra la versión, el número de serie y el puerto que está utilizando el dispositivo.

```
C:\Users\Daniel Zumba>iio_info -s
Library version: 0.23 (git tag: c14a0f8)
Compiled with backends: xml ip usb serial
Unable to create Local IIO context : Function not implemented (40)
Available contexts:
  0: 0456:b673 (Analog Devices Inc. PlutoSDR (ADALM-PLUTO)), serial=104473ce699100101300100007ef88024f [usb:1.6.5]
C:\Users\Daniel Zumba>
```

Figura 4.1 Comando `iio_info -s` en la consola de Windows para verificar la instalación de los drivers

Los dispositivos ADALM-PLUTO se adaptan fácilmente con las plataformas de Matlab y GNU radio, nuestro proyecto integrador se desarrolló en Matlab específicamente en el entorno de Simulink, por lo que era necesario instalar el paquete de complementos de “Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio”, el cual se puede observar en la figura 4.2

Name	Author	Install Date	
 RF Blockset Models for Analog Devices RF Transceivers version 21.1.0	 MathWorks	20 June 2022	:
 Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio version 21.1.4	 MathWorks	8 June 2022	 :
 WLAN Toolbox version 3.2	 MathWorks	7 June 2022	:

Figura 4.2 Instalación de complementos ADALM-Pluto Radio

## 4.2 Parámetros de simulación

En cada una de las etapas del sistema receptor 5G se debe de configurar ciertos parámetros que permitirán un mejor análisis en los resultados. Como se había mencionado, se crearon bloques de Simulink mediante código de programación en Matlab, específicamente para la sincronización de tiempo de símbolo, frecuencia de portadora, y de trama. En cada uno de estos bloques se estableció una máscara o entorno para que el estudiante pueda establecer los parámetros de una manera más interactiva sin recurrir necesariamente al código.

### 4.2.1 Modulaciones digitales

Por medio de los bloques de modulación que se encuentran en las librerías de communications toolbox, se han considerado tres de los esquemas de modulación más importantes que son BPSK, QPSK, y 16 QAM para que operen en conjunto con los módulos de transmisión y recepción del ADALM-PLUTO.

En la figura 4.3 se puede observar la constelación de la señal recibida para cada uno de los esquemas de modulación antes mencionados; para las modulaciones BPSK y QPSK se tiene una rotación de símbolos a manera de un ovalo y circunferencia respectivamente, mientras que para la constelación de 16 QAM se tiene una forma cuadrangular. A diferencia de otros sistemas de comunicación antes estudiados las constelaciones aparecían en sus posiciones de fase y cuadratura de referencia; sin embargo, al tratarse de un medio inalámbrico esto ya no es posible debido a la interferencia entre símbolos ocasionados por la formación de componentes multipasos, o errores de muestreo.

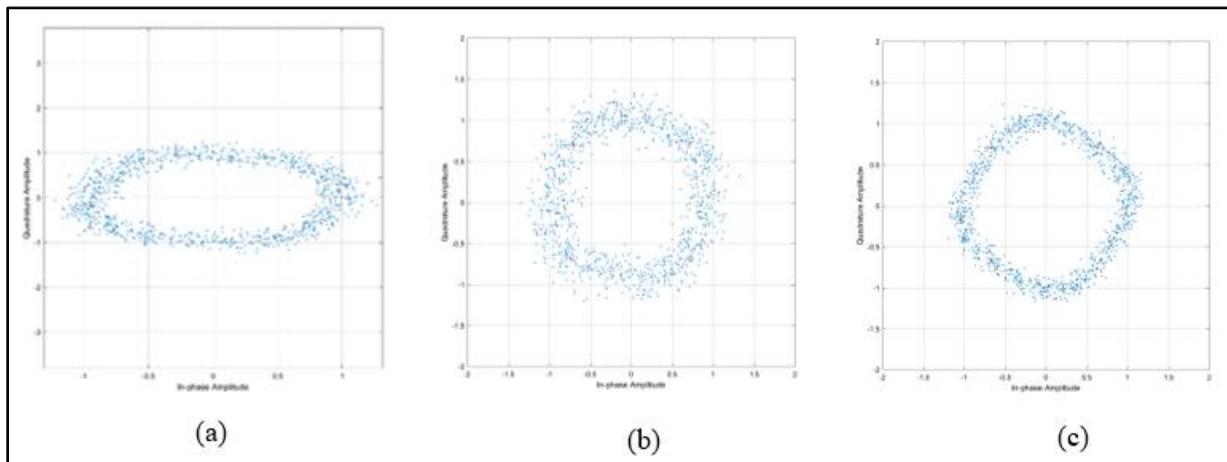


Figura 4.3 Gráficas obtenidas después de las modulaciones: BPSK, QPSK Y 16QAM

#### 4.2.2 Filtro de acoplamiento del receptor (Matched Filter)

En la figura 4.4 podemos apreciar la máscara o interfaz del bloque del filtro de acoplamiento, en el cual debemos seleccionar la forma de coseno de raíz elevado. El factor de roll es un parámetro que puede tomar valores entre 0 y 1. Se realizaron 3 pruebas en total con diferentes valores de roll off, en donde se pudo evidenciar un gran cambio en el ancho de banda, constelación de la señal recibida y diagrama de ojos a medida que se alteraba dicho valor.

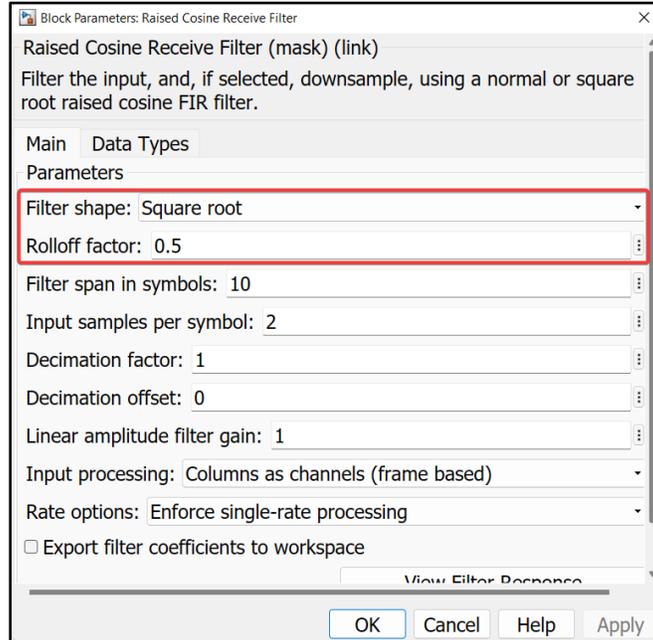


Figura 4.4 Parámetros del bloque del filtro de acoplamiento

En la figura 4.5 se puede observar el espectro que se obtiene a la salida del filtro de acoplamiento en el receptor, y como varía proporcionalmente al factor de roll-off. Es decir, que para un valor de roll off mínimo de cero se tiene un ancho de banda pequeño de aproximadamente 500 MHz, con un valor de 0.5 el ancho de banda se incrementa a 600 MHz, y para un valor máximo de 1 se tiene un ancho de banda aproximado de 700MHz.

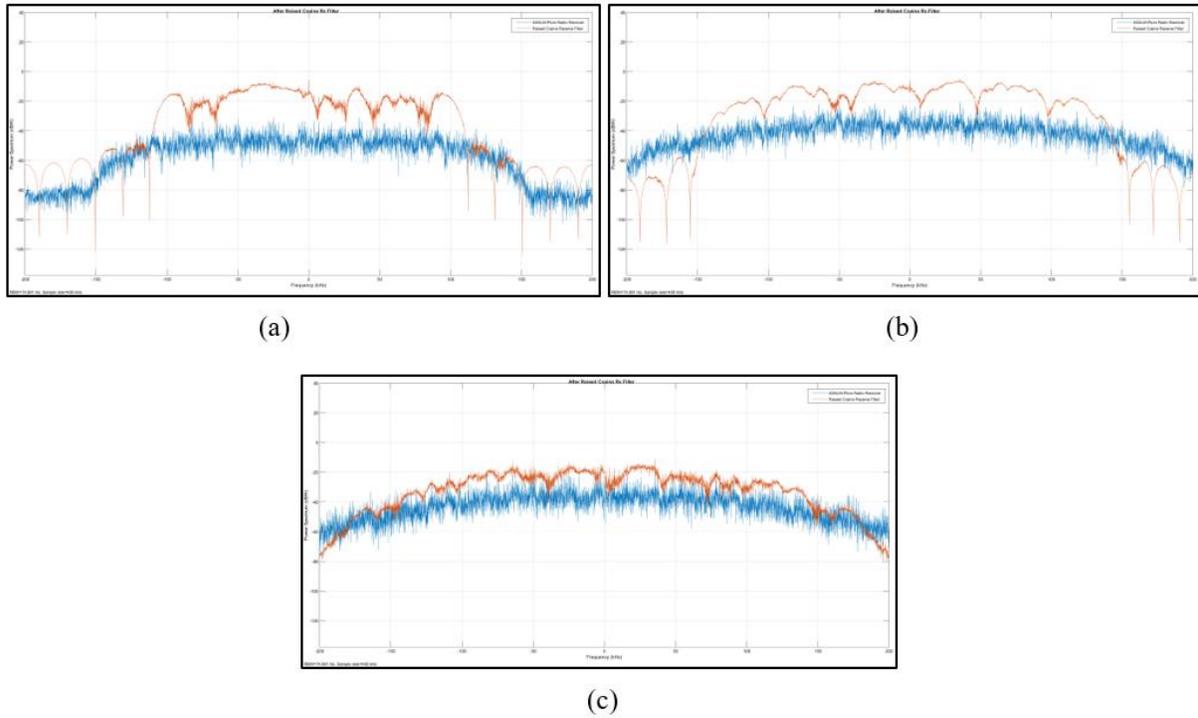


Figura 4.5 Espectro de la señal recibida y ancho de banda para un valor de roll off de  $\alpha=0$  (a),  $\alpha=0.5$  (b), y  $\alpha=1$  (c)

El factor de roll off no solo influye en el espectro de la señal, también afecta a la constelación de la señal recibida; por ejemplo, podemos percibir como los símbolos se encuentran dispersos cuando el valor es mínimo, y empiezan a agruparse y ordenarse a medida que se incrementa el valor de roll off, tal y como se muestra en la figura 4.6.

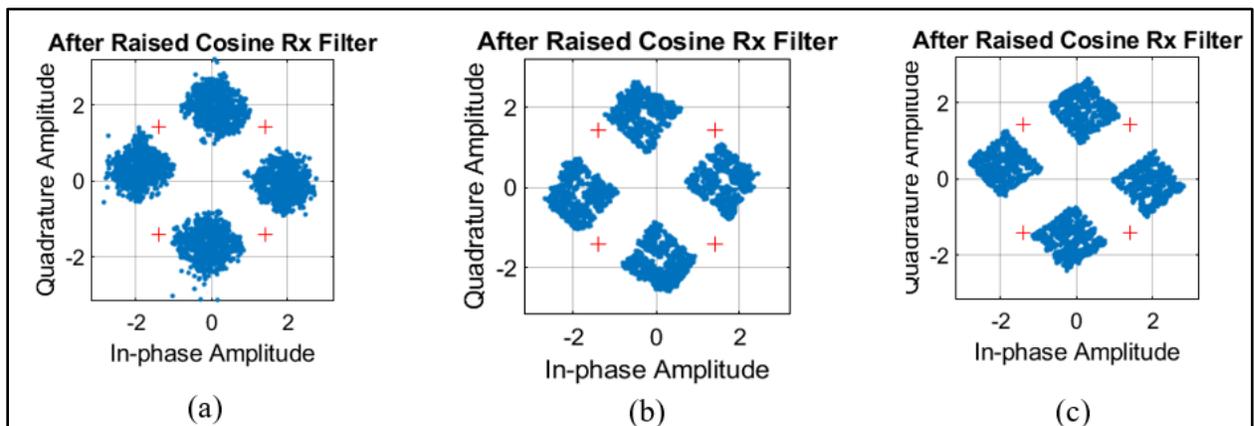


Figura 4.6 Constelación de la señal recibida para un valor de roll de  $\alpha=0$  (a),  $\alpha=0.5$  (b), y  $\alpha=1$  (c)

### 4.2.3 Recuperación de tiempo de símbolo

Consideremos la señal muestreada después del filtro de acoplamiento en el receptor, debido a los errores de muestreo se introduce un retraso desconocido que a su vez ocasiona la interferencia entre símbolos o ISI, y para compensarlo se requiere de la sincronización del símbolo.

En la figura 4.7 se tiene la interfaz del bloque creado mediante código en Matlab para la sincronización del símbolo. En este bloque se corrige el sesgo que existe entre los relojes de sincronización de símbolos entre el transmisor y receptor del sistema; es decir, que establece el tiempo de símbolo adecuado para que se puedan sincronizar debidamente. Adicionalmente, con esto se estaría eliminando el ISI del sistema. En cuanto a los parámetros de configuración se elige como tipo de modulación QPSK y un detector Gardner.

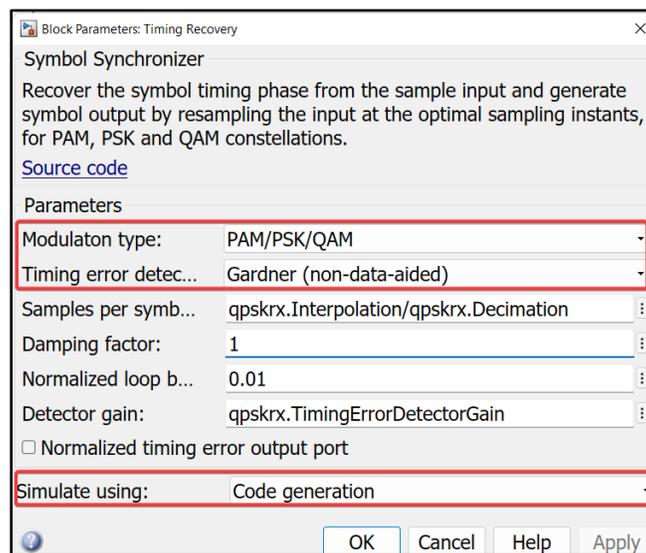


Figura 4.7 Parámetros del bloque de recuperación de tiempo de símbolo

En la figura 4.8 se tiene la constelación de la señal recibida después de la sincronización de símbolo; en este caso se visualiza que los símbolos se encuentran rotando sobre su constelación de referencia dado a que aún no se ha ejecutado el sincronizador de frecuencia.

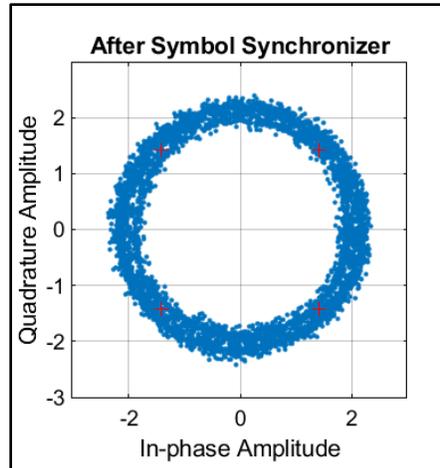


Figura 4.8 Constelación de la señal recibida después de la sincronización de símbolos

#### 4.2.4 Sincronización de la frecuencia de portadora

Los desplazamientos de frecuencia de portadora se producen cuando el transmisor sube la frecuencia que se transmite, por ende, el receptor posee una frecuencia distinta y no es capaz de reconocerla. Por ello, aplicando este filtro lo que se hace es corregir esta rotación de símbolos que se generan por el desplazamiento de frecuencia; obteniendo así los símbolos en la ubicación de constelación de acuerdo a su modulación correspondiente.

En la figura 4.9 se puede ver los parámetros que tiene el bloque sincronizador de frecuencia, donde se escoge la modulación en la que se está trabajando, en este caso se ha hecho la prueba con una QPSK.

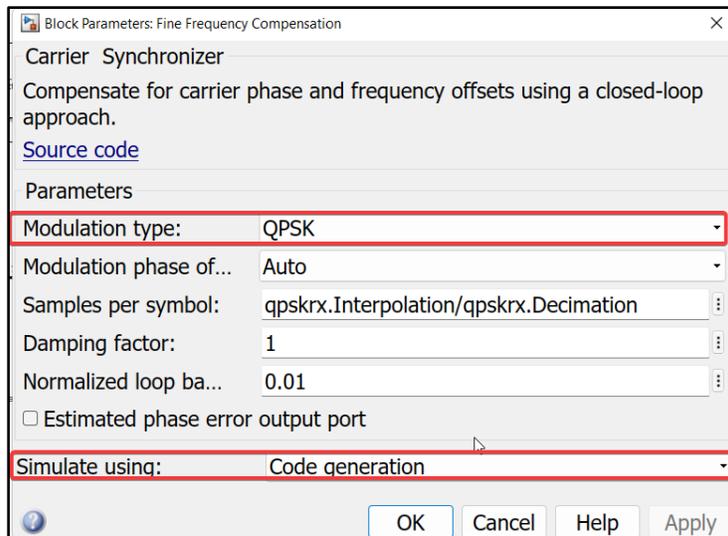


Figura 4.9 Parámetros del bloque sincronizador de frecuencia portadora

En este caso, se puede evidenciar en la figura 4.10 que se ha corregido el desplazamiento de frecuencia, ya que anteriormente se tenía símbolos rotando en forma circular; una vez corregido esto se puede observar que los símbolos en la constelación se asemejan a una QPSK y se encuentran alineadas como corresponde (ya no están rotando).

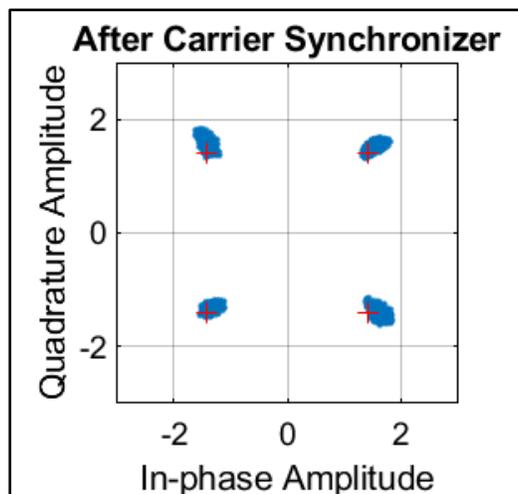


Figura 4.10 Constelación de la señal recibida después de la sincronización de portadora

#### 4.2.5 Sincronización de trama y codificación de canal

La sincronización de trama se ejecuta para poder mitigar los retrasos existentes en la recepción de la señal para los símbolos que son múltiplos del periodo. Al igual que los bloques anteriores también se tiene un interfaz en donde los parámetros de interés son el tamaño de la trama y la longitud del código Barker. (Figura 4.11)

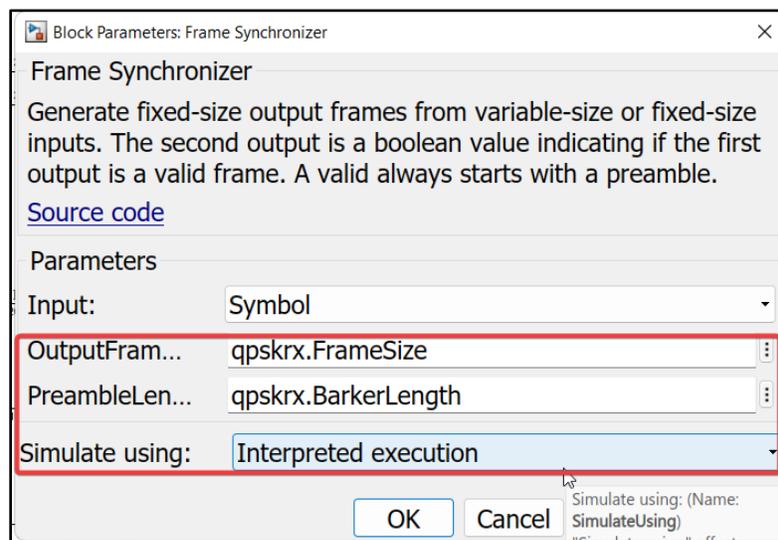


Figura 4.11 Parámetros del bloque sincronizador de trama

Se puede observar en el display presentado el BER del sistema, en este caso se tiene de aproximadamente 0.027 o si se lo quiere dar en porcentaje un 2,7%. El siguiente dato significaría los datos que tienen error  $1,483 \times 10^5$  y en la última casilla muestra el total de datos que se enviaron, en este caso  $5.482 \times 10^6$ ; por consiguiente, si se hace la respectiva división se obtiene el valor inicialmente mencionado del BER. (Figura 4.12)

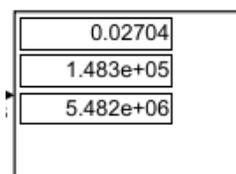


Figura 4.12 BER obtenido

### 4.3 Descripción de pruebas

El siguiente proyecto fue desarrollado inicialmente desde nuestros ordenadores en donde ya se tenía instalado el software de Matlab, y los drivers necesarios para comunicarse con el dispositivo SDR ADALM-PLUTO. Una vez que se tenía el sistema montado en Simulink se procedió a realizar las pruebas en el laboratorio de radiocomunicaciones de la FIEC. (Figura 4.13)

Se realizaron pruebas con dos dispositivos uno en modo transmisor y otro receptor, cabe mencionar que para un mejor análisis de resultados se escogió un medio alámbrico conectando el cable de loopback y un medio inalámbrico usando las antenas. (Figura 4.13)

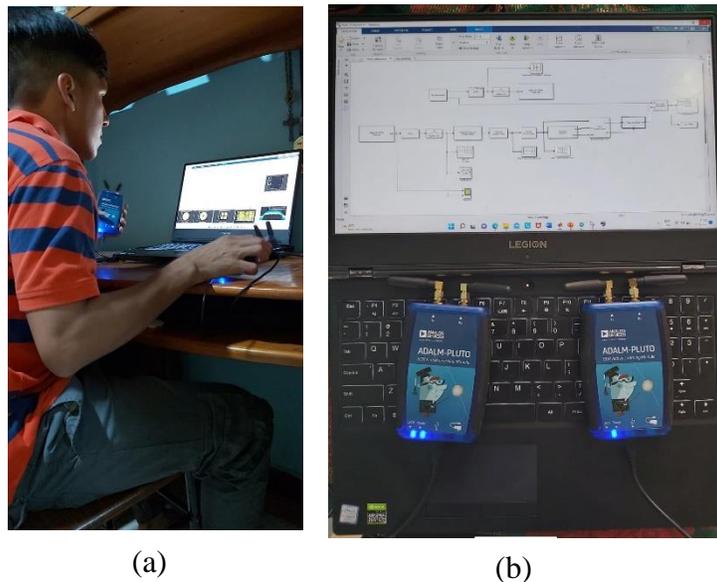


Figura 4.13 Pruebas realizadas en el laboratorio de radiocomunicaciones de la FIEC  
(a) pruebas realizadas con dos dispositivos SDR ADALM-PLUTO mediante antenas (b)

Dado a que se tenía que trabajar con múltiples parámetros de simulación se tomó la decisión de explorar otras herramientas que ofrece Matlab y Simulink; por ejemplo, el uso de estructuras. Una estructura es un conjunto de variables agrupadas dentro de un nombre general, que pueden insertadas en bloques de Simulink y llamar a los valores directamente desde Matlab.

En la figura 4.14 se puede apreciar las dos estructuras que se crearon para las variables que van a ser usados tanto en la etapa de transmisión como recepción de la secuencia de bits.

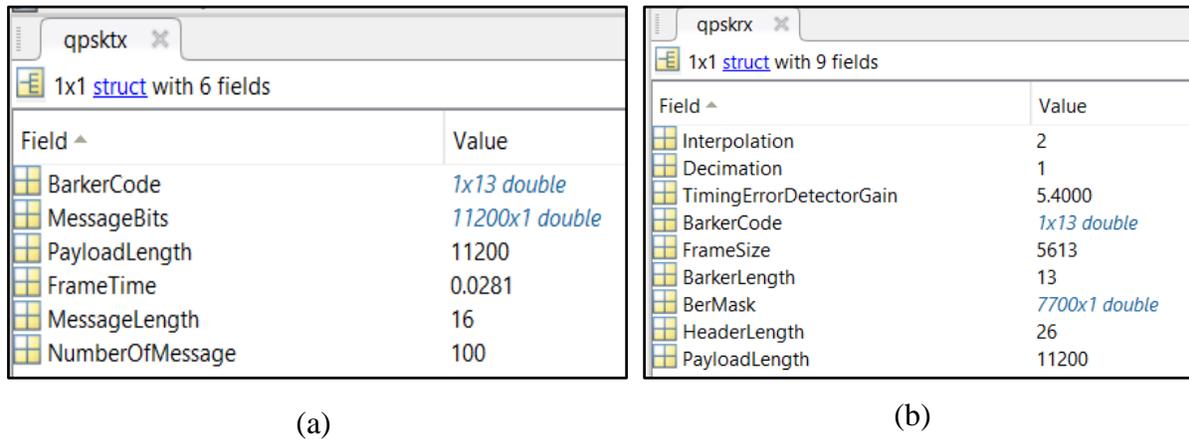


Figura 4.14 Estructura qpsktx para el transmisor (a) y estructura qpskrx para el receptor (b)

Por último, en la figura 4.15 se tiene el diagrama de bloques completo de nuestro sistema receptor 5G con los bloques de sincronización de símbolo, frecuencia de portadora y trama previamente codificados en Matlab para una secuencia de bits con código Barker de longitud 13 y para una portadora única de 915 MHz con modulación QPSK. En adición, en la figura 4.16 se puede apreciar la pantalla de Simulink con los gráficos obtenidos después de un tiempo de simulación de 20 s.

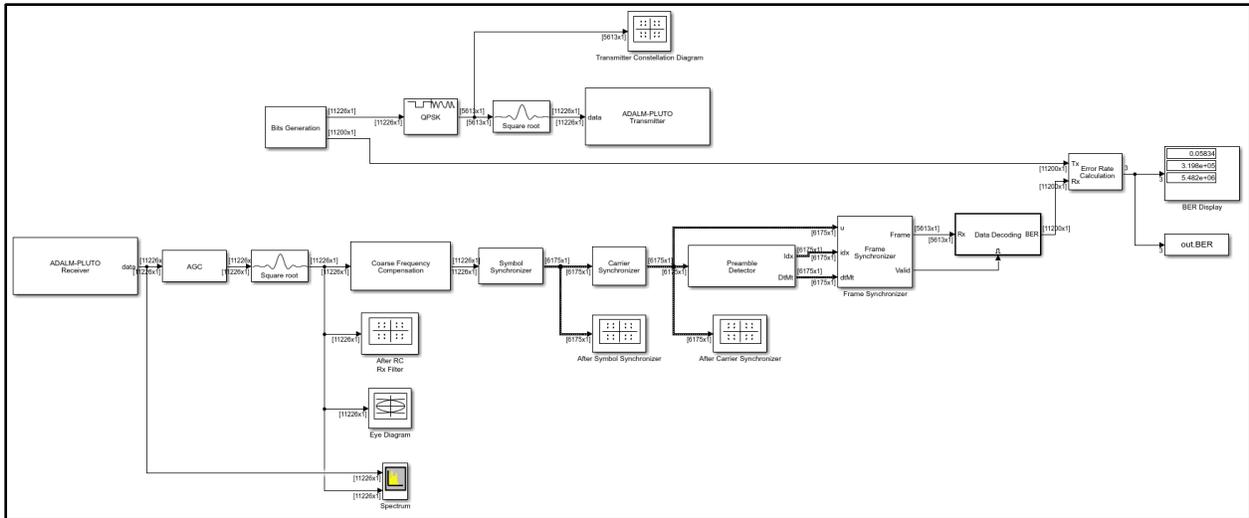


Figura 4.15 Diagrama de bloques del sistema receptor 5G con modulación QPSK

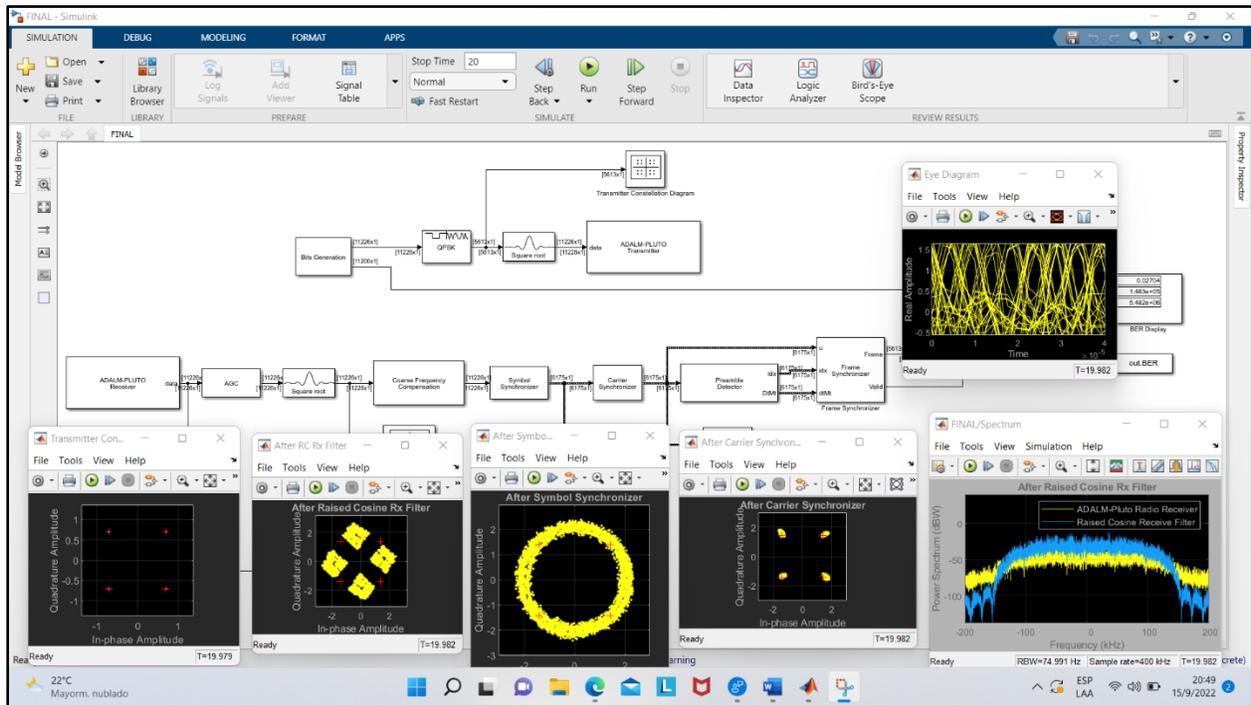


Figura 4.16 Resultados obtenidos de la simulación del sistema receptor 5G en la pantalla de Simulink

## CAPÍTULO 5

### 5 Conclusiones y recomendaciones

#### 5.1 Conclusiones

Las prácticas realizadas guiarán al estudiante para aplicar conocimientos previamente adquiridos, enfocándose más en el uso de cada parámetro utilizado en cada bloque, de manera que van a poder consolidar sus conocimientos y entender su uso en diversas aplicaciones.

Se evidenció el correcto desarrollo de los respectivos bloques programados en Matlab y aplicados en Simulink, de forma que los estudiantes puedan entender la respectiva secuencia lógica que se necesita para poder elaborar por sí solo un esquema de recepción 5G, comprendiendo cada uno de sus componentes.

El sistema receptor 5G implementado en Simulink no alcanzó el tiempo de latencia de 1 ms debido al reloj del dispositivo SDR ADALM-PLUTO; sin embargo, se logró enviar y recibir una secuencia de bits y corregir los principales problemas de comunicaciones inalámbricas como la sincronización de símbolo, portadora y trama causados por la ISI. Por otro lado, el sistema trabajó de manera eficaz la modulación QPSK, recuperando con éxito el mensaje transmitido y ubicando los símbolos en su constelación de referencia. Aunque, la tecnología 5G promete alcanzar esquemas de modulaciones más altas como 512 QAM y 1024 QAM cabe mencionar que aún se encuentra en etapa de desarrollo.

Se tomó como frecuencia de operación los 915 MHz, pertenecientes al espectro de 5G que va desde 450 MHz hasta 6 GHz de tal manera que se pudo comparar y evaluar el desempeño con el estándar WiFi IEEE.802.11b; aunque 5G también ofrece una banda alta de frecuencia de 24.25 GHz hasta 52.60 GHz esta última presenta una menor cobertura y dificultades de propagación entre los objetos.

Se concluye que el uso de un dispositivo de radio definida por software es muy didáctico para poder hacer prácticas de laboratorio de telecomunicaciones en cuanto a sistemas de 5G ya que trabaja directamente en la temática, incorpora elementos necesarios para hacer comprensible el tema y es mucho más sencillo de adquirir y utilizar que un dispositivo de enseñanza tradicional.

## **5.2 Recomendaciones**

Se recomienda para este tipo de simulaciones desarrollados mediante ADALM-PLUTO hacer pruebas con el loopback y las antenas para comprobar las diferencias que existen cuando se hace uso del medio inalámbrico y cableado.

Cuando se trabaja con las antenas asegurarse de que las mismas se encuentren correctamente conectadas y que apunten una a la otra para obtener mejor resultados.

Para las prácticas de modulaciones se recomienda hacer uso de dos dispositivos ADALM-PLUTO, uno para transmitir y otro para recibir; de este modo se obtienen los resultados esperados.

Una vez que se tenga todo el sistema receptor 5G montado en el entorno de Simulink se aconseja que el bloque del controlador automático de ganancia, AGC, cuente un step size de 0.05, averaging length 20 y maximum power gain de 15 de tal manera que se aprecie mucho mejor las figuras de constelaciones después de cada una de las etapas.

## BIBLIOGRAFÍA

- [1] C. L. B. Adolfo Oswaldo Acevedo Borrego, «El desarrollo tecnológico de las comunicaciones,» Revista AHCIET, 2006. [En línea]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=2285983>. [Último acceso: 12 07 2022].
- [2] D. V. Gonzalez, «Implementación del uso de la Radio Definida por Software como,» 14 06 2019. [En línea]. Available: [https://www.researchgate.net/publication/295918680\\_Implementacion\\_de\\_un\\_radio\\_definido\\_por\\_software\\_SDR\\_para\\_la\\_banda\\_de\\_30\\_KHz\\_-\\_40\\_MHz](https://www.researchgate.net/publication/295918680_Implementacion_de_un_radio_definido_por_software_SDR_para_la_banda_de_30_KHz_-_40_MHz). [Último acceso: 12 07 2022].
- [3] P. C. Julio Cerezo, «El impacto del 5G,» Cuadernos de tecnología, [En línea]. Available: <https://evocaimagen.com/cuadernos-tecnologia/cuadernos-evoca-tecnologia-1.pdf>. [Último acceso: 12 07 2022].
- [4] C. García, «Radio Definido por Software usando MATLAB,» Facultad de Ingeniería Eléctrica, 2011. [En línea]. Available: <https://dspace.uclv.edu.cu/bitstream/handle/123456789/4729/Carlos%20Manuel%20Garc%C3%ADa%20Algora.pdf?sequence=1&isAllowed=y>. [Último acceso: 13 07 2022].
- [5] G. M. CHÁIDEZ, «Radio definido, futuro de las comunicaciones inalámbricas,» UNIVERSIDAD AUTÓNOMA DE ZACATECAS, 04 2019. [En línea]. Available: <https://fdocuments.ec/document/radio-definido-por-software-universidad-autnoma-de-.html?page=1>. [Último acceso: 12 07 2022].
- [6] «ADALM-PLUTO,» ANALOG DEVICES, [En línea]. Available: <https://www.analog.com/media/en/news-marketing-collateral/product-highlight/ADALM-PLUTO-Product-Highlight.pdf>. [Último acceso: 12 07 2022].
- [7] «Introducción a la tecnología 5G,» Agencia Española Protección Datos, 05 2020. [En línea]. Available: <https://www.aepd.es/sites/default/files/2020-06/nota-tecnica-privacidad-5g.pdf>. [Último acceso: 12 07 2022].
- [8] C.-F. L. T. K. S. & H.-C. C. HSIN-HUNG CHO, «Integration of SDR and SDN for 5G,» *IEEE Access*, vol. 2, pp. 1196-1202, 21 Octubre 2014.
- [9] I. A. V. L. M. Sc., «Radio Definido por Software,» CIEYT, 16 Mayo 2020. [En línea]. Available: <https://telecomunicaciones.edu.ec/repositorio/articulos-blog/redes/radio-definido-por-software>. [Último acceso: Junio 2022].
- [10] J. G. Oscar Salamanca, «El camino hacia la tecnología 5G,» *Revista Electrónica de Estudios Telemáticos, TÉLÉMATIQUE*, pp. 35--39, 2016.
- [11] ASIET, Telecomunicaciones de América Latina, 06 01 2021. [En línea]. Available: <https://asiet.lat/magazine-digital/del-1g-al-5g-la-evolucion-de-las-redes-moviles-tras-el-cambio-de-generacion/>. [Último acceso: 12 07 2022].
- [12] «Espectro 5G,» GSMA, 07 2019. [En línea]. Available: <https://www.gsma.com/spectrum/wp-content/uploads/2019/10/5G-Spectrum-Positions-SPA.pdf>. [Último acceso: 20 07 2022].
- [13] W. S. H. M. W. a. R. N. A. M. a. S. F. a. I. A. a. A. F. a. J. M. Z. a. Z. M. Ahmad, «5G technology: Towards dynamic spectrum sharing using cognitive radio networks,» *IEEE Access*, vol. 8, pp. 14460--14488, 2020.

- [14] VIAVI Solutions Inc., «VIAVI,» 2022. [En línea]. Available: <https://www.viavisolutions.com/es-es/arquitectura-de-la-tecnologia-5g>. [Último acceso: Julio 2022].
- [15] H. Remmert, «DIGI,» 19 Marzo 2021. [En línea]. Available: <https://es.digi.com/blog/post/5g-network-architecture>. [Último acceso: Julio 2022].
- [16] R. M. A. M. A. I. Cristóbal Escoda, «Innovaciones en arquitectura de red y espectro para el 5G,» *NAE*, 2018.
- [17] P. Gianluca Cornetta, «Demodulación y detección,» Universidad San Pablo, Abril 04 2015. [En línea]. Available: [https://www.cartagena99.com/recursos/alumnos/apuntes/Leccion\\_06.pdf](https://www.cartagena99.com/recursos/alumnos/apuntes/Leccion_06.pdf). [Último acceso: 8 Septiembre 2022].
- [18] T. F. Collins y D. P. & A. M. W. Robin Getz, *SOFTWARE-DEFINED RADIO for ENGINEERS, Mobile Communications*, 2018.

## ANEXOS

### Sincronización de tiempo de símbolo

#### Código del Interpolador

```
% Se definen los coeficientes del interpolador
alpha = 0.5;
InterpFilterCoeff = ...
    [ 0, 0, 1, 0; % Constantes
      -alpha, 1+alpha, -(1-alpha), -alpha; % Lineal
      alpha, -alpha, -alpha, alpha]; % Cuadrático

% Entrada de datos del filtro
ySeq = [y(i); InterpFilterState]; % Línea de retraso actualizada
% Salida del filtro
filtOut = sum((InterpFilterCoeff * ySeq) .* [1; mu; mu^2]);
InterpFilterState = ySeq(1:3); % Guardar los datos de entrada del filtro
```

#### Código del Timing error detector

```
% Cálculos de ZC-TED
if Trigger && all(~TriggerHistory(2:end))
    % Cálculo de muestras medias para valores pares o impares
    t1 = TEDBuffer(end/2 + 1 - rem(N,2));
    t2 = TEDBuffer(end/2 + 1);
    midSample = (t1+t2)/2;
    e = real(midSample)*(sign(real(TEDBuffer(1)))-sign(real(filtOut))) ...
        imag(midSample)*(sign(imag(TEDBuffer(1)))-sign(imag(filtOut)));
else
    e = 0;
end
% Actualizar el buffer TED
switch sum([TriggerHistory(2:end), Trigger])
    case 0
        % Caso cuando no se necesita actualización
    case 1
        % Cambie el búfer TED con regularidad si se activa UN disparo en N
        muestras
        TEDBuffer = [TEDBuffer(2:end), filtOut];
    otherwise % > 1
        % Rellene una muestra faltante si DOS disparan a través de N muestras
        TEDBuffer= [TEDBuffer(3:end), 0, filtOut];
End
```

#### Código del Loop filter

```
% Loop filter
loopFiltOut = LoopPreviousInput + LoopFilterState;
g = e*ProportionalGain + loopFiltOut; % Señal de error en el filtro
LoopFilterState = loopFiltOut;
LoopPreviousInput = e*IntegratorGain;
% Filtro de bucle (alternativa con objetos de filtro)
```

```
lf = dsp.BiquadFilter('SOSMatrix',tf2sos([1 0],[1 -1])); % Se crea el filtro
g = lf(IntegratorGain*e) + ProportionalGain*e; % Señal de erros del filtro
```

### Código del Interpolator control logic

```
% Interpolation Controller with modulo-1 counter
d = g + 1/N;
TriggerHistory = [TriggerHistory(2:end), Trigger];
Trigger = (Counter < d); % Check if a trigger condition
if Trigger % Update mu if a trigger
    mu = Counter / d;
end
Counter = mod(Counter - d, 1); % Update counter
```

### **Sincronización de frecuencia de portadora**

```
classdef (StrictDefaults, TunablesDetermineInactiveStatus)
CarrierSynchronizer < ...
    comm.internal.Helper
% Sincronizador de portadora

    properties (Nontunable)
        %Tipo de modulación
        % Especifique el tipo de modulación como uno de | 'BPSK' |
        % |'QAM' | 'QPSK'|
        Modulation = 'QAM';
    end
switch obj.pPED
    % For loop dentro de cada caso para una mejora adicional del
    rendimiento.
    case 1 % QAM y QPSK
        for k = 1:length(inputC)
            %Error de fase
            phErr = sign(real(previousSample)).*imag(previousSample)...
                - sign(imag(previousSample)).*real(previousSample);
            % Fase acumulada y correcta
            output(k) = inputC(k)*exp(1i*obj.pPhase);

            % Loop Filter
            loopFiltOut = phErr*obj.pIntegratorGain + loopFiltState;
            loopFiltState = loopFiltOut;

            % Sintetizador digital directo implementado como integrador
            DDSOut = DDSPreviousInp + integFiltState;
            integFiltState = DDSOut;
            DDSPreviousInp = phErr*obj.pProportionalGain+loopFiltOut;

            obj.pPhase = obj.pDigitalSynthesizerGain*DDSOut;

            phaseCorrection(k) = obj.pPhase;
            previousSample = output(k);
        end
end
```

## Sincronización de trama

```
classdef (StrictDefaults)FrameSynchronizer < matlab.System &
matlab.system.mixin.Propagates & ...
    matlab.system.mixin.CustomIcon
    % Sincronizador de fotogramas
    % Bloque encargado de convertir entradas fijas o variables en fotogramas
    % de salida de longitud fija.

function setBuffer(obj, x, preambleEndIdx)

    if ~isempty(x)
        % Proceso definido como x
        xLen = length(x);
        linearIdx = (obj.pBufferStartIdx + obj.pNBuffer) + (1:xLen);
        circularIdx = mod(linearIdx - 1, obj.pBufferLength) + 1;
        obj.pBuffer(circularIdx, 1) = x;
        oldNBuffer = obj.pNBuffer;
        obj.pNBuffer = obj.pNBuffer + xLen;
        obj.pBufferEndIdx = circularIdx(end);
        if obj.pNBuffer > obj.pBufferLength
            % Datos al comienzo del buffer
            % Sobrecribir con nuevos datos
            obj.pNBuffer = obj.pBufferLength;
            obj.pBufferStartIdx = circularIdx(end);

            for k = obj.pNPrbIdxBuffer : -1 : 1
                linIdx = obj.pPrbIdxBufferStartIdx - 1 + k;
                cirIdx = mod(linIdx - 1, obj.pPrbIdxBufferLength) +
1;

                if obj.pPrbIdxBuffer(cirIdx) <= circularIdx(end)
                    obj.pPrbIdxBuffer(cirIdx) = 0;
                    obj.pNPrbIdxBuffer = obj.pNPrbIdxBuffer - 1;
                end
            end
        end
    end
end
```

## PRÁCTICA # 1

### “Introducción a la radio definida por software (SDR) Adalm-Pluto”

#### OBJETIVOS

- Crear un sistema de radio definida por software usando del dispositivo Adalm-Pluto para el análisis de una señal WiFi.
- Configurar el dispositivo SDR Adalm-Pluto mediante los drivers correspondientes para que se comunique con la computadora.
- Instalar los complementos de Adalm-Pluto radio a través del apartado de add-ons de Matlab para utilizar los módulos de transmisor y receptor.

#### DIAGRAMA DE BLOQUES DEL SISTEMA

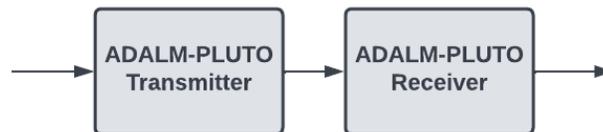


Figura 1 Módulos de transmisión y recepción del Adalm-Pluto radio

Para la siguiente práctica se debe utilizar los bloques de ADALM-PLUTO transmitter y receiver para analizar una señal WiFi en el dispositivo SDR (Figura 1). También se debe recurrir a los bloques de “spectrum analyzer” y “scope” que se encuentran en las librerías de Simulink [1].

#### MARCO TEÓRICO

La combinación de procesamiento digital y RF analógica siempre ha conformado los sistemas de comunicación. En los sistemas modernos de hoy en día, el procesamiento de señales ha progresado hasta tal punto que la mayoría de las funciones de banda base se implementan en el software. La flexibilidad del hardware de RF para ser rediseñado y reconfigurado ha llevado a que una interfaz de radio maneje la mayoría de los sistemas de RF. Normalmente, el front-end de RF está controlado por software en lugar de definido por software. Esta combinación moderna de interfaces de RF flexibles y procesamiento de señales en software ha dado lugar al nacimiento de la radio definida por software.

Con la evolución del sistema de comunicación digital hacia dispositivos altamente complejos, se hizo evidente que se necesitaba una estrategia de divide y vencerás para que el diseño y la implementación de tales sistemas sean factibles y manejables.

En consecuencia, los investigadores dividieron un sistema de comunicación digital en una colección de capas complementarias, y cada capa realizaba una función específica como parte del proceso general de transmisión y recepción de información. Como resultado de esta estrategia de divide y vencerás, los sistemas de comunicación evolucionaron rápidamente hasta convertirse en plataformas altamente capaces que realizan una amplia gama de operaciones, como navegar por la web y enviar correos electrónicos a transmitir contenido multimedia. De hecho, esta estrategia de dividir las operaciones de un sistema de comunicación en capas tuvo tanto éxito que existen comunidades de investigación enteras que solo se enfocan en una de las capas y ninguna en las otras; dan por sentada la información proveniente de las capas por encima y por debajo de su capa.

En general, existen dos modelos para dividir un sistema de comunicación en capas: el modelo de 7 capas de interconexión de sistema abierto (OSI) y el modelo de 5 capas de protocolo de control de transmisión (TCP)/protocolo de Internet (IP) [1].

Ambos modelos poseen aproximadamente la misma funcionalidad, pero el modelo TCP/IP fusiona varias capas superiores en una sola. Centrándonos en el modelo de 5 capas de TCP/ IP, este consta de las siguientes capas, de arriba a abajo:

- Capa de aplicación: Interconecta al usuario con los datos del sistema de comunicación. Por ejemplo, la capa de aplicación incluiría datos que se originen o estén destinados a software que ejecuta navegadores web, clientes de correo electrónico e interfaces de transmisión de medios. Estas aplicaciones generalmente se abordan a través de un socket designado.
- Capa de Transporte: Responsable de transportar los mensajes presentes en la capa de aplicación entre la aplicación cliente y la aplicación servidor. Esta capa garantiza una transmisión de datos fiable.
- Capa de red: Responsable de mover entre los diferentes hosts cada uno de los paquetes presentes en la capa de red. Define el formato de los datagramas y cómo los sistemas finales y los enrutadores actúan sobre el datagrama, así como también determina las rutas que toman los datagramas entre las fuentes y los destinos.
- Capa de enlace: Maneja el problema que se presenta entre varios dispositivos que se conectan de manera directa con el objetivo de poder compartir información. Fiabilidad: Lo cual implica hallar y remediar fallas, así como el direccionamiento de diferentes sistemas de comunicación.
- Capa física: Envía bits individuales desde un sistema de comunicación directamente a otro sistema de comunicación. [2].

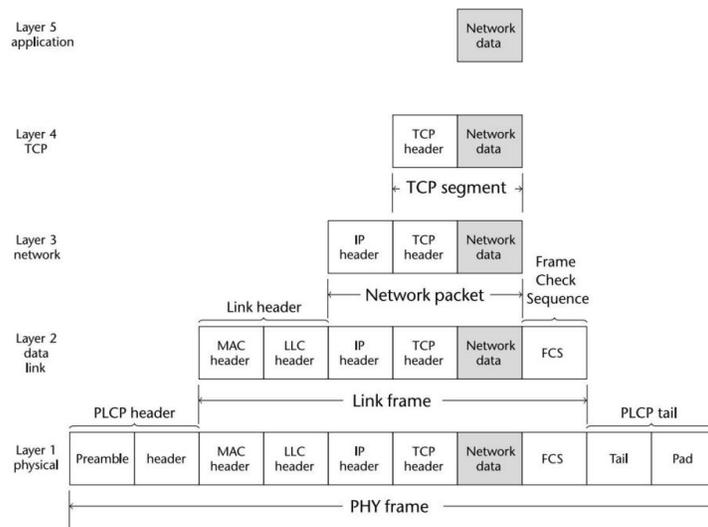


Figura 2: Efectos de la estructura de paquetes SDR, PLCP = Protocolo de convergencia de la capa física

## ACTIVIDADES A DESARROLLAR

### Configuración del SDR Adalm-Pluto

1. Instalar los complementos (add-ons) del Adalm-Pluto Radio, el tutorial de instalación se lo enviará por la plataforma de Aula Virtual.
2. Instalar el driver de comunicación entre el puerto USB de la PC con el dispositivo Adalm-Pluto; este driver se lo puede obtener del siguiente enlace: <https://github.com/analogdevicesinc/plutosdr-m2k-drivers-win/releases>



Figura 3 Driver de comunicación entre el puerto USB de la PC y el dispositivo SDR ADALM-PLUTO

3. Instalar el driver de comunicación con el cheap check del Adalm-Pluto; este driver se lo puede descargar del siguiente link: <https://github.com/analogdevicesinc/libiio/releases>

libiio-0.24.gc4498c2-macOS-11.pkg	143 KB	08 Jul 2022
libiio-0.24.gc4498c2-macOS-11.tar.gz	132 KB	08 Jul 2022
libiio-0.24.gc4498c2-macOS-12-no-libzstd.pkg	142 KB	15 Jul 2022
libiio-0.24.gc4498c2-macOS-12.pkg	142 KB	08 Jul 2022
libiio-0.24.gc4498c2-macOS-12.tar.gz	131 KB	08 Jul 2022
libiio-0.24.gc4498c2-Windows-setup.exe	2.51 MB	08 Jul 2022
Windows-VS-2019-x64.zip	1.44 MB	08 Jul 2022
Windows-VS-2022-x64.zip	1.46 MB	08 Jul 2022
Source code (zip)		08 Jul 2022
Source code (tar.gz)		08 Jul 2022

Figura 4 Driver de comunicación con el cheap check del ADALM-PLUTO

4. Actualizar el firmware manualmente en caso de que no se reconozca el dispositivo en la PC.
5. Descargar el ejemplo de analizador espectral y ejecutar las pruebas. [https://espolec-my.sharepoint.com/:u:/g/personal/dcali\\_espol\\_edu\\_ec/EQQ8X2DQGgdOsnfpKHpkD0B\\_09ODO0VkerPAI2h\\_yKtvG?e=SH9eX0](https://espolec-my.sharepoint.com/:u:/g/personal/dcali_espol_edu_ec/EQQ8X2DQGgdOsnfpKHpkD0B_09ODO0VkerPAI2h_yKtvG?e=SH9eX0)

## INSTRUCCIONES

1. Para la instalación de los drivers, se debe seleccionar todo “Aceptar” y “Continuar”, y una vez instalado correctamente escribimos en la consola de Windows el siguiente comando: `üo_info -s`; para recibir información de los dispositivos SDR que se encuentran conectados actualmente.

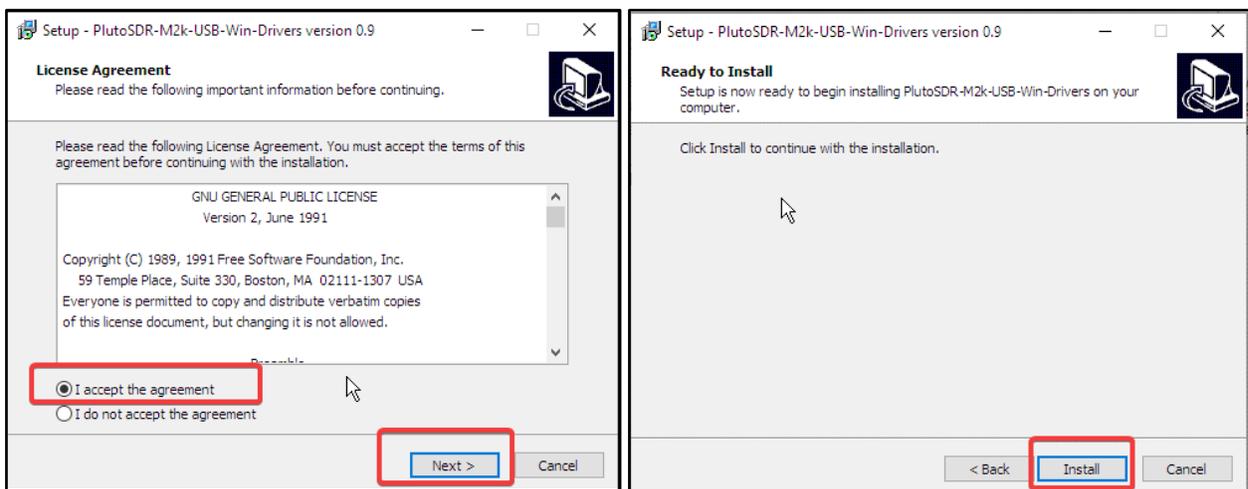


Figura 5 Instalación del driver PlutoSDR-M2k-USB-Win

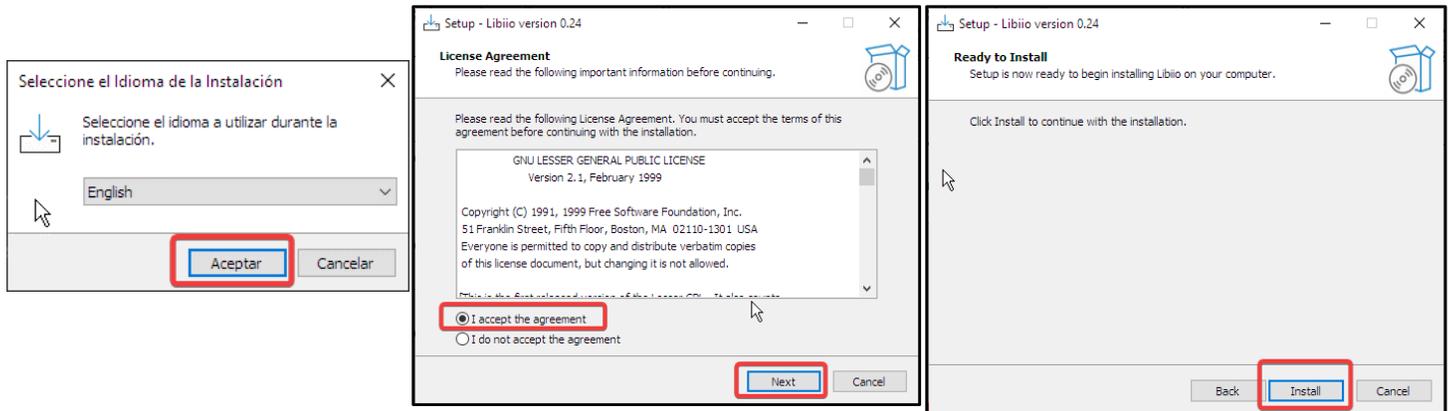


Figura 6 Instalación del driver Libio

2. Abrir Matlab 2017 o posterior, y luego iniciar Simulink con el ejemplo previamente descargado. Verificar que los parámetros de configuración se encuentren correctamente y presionar “Run”.

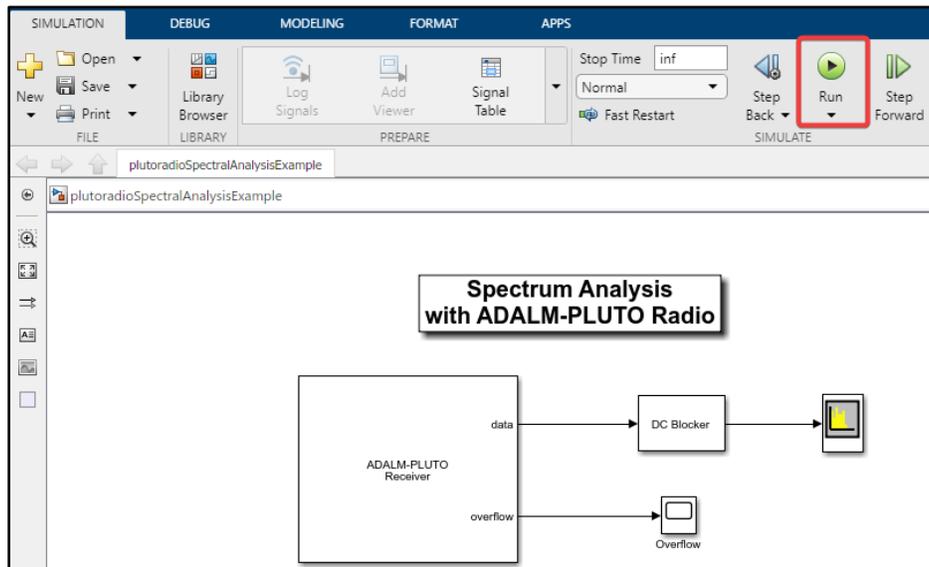


Figura 7 Analizador de espectro con SDR ADALM-PLUTO

3. Agregar elementos en la ventana del analizador de espectro para observar a mayor detalle la medición.

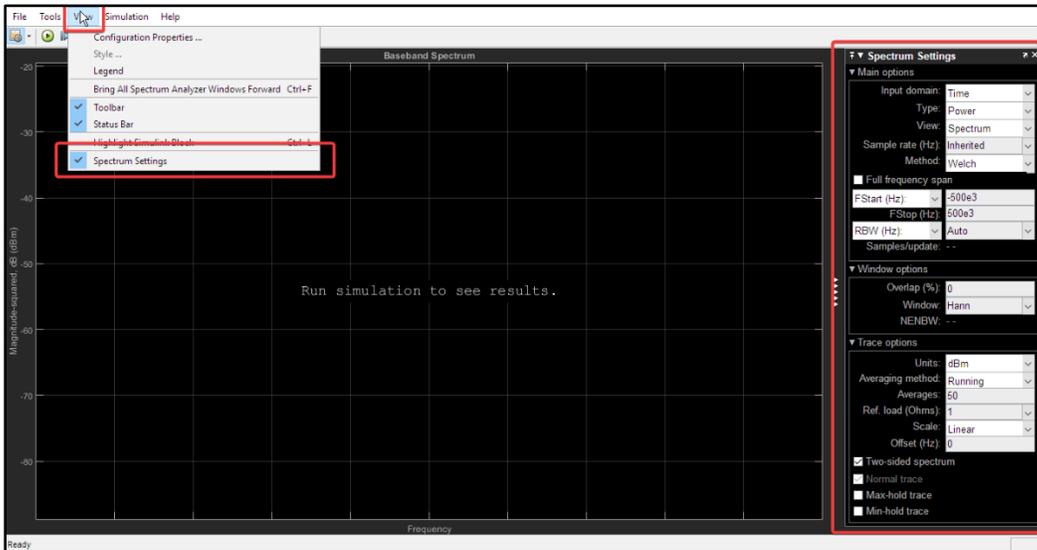


Figura 8 Elementos del analizador de espectro de Simulink

## PRUEBAS A REALIZAR

1. Realizar una captura de la consola de Windows que muestre la información del dispositivo SDR Adalm-Pluto.
2. Muestre el espectro de la señal recibida y el overflow.
3. Señale el ancho de banda y la densidad de potencia.
4. Muestre el espectrograma de la señal recibida.
5. Indique los picos máximos con ayuda de los cursores.
6. Realizar las mediciones del canal y de distorsión.
7. Realizar la gráfica acumulada de la señal.

## PREGUNTAS A CONTESTAR

1. Mencione las principales aplicaciones de la radio definida por software.
2. Explique lo que es la radio cognitiva.
3. ¿Cuál es el ancho de banda máximo del ADALM-PLUTO?
4. ¿Qué datos se obtiene al ejecutar el comando `iio_info -s`?

## BIBLIOGRAFÍA

- [1] R. G. D. P. A. W. Travis Collins, "Software-Defined Radio for Engineers," Analog Devices, 2018.
- [2] Á. Gutiérrez, "Implementación de software denined radop en sistemas de comunicación actuales," 2021. [Online]. Available: <https://biblus.us.es/bibing/proyectos/abreproy/93378/fichero/TFG-3378+GUTI%C3%89RREZ+RIVERA%2C+%C3%81LVARO.pdf>. [Accessed 19 Octubre 2022].

## PRÁCTICA # 2

### “Modulaciones digitales utilizando el ADALM-PLUTO radio”

#### OBJETIVOS:

- Utilizar los bloques de modulador y demodulador digital por desplazamiento de fase y amplitud en cuadratura para la interacción con el ADALM-PLUTO radio.
- Implementar un sistema de comunicación para la transmisión de información digital a través de un canal SISO usando Simulink.
- Analizar la respuesta del sistema usando constelaciones y gráficas de BER vs. SNR.

#### DIAGRAMA DE BLOQUES DEL SISTEMA:



Figura 18 Diagrama de bloques del sistema de comunicación

De acuerdo al diagrama de bloques de a figura 1, se utilizarán los bloques de modulación BPSK, QPSK, y 16-QAM para la transmisión de información digital, posterior utilizaremos los bloques de transmisor y receptor del ADALM-PLUTO radio, finalmente pasará por el bloque de demodulación correspondiente. Se realizarán dos pruebas, el primero utilizando el bloque de canal SISO, el segundo utilizando las antenas del SDR.

#### MARCO TEÓRICO

##### Modulación QAM

La modulación de amplitud en cuadratura (QAM) implica algún tipo de Amplitud modulada (es una señal bidimensional). Las dos dimensiones de la modulación QAM; a saber, los componentes en fase y en cuadratura, son ortogonales entre sí, lo que implica que uno puede esencialmente duplicar la tasa de transmisión de datos de forma libre. [1]

Matemáticamente, si una constelación de señales QAM rectangular consta de M formas de onda únicas, esto podría representarse potencialmente como  $\sqrt{M}$  – PAM transmisiones operando simultáneamente en dimensiones ortogonales. Tenga en cuenta que la señal QAM las constelaciones también podrían tomar la forma de círculos anidados (llamados circular QAM), o cualquier otro patrón geométrico que involucre moduladores ortogonales. Rectangular QAM es un esquema de modulación popular debido a su estructura de receptor relativamente simple, donde

cada dimensión usa un Detector PAM  $\sqrt{M}$ -ario. Para determinar la eficiencia energética de M-QAM, primero definamos la representación matemática de una señal de forma de onda que pertenece a esta forma de onda modulación: [1]

$$s_{ij}(t) = A_i \cdot \cos(\omega_c t) + B_j \cdot \sin(\omega_c t),$$

donde  $\omega_c$  es la frecuencia portadora, y  $A_i$  y  $B_j$  son las frecuencias en fase y en cuadratura niveles de amplitud. Observe cómo se usan las funciones coseno y seno para modular estos niveles de amplitud en dimensiones ortogonales. Ver M-QAM como una señal constelación, las formas de onda de la señal asumirán posiciones tanto en el real como los ejes imaginarios, como se muestra en la siguiente figura [1].

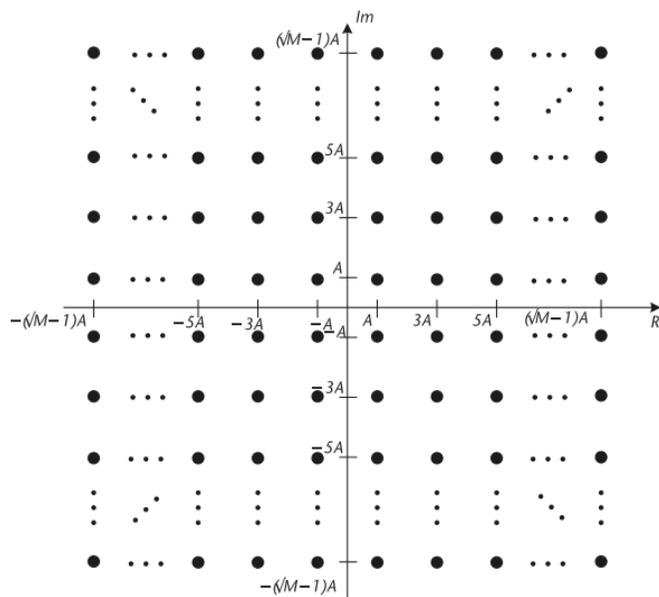


Figura 16 Constelación de señales M-QAM.

Para calcular la eficiencia energética de las modulaciones es:

$$\varepsilon_p = \frac{d_{\min}^2}{E_b}.$$

Para calcular la eficiencia energética de M-QAM se tiene:

$$\varepsilon_{p,M-QAM} = \frac{3!b}{2^b - 1}.$$

## Modulación PSK

Phase shift keying, es una técnica que basa su modulación a través del cambio de su fase (o portadora). Se conoce que la modulación digital PSK posee fases finitas, a las cuales se les proporciona un código binario (irreperible). [1]

Por otra parte, el demodulador que se configura de forma que sea capaz de determinar la fase la fase a la que trabaja el modulador y la obtiene para posteriormente asignarle el símbolo correspondiente, y de esta manera, se consiguen los datos originales. [1]

PSK caracteriza los símbolos por su fase. Matemáticamente, una señal PSK forma de onda está representada por:

$$s_i(t) = A \cos(2\pi f_c t + (2i - 1)\frac{\pi}{m}), \quad \text{for } i = 1, \dots, \log_2 m,$$

donde A es la amplitud,  $f_c$  es la frecuencia portadora y  $(2i - 1)\pi/m$  es el cambio de fase de cada símbolo. PSK presenta un conjunto interesante de compensaciones con PAM y QAM. En los esquemas de modulación de amplitud, la ecualización de canales es una parte importante de decodificando los símbolos correctos. En esquemas PSK, la fase de la señal recibida es mucho más importante que la información de amplitud. Existen varios tipos de esquemas de modulación PSK basados en el número de M valores de fase posibles que se pueden asignar a una forma de onda PSK particular. Uno de los La más popular y robusta es la modulación PSK binaria, o B-PSK. En general, la regla de modulación para B-PSK modulación es la siguiente:

$$\begin{aligned} \text{"1"} &\rightarrow s_1(t) = A \cdot \cos(\omega_c t + \theta) \\ \text{"0"} &\rightarrow s_2(t) = -A \cdot \cos(\omega_c t + \theta) \\ &= A \cdot \cos(\omega_c t + \theta + \pi) \\ &= -s_1(t). \end{aligned}$$

En otras palabras, las dos formas de onda de señal que constituyen una modulación B-PSK esquema están separados en fase por  $\theta$ .

la energía del símbolo de los cuatro símbolos (QPSK) es igual a  $E_{S1} = E_{S2} = E_{S3} = E_{S4} = A^2 T$  Finalmente, despejando la eficiencia energética usando se tiene:

$$\varepsilon_{p,QPSK} = \frac{d_{\min}^2}{\bar{E}_b} = 4,$$

De forma parecida, se tiene para BPSK:

$$\varepsilon_{p,BPSK} = \frac{d_{\min}^2}{\bar{E}_b} = 4.$$

## ACTIVIDADES A DESARROLLAR

Se debe implementar usando bloques de Simulink un sistema de comunicación digital con las siguientes etapas:

- Generación de datos binarios por medio de la función “random integer”.
- Modulador: los datos aleatorios deben ser modulado usando modulación BPSK, QPSK y 16 QAM.
- Canal: a la señal modulada se debe agregar un ruido AWGN, y usar los bloques de ADALM-PLUTO radio conectando al equipo por medio de antenas.
- Demodulador: luego de pasar la señal por el canal debe ser demodulada para obtener los bits transmitidos.
- Gráfica de la constelación.
- Gráfica de BER vs. SNR.

## INSTRUCCIONES

1. Abrir el Simulink y crear un nuevo modelo en blanco.

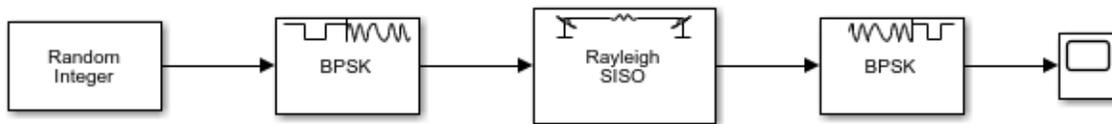


Figura 20 Diagrama de bloques del sistema de comunicación inalámbrico con canal SISO

2. Construir el diagrama de bloques que se muestra en la figura 3 y cambiar los bloques de modulación y demodulación BPSK, QPSK, 16 QAM [2].
3. Construir el diagrama de bloques que se muestra en la figura 4, y cambiar los bloques de modulación y demodulación BPSK, QPSK, 16 QAM.



Figura 21 Diagrama de bloques del sistema de comunicación inalámbrico con los módulos Tx y Rx del ADALM-PLUTO radio

4. Colocar diagramas de constelaciones y scopes para analizar el comportamiento de la señal recibida.

## PRUEBAS A REALIZAR

1. Muestre la constelación recibida con esquema de modulación BPSK con canal SISO y con los módulos del ADALM-PLUTO radio.
2. Muestre la constelación recibida con esquema de modulación QPSK con canal SISO y con los módulos del ADALM-PLUTO radio.
3. Muestre la constelación recibida con esquema de modulación 16-QAM con canal SISO y con los módulos del ADALM-PLUTO radio.
4. Obtenga la gráfica del BER vs. SNR para cada esquema de modulación, tanto con canal SISO y módulos del ADALM-PLUTO radio.

## PREGUNTAS A CONTESTAR

1. ¿Qué es el AGC, y para qué sirve?
2. ¿Explique porque se necesita un bloque AGC cuando se trabaja con los módulos de transmisión y recepción del ADALM-PLUTO radio?
3. Compare las constelaciones de la señal recibida para los dos canales propuestos, y explique las principales diferencias.
4. Compare las gráficas de BER vs. SNR.

## BIBLIOGRAFÍA

- [1] U. d. Valencia, "Modulación digital," [Online]. Available: <https://www.uv.es/~hertz/hertz/Docencia/teoria/Trasmdigital.pdf>. [Accessed 19 Octubre 2022].
- [2] R. G. D. P. A. W. Travis Collins, software-Defined Radio for Engineers, Analog Devices, 2018.

## PRÁCTICA # 3

### “Filtros de acoplamiento y de formación de pulso Sincronización de tiempo de símbolo “

## OBJETIVOS:

- Implementación del filtro formador de pulso y de acoplamiento utilizando el bloque del coseno elevado de Simulink.
- Analizar la constelación y el diagrama de ojo al incrementar el factor de roll-off del filtro.
- Implementación del bloque de sincronización de tiempo de símbolo mediante código de Matlab.

## DIAGRAMA DE BLOQUES DEL SISTEMA



Figura 22 Diagrama de bloques del sistema de comunicación

En la figura 1 podemos observar el diagrama de bloques que se debe implementar, los bloques de modulación y demodulación serán por desplazamiento de fase QPSK, y el filtro de tipo coseno elevado. También se usarán los módulos de transmisor y receptor del ADALM-PLUTO radio en donde se harán las pruebas con las antenas y mediante loopback en el dispositivo SDR. Finalmente, se agregará el bloque de la sincronización de tiempo de símbolo antes de la demodulación.

## MARCO TEÓRICO

### Filtrado coincidente

Al filtrar un símbolo, se reducen las transiciones bruscas de fase y frecuencia resultando en una señal más compacta y espectralmente eficiente. En donde el ancho de banda efectivo de la señal se reduce, principalmente por el sobremuestreo/interpolación que se aplica en el transmisor. Dado que el tiempo y la frecuencia están inversamente relacionados, obtenemos esta reducción en ancho de banda.

Estas implementaciones de la etapa de filtro generalmente aumentarán la muestra y reducirá la resolución de las señales, lo que reduce su ancho de banda efectivo. Sin embargo, el sobremuestreo inherentemente aumenta el llamado área de superficie de un símbolo, haciendo más fácil determinar, ya que tendremos múltiples copias de este en el receptor. Por lo tanto, se está intercambiando capacidad de recuperación por ancho de banda ya que los datos se producirán al mismo tiempo del transmisor, pero no utilizará todo el ancho de banda disponible. Estas operaciones de transiciones de velocidad (upsampling/downsampling) se realizan durante las etapas de filtrado coincidentes ya que es eficiente utilizar un solo filtro para realizar ambas operaciones. El filtro utilizado para generar esta cifra fue un coseno elevado de raíz cuadrada

(SRRC) filtro, que es un filtro común utilizado en los sistemas de comunicación. La respuesta impulsiva es: [1]

$$h(t) = \begin{cases} \frac{1}{\sqrt{T_s}} \left( 1 - \beta + 4\frac{\beta}{\pi} \right), & t = 0 \\ \frac{\beta}{\sqrt{2T_s}} \left[ \left( 1 + \frac{2}{\pi} \right) \sin \left( \frac{\pi}{4\beta} \right) + \left( 1 - \frac{2}{\pi} \right) \cos \left( \frac{\pi}{4\beta} \right) \right], & t = \pm \frac{T_s}{4\beta} \\ \frac{1}{\sqrt{T_s}} \frac{\sin \left[ \pi \frac{t}{T_s} (1 - \beta) \right] + 4\beta \frac{t}{T_s} \cos \left[ \pi \frac{t}{T_s} (1 + \beta) \right]}{\pi \frac{t}{T_s} \left[ 1 - \left( 4\beta \frac{t}{T_s} \right)^2 \right]}, & \text{otherwise} \end{cases}$$

donde  $T_s$  es el periodo del símbolo y  $\beta \in [0, 1]$  es el factor de caída. Cuando se desee conocer el ancho de banda de la señal que se está transmitiendo, es importante hacer uso de alguna herramienta matemática para poder determinarlo. La densidad espectral de potencia  $P_x(f)$ , se la puede hallar mediante la siguiente ecuación.

$$P_x(f) = E_x |G_{tx}(f)|^2$$

Donde se asume que la constelación de señales a ser transmitidas está normalizada a la unidad de energía. [1]

### Una forma de pulso común

La función de coseno elevado es uno de los pulsos de Nyquist más comunes que existen (a más de la función sinc), la cual posee esta respuesta en el dominio de la frecuencia:

$$G_{rc}(f) = \begin{cases} T, & 0 \leq |f| \leq \frac{1 - \alpha}{2T} \\ \frac{T}{2} \left[ 1 + \cos \frac{\pi T}{\alpha} \left( |f| - \frac{1 - \alpha}{2T} \right) \right], & \frac{1 - \alpha}{2T} \leq |f| \leq \frac{1 + \alpha}{2T} \\ 0, & |f| > \frac{1 + \alpha}{2T} \end{cases}$$

Donde se sabe que el factor  $\alpha$  es el que representa el exceso de ancho de banda, es conocido como factor de rolloff [1].

### Filtro formador de pulso

## IMPLEMENTACIÓN DEL FILTRO DE FORMACIÓN DE PULSOS Y UPSAMPLING EN EL TRANSMISOR

Cuando se hace uso de procesamiento de señales multi-tasa; una vez resuelto el upsampling de  $s[m]$  por  $L$ , y posteriormente filtrarlo con  $g_{tx}[n]$ , se tiene finalmente:

$$c[n] = \sum_{m=-\infty}^{\infty} s[m]g_{tx}[n - mL]$$

### IMPLEMENTACIÓN DEL FILTRO DE FORMACIÓN DE PULSOS Y DOWNSAMPLING EN EL RECEPTOR

Cuando se hace uso de procesamiento de señales multi-tasa; una vez resuelto el downsampling de  $z[n]$ , y posteriormente filtrarlo con  $g_{rx}[n]$ , se tiene finalmente: [1]

$$y[n] = \sum_{m=-\infty}^{\infty} z[m]g_{rx}[nM - m]$$

### **ACTIVIDADES A DESARROLLAR**

1. Se debe implementar usando bloques de Simulink un sistema de comunicación digital con las siguientes etapas:
  - Generación de datos binarios por medio de la función “random integer”.
  - Modulador: los datos aleatorios deben ser modulado usando modulación QPSK.
  - Canal: usar los bloques de ADALM-PLUTO radio conectando al equipo por medio de antenas.
  - Demodulador: luego de pasar la señal por el canal debe ser demodulada para obtener los bits transmitidos.
  - Gráfica de la constelación.
  - Diagrama de ojos.
2. Se debe crear un bloque de simulink correspondiente a la sincronización del tiempo de símbolo mediante código desarrollado en Matlab.

### **INSTRUCCIONES**

#### **Filtro de acoplamiento y formación de pulsos**

1. Abrir el Simulink y crear un nuevo modelo en blanco.
2. Construir el diagrama de bloques que se muestra en la figura 2 Variar el factor de roll-off del filtro en valores diferentes, y mostrar capturas de diagrama de ojo y constelaciones. [2]

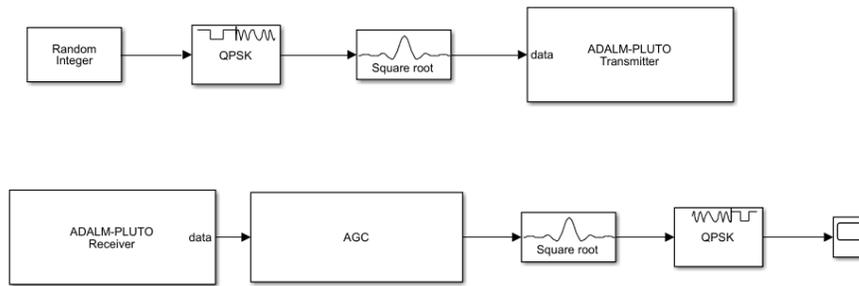


Figura 23 Diagrama de bloques del sistema de comunicación

## Sincronizador de tiempo de símbolo

1. Completar el siguiente script para formar el bloque de la sincronización del tiempo de símbolo (desde la línea 399 hasta 452)

```

% Timing error detector (TED)
switch obj.TimingErrorDetector
    case 'Zero-Crossing (decision-directed)'
        e = ZeroCrossingTED(obj, intOut);
    case 'Gardner (non-data-aided)'
        e = GardnerTED(obj, intOut);
end

switch sum([obj.pStrobeHistory(2:end), obj.pStrobe])
    case 0
        % Omite la muestra actual si NO hay luz estroboscópica en N
muestras
    case 1
        obj.pTEDBuffer = [obj.pTEDBuffer(2:end), intOut];
    otherwise % > 1
        % Rellene una muestra que falta si DOS luces estroboscópicas
en N muestras
        obj.pTEDBuffer = [obj.pTEDBuffer(3:end), 0, intOut];

end % Fin de TED

% Loop filter
v = loopFilter(obj, e);

% Controlador de interpolación
interpControl(obj, v);

end
y = obj.pSymbolHolder(1:obj.pNumStrobe, 1);
end

function y = interpFilter(obj, x)
% nterpolador parabólico por partes en estructura de farrow
xSeq = [x; obj.pInterpFilterState];
y = sum((obj.pInterpFilterCoeff * xSeq) .* [1; obj.pMu; obj.pMu^2]);

```

```

    obj.pInterpFilterState = xSeq(1:3);
end

function v = loopFilter(obj, e)
    loopFiltOut = obj.pLoopPreviousInput + obj.pLoopFilterState;
    v = e*obj.pProportionalGain + loopFiltOut;
    obj.pLoopFilterState = loopFiltOut;
    obj.pLoopPreviousInput = e*obj.pIntegratorGain;
end

function interpControl(obj, v)

    W = v + 1/obj.pSPS; % W debe ser pequeño cuando está bloqueado(masculino)
Bu
    obj.pStrobeHistory = [obj.pStrobeHistory(2:end), obj.pStrobe];
    obj.pStrobe = (obj.pNCOCCounter < W); % Compruebe si una luz
estroboscópica
    if obj.pStrobe % Actualizar mu si es una luz estroboscópica
        obj.pMu = obj.pNCOCCounter / W;
    end
    obj.pNCOCCounter = mod(obj.pNCOCCounter - W, 1); % Actualizar contador
end

```

2. Comprobar el correcto funcionamiento del bloque para la sincronización del tiempo de símbolo.

## PREGUNTAS A CONTESTAR

- Realice la simulación usando los siguientes ajustes de parámetros de filtro: “1”, “0.75”, “0.5”, “0.25” y “0” para QPSK. Recopilar capturas de pantalla tanto de constelaciones como diagrama de ojos.
- Explique mediante simulación la relación que existe entre el ISI y las variaciones del factor de roll-off.
- Explique mediante simulación la relación que existe entre el ancho de banda y las variaciones del factor de roll-off.
- Muestre que la construcción del bloque de simulink para la sincronización de tiempo de símbolo no tenga errores.

## BIBLIOGRAFÍA

- [1] R. G. D. P. A. W. Travis Collins, Software-Defined Radio for Engineers, Analog Devices, 2018.
- [2] M. Nieves, Diseño de Filtros de Acoplamiento, Sevilla: Universidad de Sevilla, 2010.

## PRÁCTICA #4

## “Sincronización de frecuencia de portadora”

### OBJETIVOS:

- Analizar la etapa de sincronización de portadora por medio de un ejemplo desarrollado en Matlab con ayuda del SDR ADALM-PLUTO radio.
- Desarrollar la función correspondiente a la sincronización de portadora mediante el uso del entorno Matlab.
- Analizar la constelación después de la etapa de sincronización de frecuencia de portadora.

### DIAGRAMA DE BLOQUES

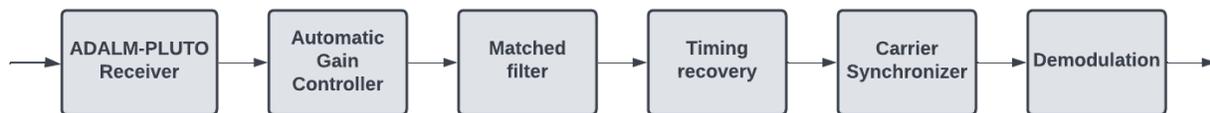


Figura 24 Diagrama de bloques del sistema

En la siguiente práctica se debe agregar a nuestro sistema receptor la etapa de sincronización de portadora, la misma que ocurre antes de la demodulación y después de la sincronización del tiempo de símbolo tal y como se aprecia en la figura 1. La etapa de transmisión ya es proporcionada en la práctica. [1]

## MARCO TEÓRICO

### Compensación de portadoras

Los nodos de recepción y transmisión son generalmente dos unidades distintas y espacialmente separadas. Por lo tanto, habrá cambios de frecuencia relativa entre sus LO debido a efectos naturales como impurezas, ruido eléctrico y diferencias de temperatura, entre otros. Dado que estas diferencias también pueden ser relativamente dinámicas, los LO se compensarán entre sí.

Estas compensaciones pueden contener ruido de fase aleatorio, compensación de frecuencia, desviación de frecuencia y compensaciones de fase inicial. Sin embargo, por simplicidad, solo modelaremos este desplazamiento como un valor fijo. Esta es una suposición razonable en la escala de tiempo de las comunicaciones de RF. Al considerar los osciladores comerciales, la compensación de frecuencia se da en partes por millón (PPM), que podemos traducir a una compensación de portadora máxima para una frecuencia determinada.

En el caso de Pluto SDR, el LO interno está clasificado en 25 PPM (2 PPM cuando está calibrado) y podemos usar la siguiente fórmula para relacionar el desplazamiento máximo de la portadora  $f$  con nuestra frecuencia de operación de la portadora  $f_c$ : [1]

$$f_{o,max} = \frac{f_c \times PPM}{10^6}$$

Matemáticamente se puede modelar una señal fuente corrupta en la banda base  $s(k)$  con un desplazamiento de frecuencia portadora de  $f_o$  (o  $\omega_o$ ) como: [1]

$$r(k) = s(k)e^{j(2\pi f_o k T + \theta)} + n(k) = s(k)e^{j(\omega_o k T + \theta)} + n(k).$$

donde  $n(k)$  es un proceso aleatorio gaussiano de media cero,  $T$  es el período del símbolo,  $\theta$  es la fase portadora, y  $\omega_o$  la frecuencia angular [1].

La recuperación de la portadora a veces se define como recuperación de la fase de la portadora o recuperación de la frecuencia de la portadora. Por lo general, todos tienen el mismo objetivo de proporcionar una constelación estable a la salida del sincronizador [1].

### **Compensación de desplazamiento de frecuencia**

Hay muchas maneras diferentes de diseñar un receptor inalámbrico, utilizando muchas técnicas de recuperación y disposición de algoritmos. En esta sección consideraremos desplazamiento de frecuencia primero y luego proceda a administrar las tareas de sincronización restantes. [1]

La transmisión de señales en una banda sin licencia, como 2,4 GHz, puede producir un desplazamiento máximo de 120 kHz entre las radios. Ya que esto es bastante grande rango desarrollaremos una técnica de compensación de frecuencia de dos etapas separada en corrección de frecuencia gruesa y fina. Este diseño es favorable, ya que puede reducir el tiempo de convergencia o bloqueo para la estimación de la portadora relativa [1].

#### Corrección de frecuencia aproximada

Cuando se habla de corrección de frecuencia gruesa, existen 2 tipos que son:

Corrección asistida por datos: Para calcular el desplazamiento respecto a la portadora hace uso de una señal recibida y está basada en correlación. A pesar de que su estimación posee buena precisión, se ve limitado por la longitud de preámbulo (su longitud es inversamente proporcional al rendimiento del sistema)

Corrección a ciegas: Es un método que es aplicable en la realidad, por ello suelen ser más beneficioso que el método anteriormente mencionado. Aunque suelen ser mayormente usados para circuitos abiertos.

Formalmente, esta estimación de frecuencia se puede escribir en una sola ecuación como:

$$\hat{f}_o = \frac{1}{2TK} \arg \left| \sum_{k=0}^{K-1} r^M(k) e^{-j2\pi kT/K} \right|$$

## Corrección de frecuencia fina

Después de la corrección de frecuencia aproximada (CFC), todavía habrá una compensación basada en la resolución configurada elegida fr. La corrección de frecuencia fina (FFC), también llamada corrección de fase de la portadora debe producir una constelación estable para una eventual demodulación. Esencialmente, esto conducirá a cero el desplazamiento de frecuencia restante de la señal recibida. Podemos describir esta corrección como la producción de una constelación estable debido a cómo los efectos de compensación de frecuencia fina se examinan típicamente con un diagrama de constelación. [1]

## Análisis de rendimiento

Para evaluar el rendimiento de la sincronización se pueden considerar una serie de variables. Estos incluyen, entre otros, el tiempo de bloqueo, el rango de activación efectivo y la magnitud del vector de error convergente (EVM). Estas métricas deben equilibrarse de manera que satisfagan las necesidades de un diseño específico, ya que chocarán entre sí. El valor normalizado retardo de bloqueo de frecuencia: [1]

$$t_{\Delta,Max} \sim \frac{32\xi^2}{B_{Loop}}$$

## **ACTIVIDADES A DESARROLLAR**

- Se debe desarrollar un ejemplo que permita comprender la sincronización de portadora en un sistema receptor.
- Se debe implementar mediante código la función de sincronización de portadora y convertirlo en un bloque de simulink. [2]

## **INSTRUCCIONES**

1. Abrir un nuevo script en Matlab y desarrollar el siguiente código correspondiente al ejemplo de la sincronización de portadora:

```
clear all; close all;
%% General system details
sampleRateHz = 1e6; samplesPerSymbol = 1; frameSize = 2^10;
numFrames = 10; numSamples = numFrames*frameSize;
DampingFactors = [3,1.3];
NormalizedLoopBandwidths = [0.03,0.24];
%% Generate symbols
order = 4; data = pskmod(randi([0 order-1], numSamples, 1),order,0); % QPSK
%% Configure LF and PI
LoopFilter = dsp.IIRFilter( ...
    'Structure', 'Direct form II transposed', ...
    'Numerator', [1 0], 'Denominator', [1 -1]);
Integrator = dsp.IIRFilter(...
    'Structure', 'Direct form II transposed', ...
    'Numerator', [0 1], 'Denominator', [1 -1]);
```

```

phErrs = zeros(numSamples,2);
index = 1;
for NormalizedLoopBandwidth = NormalizedLoopBandwidths
    DampingFactor = 1.3;
%for DampingFactor = DampingFactors
    %% Calculate range estimates
    NormalizedPullInRange = min(1, 2*pi*sqrt(2)*DampingFactor*...
        NormalizedLoopBandwidth);
    MaxFrequencyLockDelay = (4*NormalizedPullInRange^2)/...
        (NormalizedLoopBandwidth)^3;
    MaxPhaseLockDelay = 1.3/(NormalizedLoopBandwidth);
    %% Impairments
    snr = 25;
    frequencyOffsetHz = sampleRateHz*0.01;% (NormalizedPullInRange);
    noisyData = awgn(data,snr);% Add noise
    % Add frequency offset to baseband signal
    freqShift =
exp(1i.*2*pi*frequencyOffsetHz./sampleRateHz*(1:numSamples)).';
    offsetData = noisyData.*freqShift;
    %% Calculate coefficients for FFC
    PhaseRecoveryLoopBandwidth = NormalizedLoopBandwidth*samplesPerSymbol;
    PhaseRecoveryGain = samplesPerSymbol;
    PhaseErrorDetectorGain = order/2; % QPSK=2 BPSK=1
    DigitalSynthesizerGain = -1;
    theta = PhaseRecoveryLoopBandwidth/...
        ((DampingFactor + 0.25/DampingFactor)*samplesPerSymbol);
    d = 1 + 2*DampingFactor*theta + theta*theta;
    % K1
    ProportionalGain = (4*DampingFactor*theta/d)/...
        (PhaseErrorDetectorGain*PhaseRecoveryGain);
    % K2
    IntegratorGain = (4/samplesPerSymbol*theta*theta/d)/...
        (PhaseErrorDetectorGain*PhaseRecoveryGain);
    %% Correct carrier offset
    output = zeros(size(offsetData));
    Phase = 0; previousSample = complex(0);
    LoopFilter.release();Integrator.release();
    for k = 1:length(offsetData)-1
        % Complex phase shift
        output(k) = offsetData(k+1)*exp(1i*Phase);
        % PED
        phErr = sign(real(previousSample)).*imag(previousSample)...
            - sign(imag(previousSample)).*real(previousSample);
        phErrs(k,index) = phErr;
        %phErr = sign(real(previousSample))*imag(previousSample);
        % Loop Filter
        loopFiltOut = step(LoopFilter,phErr*IntegratorGain);
        % Direct Digital Synthesizer
        DDSOut = step(Integrator,phErr*ProportionalGain + loopFiltOut);
        Phase = DigitalSynthesizerGain * DDSOut;
        previousSample = output(k);
    end
    scatterplot(output(end-1024:end-10));title('');grid on;
    index = index + 1;
end
%%
f1 = figure(1);

```

```

plot(phErrs(:,2));%hold on;
xlim([0 1000]);grid on;xlabel('Time');ylabel('PED Error');
ylim([-1.1 1.1]);
f2 = figure(2);
plot(phErrs(:,1));%hold off;
xlim([0 1000]);grid on;xlabel('Time');ylabel('PED Error');
ylim([-1.1 1.1]);

```

2. Abrir un nuevo script en Matlab y completar el código para formar el bloque de sincronización de portadora (desde la línea 153 hasta la 246 del código proporcionado)

```

switch obj.pPED
    % For loop dentro de cada caso para una mejora adicional del
    rendimiento.
    case 1 % QAM y QPSK
        for k = 1:length(inputC)
            %Error de fase
            phErr = sign(real(previousSample)).*imag(previousSample)...
                - sign(imag(previousSample)).*real(previousSample);
            % Fase acumulada y correcta
            output(k) = inputC(k)*exp(1i*obj.pPhase);

            % Loop Filter
            loopFiltOut = phErr*obj.pIntegratorGain + loopFiltState;
            loopFiltState = loopFiltOut;

            % Sintetizador digital directo implementado como integrador
            DDSOut = DDSPreviousInp + integFiltState;
            integFiltState = DDSOut;
            DDSPreviousInp = phErr*obj.pProportionalGain+loopFiltOut;

            obj.pPhase = obj.pDigitalSynthesizerGain*DDSOut;

            phaseCorrection(k) = obj.pPhase;
            previousSample = output(k);
        end
    case 2 % BPSK
        for k = 1:length(inputC)
            % Encontrar error de fase
            phErr = sign(real(previousSample))*imag(previousSample);

            % Fase acumulada y corregida
            output(k) = inputC(k)*exp(1i*obj.pPhase);

            % Loop Filter
            loopFiltOut = phErr*obj.pIntegratorGain + loopFiltState;
            loopFiltState = loopFiltOut;

            % Sintetizador digital directo implementado como integrador
            DDSOut = DDSPreviousInp + integFiltState;
            integFiltState = DDSOut;
            DDSPreviousInp = phErr*obj.pProportionalGain+loopFiltOut;

```

```

    obj.pPhase = obj.pDigitalSynthesizerGain*DDSOut;

    phaseCorrection(k) = obj.pPhase;
    previousSample = output(k);
end
end
% Actualice la muestra anterior y la fase rote según lo desee
output = output*exp(1i*obj.pActualPhaseOffset);

% Cambio de signo para convertir de valor de corrección a estimación
phaseEstimate = -real(phaseCorrection+obj.pActualPhaseOffset);

% Actualización de estados
obj.pLoopFilterState = loopFiltState;
obj.pIntegFilterState = integFiltState;
obj.pPreviousSample = complex(previousSample);
obj.pDDSPreviousInput = DDSPreviousInp;
end

% Calcule las ganancias de bucle y el desplazamiento personalizado
function CalculateLoopGains(obj)

% Calcule las ganancias de los bucles y el coeficiente de filtro
PhaseRecoveryLoopBandwidth = cast(...
    obj.NormalizedLoopBandwidth *obj.SamplesPerSymbol, ...
    obj.pInputDataType);

% K0
PhaseRecoveryGain = cast(obj.SamplesPerSymbol,...
    obj.pInputDataType);

theta = PhaseRecoveryLoopBandwidth/...
    ((obj.DampingFactor + 0.25/obj.DampingFactor)*...
    obj.SamplesPerSymbol);

d = 1 + 2*obj.DampingFactor*theta + theta*theta;

% K1
obj.pProportionalGain = cast(...
    (4*obj.DampingFactor*theta/d)/...
    (obj.pPhaseErrorDetectorGain*PhaseRecoveryGain), ...
    obj.pInputDataType);
% K2
obj.pIntegratorGain = cast(...
    (4*theta*theta/d)/...
    (obj.pPhaseErrorDetectorGain*PhaseRecoveryGain), ...
    obj.pInputDataType);

% Ajuste de compensación de fase personalizado
if strcmp(obj.ModulationPhaseOffset, 'Auto')
    obj.pActualPhaseOffset = zeros(1,obj.pInputDataType);
end
end

```

## **PRUEBAS A REALIZAR**

- Mostrar capturas de la constelación después de realizar la sincronización de portadora en el ejemplo.
- Mostrar capturas del error en la detección de fase y la corrección del mismo durante la ejecución del código del ejemplo.
- Mostrar que el bloque de la sincronización de portadora funcione correctamente.

## **PREGUNTAS A CONTESTAR**

1. Realice un código en Matlab que muestre una estimación de frecuencia simple con un verdadero desplazamiento de frecuencia.
  - a. Tonos generados a partir de una señal sinusoidal y transformada de Hilbert.
  - b. Estimación de frecuencia de los tonos generados
2. Realice un código en Matlab que compare las señales en el dominio de la frecuencia con y sin desplazamiento de frecuencia.
3. Realice un código de Matlab que muestre la rotación de los símbolos para un esquema de modulación BPSK, y explique ¿qué relación mantiene con el desplazamiento de portadora.

## **BIBLIOGRAFÍA**

- [1] R. G. D. P. A. W. Travis Collins, Software-Defined Radio for Engineers, Analog Devices, 2018.
- [2] G. González, Aspectos de sincronización en frecuencia, Universidad Nacional del Sur, 2012.

## **PRÁCTICA # 5**

### **“Sincronización de trama y codificación de canal”**

#### **OBJETIVOS:**

- Analizar la etapa de sincronización de trama y codificación de canal por medio de un ejemplo desarrollado en Matlab con ayuda del SDR ADALM-PLUTO radio.
- Desarrollar la función correspondiente a la sincronización de trama mediante el uso del entorno Matlab.
- Analizar la constelación después de ejecutar la sincronización de trama y codificación de canal.

## DIAGRAMA DE BLOQUES DEL SISTEMA



Figura 25 Diagrama de bloques del sistema

Como se muestra en la figura 1 se debe implementar el bloque de sincronización de trama después de los bloques creados con las prácticas anteriores (sincronización de tiempo de símbolo y portadora). El sistema transmisor se mantiene tal cual se ha venido trabajando.

## MARCO TEÓRICO

### Sincronización de trama

El método común para determinar el inicio de una trama dado es con el uso de marcadores, incluso en redes cableadas. Sin embargo, en el caso de las señales inalámbricas, este problema se vuelve más difícil, como se ve en la Figura 8.2, que en realidad usa un marcador. Debido al alto grado de contenido de ruido en la señal, las secuencias de preámbulo diseñadas específicamente se agregan a las tramas antes de la modulación. Estas secuencias normalmente se conocen exactamente en el receptor y tienen ciertas cualidades que hacen que la estimación de fotogramas sea precisa.

Consideremos un conjunto de  $N$  sucesiones binarias diferentes  $b_n$ , donde  $n \in [1, \dots, N]$ , cada uno de longitud  $L$ . Dada una secuencia binaria adicional  $d$ , queremos determinar qué tan similar es  $d$  a las  $N$  secuencias existentes. El uso de una correlación cruzada proporcionarnos la estimación adecuada, que realizamos como:

$$C_{d,b}(k) = \sum_m d^*(m)b_n(m+k),$$

Las secuencias comunes utilizadas en los preámbulos para las comunicaciones de banda estrecha son Códigos Barker. Los códigos Barker se utilizan porque tienen propiedades de autocorrelación únicas que tienen una correlación mínima o ideal fuera de las horas pico. Específicamente, tales códigos o las secuencias  $a(i)$  tienen funciones de autocorrelación definidas como:

$$c(k) = \sum_{i=1}^{N-k} a(i)a(i+k),$$

Tal que,

$$|c(v)| \leq 1, \quad 1 \leq v < N.$$

## **Codificación de canal**

### Codificación de repetición

Uno de los bloques de construcción clave de cualquier sistema de comunicación es la corrección de errores de reenvío (FEC), donde se agregan datos redundantes al flujo transmitido para hacerlo más resistente a los errores del canal. Hay muchos tipos de técnicas FEC, como el enfoque de codificación de repetición, donde cada bit transmitido se repite varias veces. [1]

### Intercalado

Un código de repetición es una de varias herramientas útiles para un ingeniero de sistemas de comunicación con el fin de mejorar una transmisión de datos robusta. Sin embargo, a veces no es suficiente, ya que no soluciona el problema cuando se corrompe una gran cantidad de datos en bloques contiguos. Por ejemplo, si un transmisor envía el flujo de datos "101101", un codificador de repetición con un factor de repetición de 4 producirá: [1]

111100001111111100001111,

El intercalado es un enfoque en el que los datos binarios se reordenan de manera que la correlación existente entre los bits individuales dentro de una secuencia específica se reduce significativamente [1].

### Codificación

Además de intercalar múltiples copias de datos, podemos codificar los datos en secuencias alternativas que introducen redundancia. Una propiedad única de muchos esquemas de codificación es la capacidad de introducir redundancia sin aumentos en el tamaño de los datos sin orden de números enteros. Por ejemplo, en el caso de la codificación repetitiva que duplica cada bit con  $R = 2$ , este número generalmente se invierte en las discusiones de FEC ya que la tasa de un esquema de codificación convolucional puede introducir tasas más cercanas a 1. [1]

## **ACTIVIDADES A DESARROLLAR**

- Se debe desarrollar un ejemplo en Matlab que permita comprender la sincronización de trama en un sistema receptor.
- Se debe implementar mediante código la función de sincronización de trama y convertirlo en un bloque de simulink. [2]

## INSTRUCCIONES

1. Abrir un nuevo script en Matlab y desarrollar el siguiente ejemplo:

```
%% General system details
sampleRateHz = 1e6; samplesPerSymbol = 8; numFrames = 1e2;
modulationOrder = 2; filterSymbolSpan = 4;
barkerLength = 26; % Must be even
%% Impairments
snr = 15;
%% Generate symbols and Preamble
bits = randi([0 3], modulationOrder*1e3,1);
hBCode = comm.BarkerCode('Length',7,'SamplesPerFrame', barkerLength/2);
barker = hBCode()>0; frame=[barker;barker;bits];frameSize = length(frame);
% Modulate
modD = comm.DBPSKModulator(); bMod = clone(modD);
modulatedData = modD(frame>0);
%% Add TX/RX Filters
TxFlt = comm.RaisedCosineTransmitFilter(...
    'OutputSamplesPerSymbol', samplesPerSymbol,...
    'FilterSpanInSymbols', filterSymbolSpan);
RxFlt = comm.RaisedCosineReceiveFilter(...
    'InputSamplesPerSymbol', samplesPerSymbol,...
    'FilterSpanInSymbols', filterSymbolSpan,...
    'DecimationFactor', samplesPerSymbol);
RxFltRef = clone(RxFlt);
%% Setup visualization object(s)
hts1 = dsp.TimeScope('SampleRate', sampleRateHz,'TimeSpan', ...
    frameSize*2/sampleRateHz);
hAP = dsp.ArrayPlot;hAP.YLimits = [-3 35];
%% Demodulator
demod = comm.DBPSKDemodulator;
%% Model of error
BER = zeros(numFrames,1);PER = zeros(numFrames,1);
for k=1:numFrames
    % Insert random delay and append zeros
    delay = randi([0 frameSize-1-TxFlt.FilterSpanInSymbols]);
    delayedSignal = [zeros(delay,1); modulatedData;...
        zeros(frameSize-delay,1)];
    % Filter signal
    filteredTXDataDelayed = TxFlt(delayedSignal);
    % Pass through channel
    noisyData = awgn(filteredTXDataDelayed,snr,'measured');
    % Filter signal
    filteredData = RxFlt(noisyData);
    % Visualize Correlation
    hts1(filteredData);pause(0.1);
    % Remove offset and filter delay
```

```

frameStart = delay + RxFlt.FilterSpanInSymbols + 1;
frameHatNoPreamble = filteredData(frameStart:frameStart+frameSize-1);
% Demodulate and check
dataHat = demod(frameHatNoPreamble);
demod.release(); % Reset reference
BER(k) = mean(dataHat-frame); PER(k) = BER(k)>0;
end
% Result
fprintf('PER %2.2f\n', mean(PER));

```

## 2. Completar el siguiente script para formar el bloque de sincronización de trama

```

classdef (StrictDefaults)FrameSynchronizer < matlab.System &
matlab.system.mixin.Propagates & ...
    matlab.system.mixin.CustomIcon

    properties (Nontunable)
        Input = 'Symbol';
        OutputFrameLength = 100
        PreambleLength = 4

    end

    properties (Constant, Access = private)
        pNumOutFramesInBuffer = 5
    end

    properties (Access = private, Nontunable)
        pIsBitInput = false % Entrada='Bit'
        pBufferLength % Longitud del buffer
        pPrbIdxBufferLength % Longitud del búfer de índice de preámbulo
        pStepFcn % Puntero para realizar sincronización de
cuadros o no (transmisión)
    end

    properties (Access = private)
        pPrbLenOffset % Desplazamiento para obtener el índice de
inicio del preámbulo del índice final
        pBuffer % Buffer circular interno
        pNBuffer % Cantidad de elementos del buffer
        pBufferStartIdx % Índice basado en 0 (primer elemento)
        pBufferEndIdx % Índice basado en última posición (décimo)
        pPrbIdxBuffer % Inicio del preámbulo del buffer
        pNPrbIdxBuffer % Cantidad de preámbulos en buffer
        pPrbIdxBufferStartIdx % Índice de primera posición (primer
elemento)
    end

    properties (Constant, Hidden)
        InputSet = matlab.system.StringSet({ ...
            'Symbol', 'Bit'});
    end

----- Línea 252 en adelante-----
function setBuffer(obj, x, preambleEndIdx)

```

```

if ~isempty(x)
    % Process x
    xLen = length(x);
    linearIdx = (obj.pBufferStartIdx + obj.pNBuffer) + (1:xLen);
    circularIdx = mod(linearIdx - 1, obj.pBufferLength) + 1;
    obj.pBuffer(circularIdx, 1) = x;
    oldNBuffer = obj.pNBuffer;
    obj.pNBuffer = obj.pNBuffer + xLen;
    obj.pBufferEndIdx = circularIdx(end);
    if obj.pNBuffer > obj.pBufferLength
        % Datos al comienzo del buffer
        % Sobrecribir con nuevos datos
        obj.pNBuffer = obj.pBufferLength;
        obj.pBufferStartIdx = circularIdx(end);

        for k = obj.pNPrbIdxBuffer : -1 : 1
            linIdx = obj.pPrbIdxBufferStartIdx - 1 + k;
            cirIdx = mod(linIdx - 1, obj.pPrbIdxBufferLength) +
1;

            if obj.pPrbIdxBuffer(cirIdx) <= circularIdx(end)
                obj.pPrbIdxBuffer(cirIdx) = 0;
                obj.pNPrbIdxBuffer = obj.pNPrbIdxBuffer - 1;
            end
        end
    end

    % Proceso de preambleEndIdx
    if ~isempty(preambleEndIdx)
        prbStartIdx = preambleEndIdx(1) - obj.pPrbLenOffset;
        linearPrbStartIdx = obj.pBufferStartIdx + oldNBuffer +
prbStartIdx;

        if (linearPrbStartIdx > 0)
            circularPrbStartIdx = mod(linearPrbStartIdx - 1,
obj.pBufferLength) + 1;

            linIdx = obj.pPrbIdxBufferStartIdx +
obj.pNPrbIdxBuffer;

            cirIdx = mod(linIdx - 1, obj.pPrbIdxBufferLength) +
1;

            obj.pPrbIdxBuffer(cirIdx) = circularPrbStartIdx;
            obj.pNPrbIdxBuffer = obj.pNPrbIdxBuffer + 1;
            if obj.pNPrbIdxBuffer > obj.pPrbIdxBufferLength
                % Buffer overflow. Data at the beginning of the
buffer

                % overwritten with new data.
                obj.pNPrbIdxBuffer = obj.pPrbIdxBufferLength;
                obj.pPrbIdxBufferStartIdx = mod(cirIdx,
obj.pPrbIdxBufferLength) + 1;
            end
        end
    end
end

function [y, validFrm] = getBuffer(obj)

```

```

y = zeros(obj.OutputFrameLength, 1, 'like', obj.pBuffer);
validFrm = false;

if obj.pNPrbIdxBuffer > 0
    % Buffer tiene preámbulo

    nValid = obj.pBufferEndIdx -
obj.pPrbIdxBuffer(obj.pPrbIdxBufferStartIdx) + 1;
    if nValid < 1
        nValid = obj.pNBuffer -
(obj.pPrbIdxBuffer(obj.pPrbIdxBufferStartIdx) - obj.pBufferStartIdx - 1);
    end
    if nValid >= obj.OutputFrameLength
        % Suficientes datos válidos en el búfer para generar
nInvalid = obj.pNBuffer - nValid;
        linearIdx = obj.pPrbIdxBuffer(obj.pPrbIdxBufferStartIdx)
+ (0 : obj.OutputFrameLength-1);
        circularIdx = mod(linearIdx - 1, obj.pBufferLength) + 1;
y(1:obj.OutputFrameLength, 1) = obj.pBuffer(circularIdx);
        validFrm = true;
        % Update buffer states
obj.pPrbIdxBuffer(obj.pPrbIdxBufferStartIdx) = 0;
obj.pNPrbIdxBuffer = obj.pNPrbIdxBuffer - 1;
obj.pPrbIdxBufferStartIdx =
mod(obj.pPrbIdxBufferStartIdx, obj.pPrbIdxBufferLength) + 1;
obj.pBufferStartIdx = circularIdx(end);
obj.pNBuffer = obj.pNBuffer - obj.OutputFrameLength -
nInvalid;
    end
end
end

function [y, validFrm] = stepBitInput(obj, x, idx)

obj.setBuffer(x, idx);
[y, validFrm] = obj.getBuffer();

end

function [y, validFrm] = stepSymbolInput(obj, x, idx, dtMt)
if ~isempty(idx)
    [~, maxIdx] = max(dtMt(idx));
    bestIdx = idx(maxIdx);
else
    bestIdx = idx;
end
[y, validFrm] = stepBitInput(obj, x, bestIdx);
end
end
end

```

## PRUEBAS A REALIZAR

- Mostar que el bloque de la sincronización de trama funcione correctamente.
- Mostrar capturas de la constelación después de realizar la sincronización de trama.
- Realice el cálculo del BER del sistema completo y analice el resultado.
- Ejecute la simulación del sistema completo primero colocando las antenas del SDR ADALM-PLUTO y segundo realizando un loopback.

## **PREGUNTAS A CONTESTAR**

1. Realizar un código en Matlab que muestre el uso de la correlación cruzada para encontrar una secuencia con otra secuencia de datos más grandes.
  - a. Secuencia aleatoria de bits con código Barker.
  - b. Correlación cruzada entre el código Barker y la secuencia aleatoria que contiene el código.
2. Realizar un código en Matlab que compare las autocorrelaciones de secuencias de Barker de diferentes longitudes ( $N = 5$ ,  $N = 7$ ,  $N = 11$ ,  $N = 13$ )
3. ¿Qué sucede cuando se aumenta la longitud de la secuencia del código Barker?
4. Explique. ¿Cuándo ocurre la máxima correlación?

## **BIBLIOGRAFÍA**

- [1] R. G. D. P. A. W. Travis Collins, Software-Defined Radio for Engineers, Analog Devices, 2018.
- [2] C. Pérez, "Codificación de canal," [Online]. Available: <https://personales.unican.es/perezvr/pdf/codificacion%20de%20canal.pdf>. [Accessed 19 Octubre 2022].