

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

SIMULACIÓN DE OFDM MEDIANTE GNU RADIO

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN REDES Y TELECOMUNICACIONES**

PATRICIO SEBASTIÁN VALLEJO GUALACEO

DIRECTOR: FERNANDO VINIVIO BECERRA CAMACHO

DMQ, agosto 2022

CERTIFICACIONES

Yo, PATRICIO SEBASTIAN VALLEJO GUALACEO declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



PATRICIO SEBASTIAN VALLEJO GUALACEO

patricio.vallejo01@epn.edu.ec

sebitasvallejo97@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por PATRICIO SEBASTIAN VALLEJO GUALACEO, bajo mi supervisión.



FERNANDO VINICIO BECERRA CAMACHO

DIRECTOR

fernando.becerrac@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

A handwritten signature in black ink, appearing to be 'P. S. Vallejo Gualaceo', written in a cursive style.

PATRICIO SEBASTIÁN VALLEJO GUALACEO

DEDICATORIA

A mi hermosa familia que los amo un montón, a mis padres que se han esforzado mucho por ayudarme a progresar con mis hermanos y su dulce compañía en tiempos de angustia, en especial mi hermanito Esteban que deseo logre más cosas que yo. A mi hermosa novia Carlita que siempre estuvo allí cuando nadie más lo estaba, espero pronto podamos cumplir todos nuestros sueños y metas juntos y amarnos toda la vida. También a mi amigo William que su motivación ayudó a hacerme crecer como profesional y estudiar esta carrera, aún después de su muerte su memoria está presente.

AGRADECIMIENTO

Primero que nadie a Dios, que siempre me ha dado fuerzas para seguir adelante aún en mis peores infiernos. A mis profes de la carrera que no olvidaré sus consejos, ayuda y sobre todo la disciplina junto con su profunda dedicación de enseñarme. También a mi director de proyecto Ing. Fernando Becerra, que me enseñó que antes de ser profesional, uno debe ser amigo y su motivación de que no nos quedemos ahí, sino seguir y que estamos para cosas grandes.

ÍNDICE DE CONTENIDOS

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDOS	V
RESUMEN	VII
<i>ABSTRACT</i>	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general	1
1.2 Objetivos específicos	1
1.3 Alcance	2
1.4 Marco Teórico	2
2 METODOLOGÍA.....	5
3 RESULTADOS.....	6
3.1 Investigar acerca del proceso para realizar OFDM en GNU Radio	6
3.2 Construir un diagrama de bloques para simular el transmisor OFDM en GNU Radio.8	
3.3 Construir un diagrama de bloques para simular el receptor OFDM en GNU Radio.16	
3.4 Investigar acerca de las técnicas para contrarrestar el PAPR.....	22
3.5 Elección de una técnica para reducir el PAPR.	24
3.6 Implementar <i>Channel Coding</i> en GNU Radio.....	25
3.7 Realizar pruebas de funcionamiento de OFDM y <i>Channel Coding</i> en GNU Radio.26	
4 CONCLUSIONES.....	33
5 RECOMENDACIONES	34
6 REFERENCIAS BIBLIOGRÁFICAS	35

7 ANEXOS	i
ANEXO I: Certificado de Originalidad	i
ANEXO II: Video de evidencia y prueba de funcionamiento	ii

RESUMEN

El presente proyecto de titulación tiene como objetivo la simulación de la modulación OFDM mediante el programa informático GNU Radio y la implementación de una técnica para reducir el problema del PAPR en esa simulación. Una vez adquiridos los conocimientos necesarios sobre OFDM y los bloques necesarios para implementar la simulación, se desarrolla un proyecto en GNU Radio.

Este proyecto de titulación consta de cinco capítulos repartidos en el siguiente orden. El primer capítulo abarca el objetivo general, objetivos específicos, alcance y el marco teórico. El segundo capítulo describe la metodología aplicada en la implementación del proyecto.

El tercer capítulo explica el resultado de la investigación sobre el procedimiento que conlleva realizar la simulación de la modulación OFDM en GNU Radio junto con sus respectivos bloques dividiendo la simulación en dos partes: transmisor y receptor.

Además, se muestra la investigación de varias técnicas para reducir el PAPR y la razón por la que se escoge la técnica *Channel Coding* como la solución para la disminución del PAPR. También se describen los bloques necesarios y la configuración para la implementación de esta técnica en GNU Radio la misma que será adicionada al diagrama de bloques de GNU Radio del lado del transmisor con el objetivo de reducir el PAPR de la señal a transmitir. Por último se muestra mediante mediciones de potencia que se cumplió con el objetivo de reducir el PAPR.

Los capítulos restantes, cuarto y quinto se centran en las conclusiones adquiridas gracias al estudio de este proyecto de titulación y su ejecución, además de las recomendaciones alineadas con este tema de estudio.

PALABRAS CLAVE: GNU RADIO, OFDM, PAPR, MODULACIÓN, *CHANNEL CODING*, DIAGRAMA.

ABSTRACT

The objective of this degree project is to simulate OFDM modulation using the GNU Radio software and to implement a technique to reduce the PAPR problem in this simulation. Once the necessary knowledge about OFDM and the necessary blocks to implement the simulation are acquired, a project is developed in GNU Radio.

This degree project consists of five chapters distributed in the following order. The first chapter covers the general objective, specific objectives, scope and theoretical framework. The second chapter describes the methodology applied in the implementation of the project.

The third chapter explains the result of the research on the procedure involved in performing the simulation of OFDM modulation in GNU Radio along with their respective blocks dividing the simulation into two parts: transmitter and receiver.

In addition, the investigation of several techniques to reduce the PAPR and the reason why the Channel Coding technique is chosen as the solution to reduce the PAPR is shown. It also describes the necessary blocks and the configuration for the implementation of this technique in GNU Radio, which will be added to the GNU Radio block diagram on the transmitter side in order to reduce the PAPR of the signal to be transmitted. Finally, it is shown by means of power measurements that the objective of reducing the PAPR was achieved.

The remaining chapters, fourth and fifth are focused on the conclusions acquired from the study of this degree project and its execution, in addition to the recommendations aligned with this topic of study.

KEYWORDS: GNU RADIO, OFDM, PAPR, MODULATION, CHANNEL CODING, DIAGRAM.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Este proyecto de titulación está compuesto de varias secciones relevantes para su ejecución, constando en la primera parte una investigación sobre el procedimiento para la simulación de la modulación OFDM tanto del lado del transmisor como del receptor. Una vez investigado el proceso de la simulación OFDM en la siguiente sección se detallan los bloques necesarios para la simulación del transmisor OFDM y sus respectivas configuraciones y variables. Una vez que la sección del transmisor está completa se describen los bloques del receptor y la función de los bloques que lo conforman.

Al terminar las secciones del transmisor y del receptor se muestra una investigación de las técnicas disponibles para la reducción del problema del PAPR. En esta investigación se dividen las técnicas en 3 familias debido a la metodología que usan para reducir el PAPR y de detalla cómo opera cada técnica.

La siguiente sección se centra en la elección de una de las técnicas investigadas en la sección anterior. Esta sección parte desde la complejidad que maneja cada familia hasta la razón por la que se escoge la técnica *Channel Coding* para reducir el PAPR de este proyecto. Cuando ya se ha elegido la técnica de reducción de PAPR la siguiente sección se basa en los bloques necesarios y la implementación de la técnica en la simulación.

La última sección de este proyecto se centra en las pruebas realizadas para confirmar que se ha logrado modular correctamente una señal con OFDM. Además se realiza la medición de la potencia de la señal antes y después de la agregación de *Channel Coding* para verificar que se redujo el PAPR en la simulación.

1.1 Objetivo general

Simular la modulación OFDM en GNU Radio.

1.2 Objetivos específicos

- Investigar el proceso para modular OFDM en GNU Radio.
- Construir un diagrama de bloques para simular un transmisor OFDM en GNU Radio.
- Construir un diagrama de bloques para simular un receptor OFDM en GNU Radio.

- Investigar técnicas para reducir el PAPR.
- Elegir una técnica para la reducción de PAPR.
- Implementar *Channel Coding* en GNU Radio.
- Realizar pruebas de funcionamiento de OFDM y *Channel Coding* en GNU Radio.

1.3 Alcance

El presente trabajo de titulación implementará una simulación de la modulación OFDM mediante el programa informático GNU Radio. Para lograrlo se estudiarán los bloques relevantes de GNU Radio que requiere dicha modulación junto con su configuración en ejemplos similares del repositorio de GNU Radio y en trabajos de investigación de la misma línea de estudio.

Además se investigarán técnicas para reducir el problema del PAPR para luego seleccionar de entre ellas una que se pueda aplicar en GNU Radio. Una vez elegida la técnica se la implementa. Para verificar la reducción del PAPR se realizará mediciones de los picos de potencia antes y después de la adición de dicha técnica. También se aplicará una fórmula matemática que permitirá apreciar la disminución del PAPR en la señal.

1.4 Marco Teórico

OFDM

La Multiplexación por División de Frecuencias Ortogonales, es una modulación multiportadora que divide el ancho de banda disponible en subportadoras, estas subportadoras son espaciadas ortogonalmente. La ortogonalidad de las subportadoras se da cuando se mantiene la mínima separación en frecuencia que permita a una subportadora estar en su máxima amplitud mientras la señal adyacente tiene una amplitud nula de tal manera que no interfieran unas subportadoras con otras como se observa en la Figura 1.1 [1].

OFDM transmite las subportadoras combinándolas ortogonalmente para formar una señal que se caracteriza por ser de banda estrecha. Cada subportadora puede ser modulada por BPSK o QPSK que son técnicas de modulación por desplazamiento de fase de los símbolos. OFDM busca un uso eficiente del ancho de banda disponible, además por su rápido procesamiento de la señal OFDM se considera un sistema de alta velocidad. Por lo cual, se utiliza en comunicaciones cada vez más comunes de forma inalámbrica como son: 4G LTE, Wi-fi y 5G [2] [3].

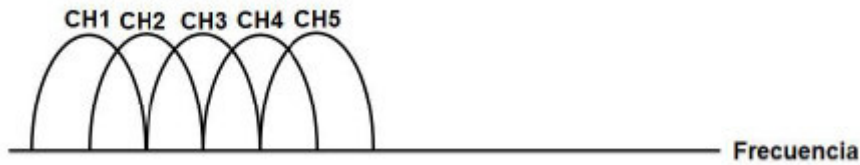


Figura 1.1 Subportadoras de OFDM.

El desarrollo de OFDM tal como se conoce hoy tardó varios años ya que OFDM en sus comienzos tenía una alta complejidad de implementación, Chang por la década de los años 60 publicó varios principios para el envío simultáneo de información, su sistema se basaba en el uso de complejos circuitos y debido al uso de varios generadores de subportadoras del tipo sinusoidales requería el mismo número de demoduladores para procesar la señal, sin mencionar que todo esto se procesaba en el dominio del tiempo, de manera que no era un sistema ni viable ni eficiente [4].

En 1971 Weinstein y Ebert reprogramaron OFDM para mejorar su eficiencia y manejar las señales en el dominio de la frecuencia, esta reprogramación eficiente llegó con la introducción de la transformada discreta de *Fourier* que también permitía modular y demodular en banda base, para afrontar las interferencias: Inter símbolos (ISI) y a las interferencias Inter *Carrier* o inter portadora (ICI) utilizaron entre los símbolos un espacio que servía de guarda el cual era vacío, aunque sin lograr la ortogonalidad ideal [5].

Para 1980 se introdujo a OFDM el uso del Prefijo Cíclico. Esta técnica reemplazaba el espacio vacío agregado 9 años antes por un pedazo del símbolo el cual copia la parte final del símbolo y lo agrega al comienzo para generar esa ortogonalidad tan anhelada [5].

PAPR

Peak to Average Power Ratio, se define como la manifestación de grandes picos en la potencia inmediata comparada con la potencia media de la señal en los sistemas que trabajan con la modulación OFDM. PAPR es un problema que debe considerarse si se requiere realizar OFDM ya que si se pasa por un amplificador de potencia, inmediatamente se manifestará una saturación lo que da a lugar a una distorsión en la señal [6].

PAPR se manifiesta en los sistemas que son multiportadora debido a que las subportadoras están desfasadas entre ellas. Estas señales presentan diferentes valores en sus amplitudes, pero en un instante determinado todas alcanzan su máxima amplitud

de manera simultánea haciendo que a la salida después de procesar la señal, se presente un pico pronunciado como se observa en la **Figura 1.2** [6].

Otras causas por las que los sistemas de comunicaciones padecen del problema del PAPR son, por ejemplo: el ruido introducido por el mismo canal de transmisión de la señal, también el ruido de cuantificación como una consecuencia directa proveniente del proceso de codificación. Este ruido suele insertarse en el transmisor y es distribuida por todo el sistema hasta la salida del receptor, el ruido que trae consigo el canal genera errores de transmisión causando que la señal no sea la misma que la señal transmitida [7].

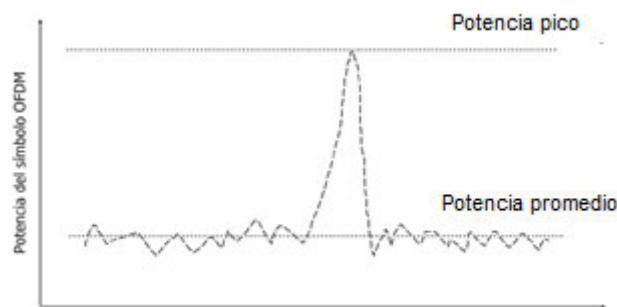


Figura 1.2 Ejemplo de PAPR en OFDM.

El PAPR de una señal se puede definir matemáticamente como se puede apreciar en la Ecuación 1.1.

$$PAPR (db) = \frac{\max_{0 \leq t \leq T} \{|x(t)|^2\}}{E \{|x(t)|^2\}}$$

Ecuación 1.1 Ecuación del PAPR [8].

Donde:

$T \{|x(t)|\}$: Es el pico máximo de la potencia de la señal en un determinado instante de tiempo, dentro del intervalo que dura la señal.

$E\{|x(t)|\}$: Es la potencia promedio de la señal.

t : Es el instante de tiempo cuando la señal es medida.

GNU Radio

Es un programa informático gratuito y de código abierto enfocado en el procesamiento de señales para sistemas de comunicaciones a través de bloques que simulan los procesos de los aparatos de hardware que tratan las señales en telecomunicaciones.

Las simulaciones se pueden acompañar con equipos especializados en la implementación de proyectos se llama USRP. GNU Radio se centra en el procesamiento de las señales en banda base mientras los USRP trabajan a nivel de canal o radio frecuencia [9].

GNU Radio facilita la implementación de esquemas de comunicaciones partiendo de diagramas de bloques que cumplen funciones específicas en el procesamiento de las señales y sin costo alguno. Esto abrió la oportunidad para que cada vez más usuarios, desde aficionados hasta investigadores de comunicaciones inalámbricas, puedan usar esta tecnología en sus proyectos y desplegar sus propios algoritmos para procesar señales [9].

2 METODOLOGÍA

Para empezar se indagó sobre cuáles son los bloques que se necesitan para tratar una señal con la técnica de modulación OFDM en GNU Radio con sus respectivas configuraciones y variables, para ello se tomó como guía el trabajo que realizaron Vishrant Tripathi junto a Bansal Kshitiz [10]. Una vez analizados los bloques necesarios y su configuración para la simulación de OFDM se precedió a su despliegue en GNU Radio.

Se procedió con el despliegue del diagrama de bloques del transmisor, su conexión y la creación de 12 variables para el procesamiento de la señal. Al terminar el diagrama del transmisor se continuó con la implementación de los bloques del receptor y sus 3 nuevas variables debido a que se reutilizaron las del transmisor para evitar errores de configuración en el momento del despliegue de los bloques, pues se requieren los mismos parámetros del transmisor para procesar la señal.

Luego se investigó sobre las técnicas para la reducción de PAPR. Al haber recopilado información de las técnicas se procedió con la elección de *Channel Coding* como técnica a implementar en la simulación. Después de lo cual se procedió con el estudio de los bloques necesarios y su respectiva configuración para la implementación de *Channel Coding* en GNU Radio. Para implementar esta técnica se tomó en consideración el modelo de Goswami, Kakkar y Rani [8].

Una vez implementados los bloques se precedió a colocar visualizadores que permitieron comprobar: la forma de la señal OFDM y medir la potencia de los picos para poder utilizarlos en una ecuación que permitió determinar la medida del PAPR y su reducción después de agregar el *Channel Coding*.

3 RESULTADOS

Como primera parte se presenta la investigación del proceso que se debe realizar en GNU Radio para simular OFDM, este proceso se divide en dos partes: transmisor y receptor. La segunda parte muestra los bloques utilizados para la construcción de un transmisor OFDM y su configuración.

En la tercera parte se aprecian los bloques del receptor OFDM y los parámetros añadidos a ellos. Para la cuarta parte se presenta la investigación de las técnicas utilizadas para la reducción del PAPR en OFDM.

La quinta parte explica la razón por la que se escoge la técnica *Channel Coding* para la reducción de PAPR en este proyecto. La sexta parte muestra la investigación de *Channel Coding* y los bloques necesarios para su implementación en GNU Radio como técnica para reducir el PAPR. La séptima parte detalla las pruebas de funcionamiento de los bloques y el cálculo del PAPR como evidencia de su reducción.

3.1 Investigar el proceso para modular OFDM en GNU Radio

El proceso para realizar OFDM se puede dividir para una mejor comprensión en dos etapas: transmisor y receptor.

Transmisor

El diagrama de bloques en GNU Radio para simular OFDM parte de la señal generada por una fuente que emite la señal para ser procesada. A esta señal inicial se le debe agregar etiquetas por medio de un bloque etiquetador *Stream To Tagged Stream*, estas etiquetas permiten segmentar el flujo de la señal en partes iguales o simétricas para luego separarlo según las etiquetas [4]. Una vez etiquetado el flujo de información se agrega a la señal un código CRC mediante el bloque *Stream CRC32*. Este bloque se agregó para la detección de errores y de esta manera se pueda identificar en el receptor posibles fallas surgidas en la transmisión [11].

Al llegar a este punto la señal se divide para obtener una cabecera mediante al bloque *Packet Header Generator* y de esa manera identificar los paquetes generados. Además se empaqueta la señal por el bloque *Repack Bits*. Las etiquetas antes añadidas permitirán la segmentación en este punto y permitirán el mapeo del flujo de información [4] [11]. A la cabecera generada se la modula por el método BPSK al pasar por el bloque *Chunks to Symbols*. La información empaquetada es modulada por el bloque *Chunks to Symbols* mediante QPSK para luego unir la cabecera de la información junto a la carga

útil mediante un multiplexor *Tagged Stream Mux* el cual permite concatenar los flujos y de esa manera formar los paquetes para ser procesados y convertidos en símbolos OFDM [12].

El siguiente paso del procedimiento para la simulación de OFDM en GNU Radio consiste en designar las funciones que cumplirá cada subportadora todo esto por medio del bloque *OFDM Carrier Allocator*. Dentro de las atribuciones que asigna este bloque a las subportadoras están las funciones de palabras de sincronismo y los símbolos piloto que permiten se realice la simulación al ser las que den forma a la señal OFDM resultante en GNU Radio [12].

El bloque *OFDM Carrier Allocator* realiza la conversión de serie a paralelo internamente para enviar la información en paralelo al bloque FFT configurado de tal manera que sea el bloque de la transformada inversa de *Fourier* para la conversión del dominio de la frecuencia al del tiempo. Luego la señal pasa por el bloque *OFDM Cyclic Prefixer* este bloque inserta una banda de guarda de la parte final del símbolo OFDM al inicio de este. Terminando de esa manera el procedimiento en el transmisor [13].

Receptor

Es vital en esta parte tener buena sincronización ya que de ello depende la capacidad de reconocer el símbolo en la transmisión. Para ello a la entrada del receptor se usa un bloque *Schmidl & Cox OFDM sync* como sincronizador este bloque funciona con otros más que ayudan a filtrar y modelar el canal como *Channel Model* [11]. Luego los símbolos son demultiplexados mediante el bloque *Header/Payload Demux* permitiendo separar la carga útil y la de cabecera para su posterior procesamiento. Ambos flujos deben ser procesados por el bloque FFT para que sean transformarlos al dominio de la frecuencia y de esa manera convertirlos de serie a paralelo. Una vez que los flujos estén en paralelo se procede a retirar las características añadidas de OFDM como el prefijo cíclico [11].

Se censa la señal con la información dentro de la cabecera para que con el bloque *OFDM Channel Estimation* poder ecualizar la señal mediante el bloque *OFDM Frame Equalizer*. Una vez llegado a este punto es importante que el flujo de la cabecera y la carga útil vuelvan a estar en serie y se les retire las subportadoras antes añadidas como: subportadoras piloto y palabras de sincronismo, esta función la realiza el bloque *OFDM Serializer* tanto a la carga útil como a la cabecera [4].

En este punto se debe agregar bloques de demodulación por lo que se añade el bloque *Contellation Decoder* y de esta manera se realice la demodulación BPSK en la cabecera.

Además en ese bloque se realiza una conversión del dominio de la frecuencia al dominio del tiempo. Para la demodulación se asocia la tabla de modulación con la variable utilizada en el transmisor [12].

Con el flujo de la cabecera ya demodulado se realiza una retroalimentación con el bloque *Header/Payload Demux* del inicio. El flujo de la carga útil es demodulado por el bloque *Contellation Decoder* [12]. Después de la demodulación se debe pasar el flujo de la carga útil por el módulo de desempaqueado del flujo el cual es el bloque *Repack Bits*. Finalmente se retira el código de detección de errores del inicio con el bloque *Stream CRC32* [4] [12].

3.2 Construir un diagrama de bloques para simular un transmisor OFDM en GNU Radio.

Al iniciar la elaboración del transmisor de la modulación OFDM contemplado en la Figura 3.1 cobra relevancia la selección de la clase de señal que se desea transmitir ya sean archivos de texto, audios y señales aleatorias. Esto es posible pues GNU Radio permite usar como fuente de señal varios tipos de bloques que incorporan la posibilidad de integrar las clases de archivos mencionados.

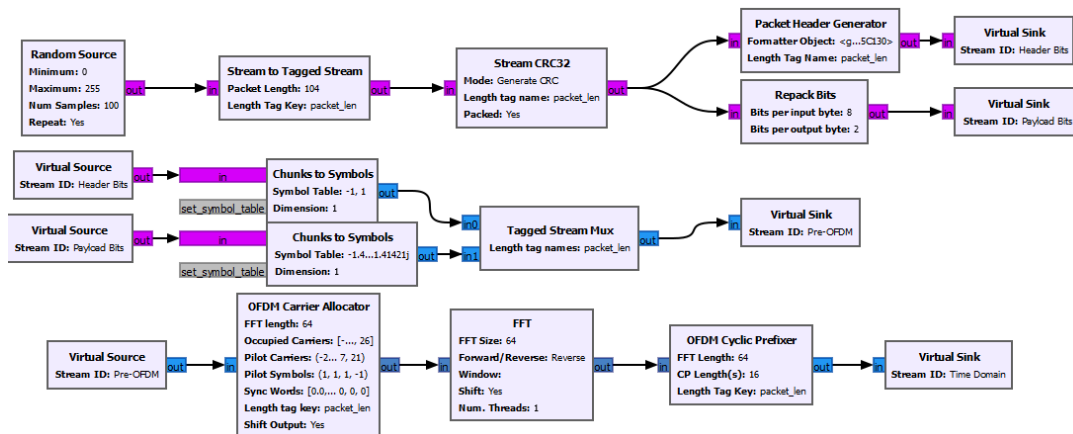


Figura 3.1 Transmisor OFDM.

En este proyecto se desea transmitir una señal de tipo aleatoria. Para aquello se elige el bloque de fuente aleatoria como se observa en la Figura 3.2. Dicha fuente solo necesita la configuración el máximo y mínimo de la señal los cuales son muestras de los números aleatorios generados por la fuente [12]. El número de muestras que servirá para ser apreciable a lo largo de la transmisión viene dado por la variable “samp_rate” cuyo valor se fijó en 100 para una mejor visualización en los análisis posteriores. La opción *Repeat* en modo *yes* se utiliza para enviar un flujo constante de la señal, si se

coloca *no* solo enviará una vez la señal y será imposible estudiar la señal ya que solo se mostrará una sola vez en los bloques visualizadores [12].

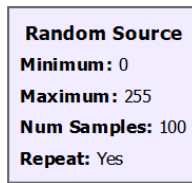


Figura 3.2 Bloque Fuente Aleatoria.

Etiquetado

El bloque *Stream to Tagged Stream* representado por la Figura 3.3 permite etiquetar el flujo de bits dividiéndolo simétricamente, este etiquetado se da luego de “n” bits. Gracias a este proceso los siguientes bloques organizaran los paquetes con el tamaño correspondiente para su procesamiento. Para este proyecto de titulación se etiquetará el flujo en 104 bits valor que se definió en la variable “packet_len” para que la carga útil llamada como *payload* se de cada 2 bits por símbolo, como resultado de la etiquetación se tiene 52 bits gracias a la modulación BPSK como se sugiere en el trabajo de Vishrant Tripathi junto a Bansal Kshitiz [10].

La longitud de la clave también se define por el mismo valor de la variable “packet_len”, pero es establecida dentro de otra variable con el nombre de: “packet_length_tag_key”. Ambas variables serán utilizadas en la configuración de muchos otros bloques dentro de este proyecto.



Figura 3.3 Bloque etiquetador de bits.

Agregación de código para detección de errores

Una vez etiquetado el flujo se adiciona al diagrama el bloque CRC32 que tiene como entrada un flujo de bytes. Dicho flujo se encarga de formar un paquete con el flujo que viene del bloque *Stream to Tagged Stream*.

Por consiguiente, el bloque *Stream CRC32*, tiene como función insertar un código del mismo nombre con el propósito de detectar errores que se pueden presentar en el proceso anterior de etiquetación. Esto ayuda a detectar dichos errores en el proceso de recepción con el bloque encargado de identificar la integridad de la señal.

El bloque codificador puede apreciarse en la Figura 3.4. Para la configuración de este bloque se debe seleccionar el modo en el que operará, si se selecciona “*Generate CRC*” le indica al bloque que debe codificar sino cumplirá la función de detectar errores. En la longitud del bloque *Stream CRC32* se coloca la longitud que se utilizará, este parámetro está definido con la variable: “packet_len” con el valor de 104 para evitar errores en toda la simulación.

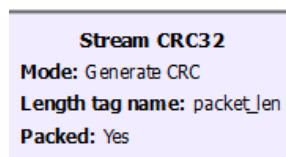


Figura 3.4 Bloque CRC32.

A la salida del bloque se obtiene los mismos bytes de la entrada, pero a la información se le añadió un código que tiene un valor de 32 bits que viene por defecto por ser CRC32. Luego la señal se bifurca para ser tratada en paralelo por los bloques *Packet Header Generator* y *Repack Bits*.

Generador de cabecera

El bloque *Packet Header Generator* mostrado en la Figura 3.5 se encarga de generar la cabecera para la señal. Dentro de este bloque se hallan dos variables de relevante valor que son: longitud y formato. El valor de la longitud está asociado a la variable: “packet_len”, la variable creada para el formato es “header_formatter” la cual es configurada en sus parámetros para ser utilizada en OFDM. Sin embargo el campo formato es el que se encarga de generar la cabecera para los símbolos OFDM mediante la variable “header_formatter”. A la salida de este bloque continúa el proceso de los datos, sin embargo, las cabeceras ya son moduladas.

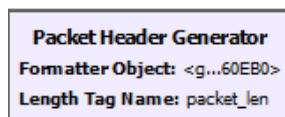


Figura 3.5 Bloque generador de cabecera.

Empaquetado de bits

Por otra parte, el bloque *Repack Bits* observado en la Figura 3.6 es el encargado de empaquetar los flujos para su posterior modulación. En este bloque es importante delimitar la cantidad de bits de entrada y salida para que se asigne los bits por cada símbolo.

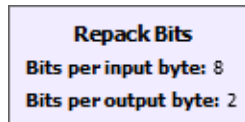


Figura 3.6 Bloque empaquetador de flujos

. Para este proyecto se configura este bloque con los parámetros:

- *Bits per input byte: 8.*
- *Bits per output bytes: 2.*

Cabe destacar que el valor 2 en el segundo parámetro del bloque *Repack Bits* no es arbitrario, sino es el resultado de una función como se observa en la Figura 3.7. Dicha función se establece de la siguiente manera: “payload_mod.bits_per_symbol()” [4].

La función se divide en dos partes:

- payload_mod: Es una variable que fue configurada con el tipo de modulación que tendrá la carga útil que es QPSK.
- bits_per_symbol: Es el operador matemático de la función y extrae el número de símbolos que maneja la modulación configurada en el primer parámetro.

Al utilizar una modulación QPSK de 4 fases la función extrae 1 símbolo por cada 2 bits. En otras palabras lo que entiende el bloque es que por cada 8 bits se alojen nada más 2 bits lo que nos arroja un total de 4 símbolos. La longitud de la clave se almacena en la variable “length_tag_key” que tiene el mismo valor que la función “packet_len” para evitar errores en la simulación [11].

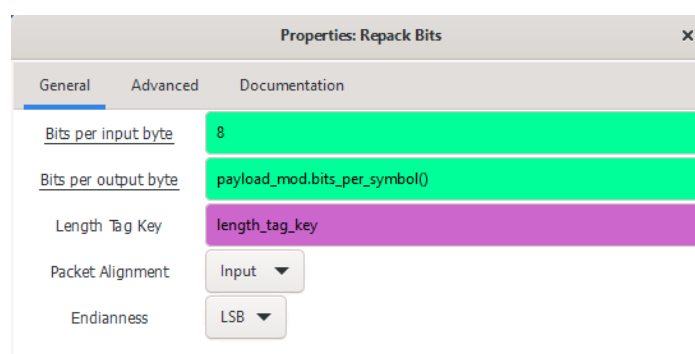


Figura 3.7 Bloque *Repack Bits* y su configuración

Los bloques *Virtual Sink* y *Virtual Source* presentes en la Figura 3.8 son sincronizadores que permiten al usuario presentar un orden más estético solamente colocando al final de un proceso al bloque *Virtual Sink* y asignándole un nombre o identificación. Para proseguir con el proceso requerido el usuario únicamente coloca un bloque *Virtual*

Source con la identificación asignada al bloque *Virtual Sink* anterior y de esa manera no se necesita de cables para conectarlos [12].



Figura 3.8 Bloques Sincronizadores

Modulación

El siguiente bloque en el proceso de modulación de OFDM es el mapeo de los bits que están fluyendo para ello se recopilan muestras que serán consignadas a los distintos valores aceptados basados en la tabla de modulación los cuales son en función del tipo de modulación: BPSK y QPSK. Para el mapeo en GNU Radio se utiliza el bloque *Chunks to Symbol* mostrado en la Figura 3.9, este bloque inserta valores del tipo racionales o “Q” e imaginarios denominados como “I” en una tabla para designar dichos valores a los bits entrantes con la condición de cuál es la modulación que se requiere [12].

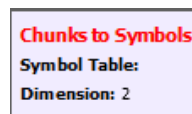


Figura 3.9 Bloque mapeador

Para configurar este bloque se empieza seleccionando el tipo de formato que tiene el flujo el cual está en modo *byte* y para la salida se debe seleccionar el formato *complex* lo que hará que se agregue la parte imaginaria y se desplacen los símbolos según la modulación establecida para la cabecera y la carga útil. Para mapear los símbolos de la cabecera se configura el bloque *Chunks to Symbols* como se observa en la **Figura 3.10**, el parámetro *Symbol Table* es la tabla como se ordenarán los símbolos [12]. Para la cabecera este parámetro es establecido por la función: “header_mod.points()” que extrae el número de los símbolos y su distribución de acuerdo con la variable “header_mod” [4] [11]. Esta variable “header_mod” fue configurada para que se ejecute la modulación BPSK arrojando dos símbolos separados 180°.

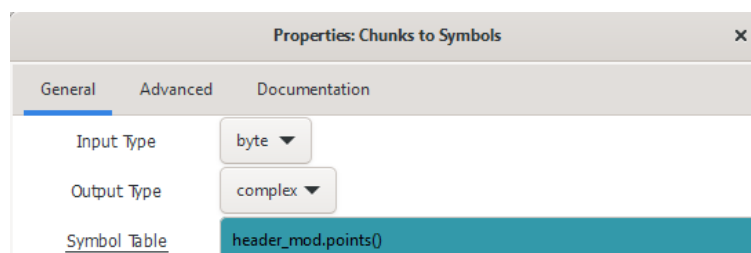


Figura 3.10 Configuración del mapeador para la cabecera

Para mapear los símbolos de la carga útil se configura el bloque *Chunks to Symbols* como se observa en la **Figura 3.11**. El parámetro *Symbol Table* es la tabla como se ordenarán los símbolos. Para la carga útil este parámetro es establecido por la función: “payload_mod.points()” que extrae el número de los símbolos y su distribución de acuerdo con la variable “payload_mod”, esta variable “payload_mod” fue configurada para que se ejecute la modulación QPSK arrojando cuatro símbolos separados 90° [4] [11].

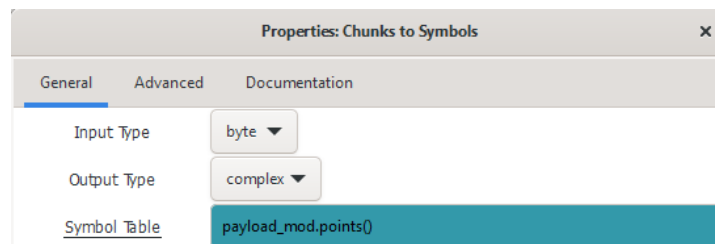


Figura 3.11 Configuración del mapeador de la carga útil

La modulación de la cabecera es BPSK, esta se centra únicamente en el desplazamiento de fase para un par de símbolos que están separados por un ángulo de 180°. Esto se puede apreciar en el diagrama de constelación de la Figura 3.12 que muestra a la cabecera mediante la modulación BPSK [12].

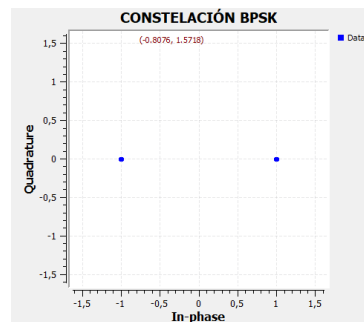


Figura 3.12 Diagrama de constelación de BPSK

Por otra parte, QPSK maneja cuatro diferentes puntos separados entre sí por 90° como se observa en la Figura 3.13, los cuales tienen el valor de: $\{(-1.414, 1.414); (1.414, 1.414); (1.414, -1.414); (-1.414, -1.414)\}$. Esto debido a que la modulación QPSK no altera ni la amplitud ni la frecuencia, únicamente la fase. Los símbolos resultantes se mantienen distanciados de manera idéntica.

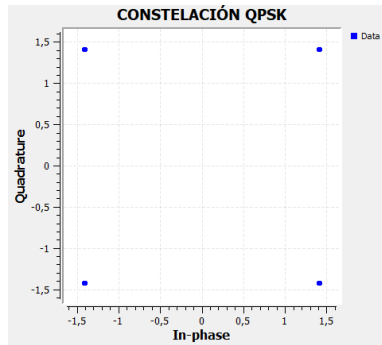


Figura 3.13 Diagrama de constelación de QPSK

Una vez culminada la modulación es muy importante juntar la cabecera con la carga útil para que de esta manera se logren formar los paquetes para ser procesados. Ahí es donde el bloque *Tagged Stream Mux* entra en escena y cumple las funciones de un multiplexor, se puede visualizar el bloque en la Figura 3.14 y tiene como variable “packet_len” configurada anteriormente para evitar errores en la simulación.

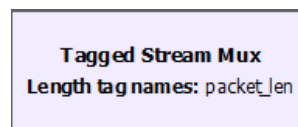


Figura 3.14 Bloque Multiplexor.

Asignación de las subportadoras

Para esta sección es importante el bloque *OFDM Carrier Allocator* mostrado en la Figura 3.15, este bloque convierte el flujo de a vectores para ser ingresados al bloque IFFT. Además agrega los símbolos piloto en las portadoras, portadoras piloto, y palabras de sincronización [12].

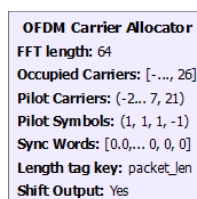


Figura 3.15 Bloque Asignador de portadora.

La configuración de este bloque depende de 6 variables que llevan el mismo nombre dentro de la configuración del bloque *OFDM Carrier Allocator* como se puede apreciar en la Figura 3.16 [4]. Para mayor comodidad al llamar las variables a las funciones que las requieren:

- `fft_len`: Define la cantidad total de las subportadoras para este caso de estudio 64.
- `occupied_carriers`: Asigna el lugar correspondiente a las subportadoras que serán usadas para los datos, estos estarán entre los rangos de las posiciones: primera (-26,-22); segunda (-20,-8); tercera (-6,-1); cuarta (1,-6); quinta (8,20); sexta (22,26).
- `pilot_carriers`: Establece el lugar de las subportadoras piloto, éstas son: -21,-7,7,21.
- `pilot_symbols`: Permite a los símbolos tener definido su símbolo piloto el cual se define con: (1, 1, 1, -1).
- `sync_word`: Permite agregar sincronismo agregando bits nulos, el conjunto debe tener la longitud de la variable `fft_len`: 64, este parámetro también se usa en el sincronismo.

Una de las principales ventajas del bloque *OFDM Carrier Allocator* es que a la salida del bloque la señal ya ha sido convertida de serie a paralelo para el siguiente bloque.

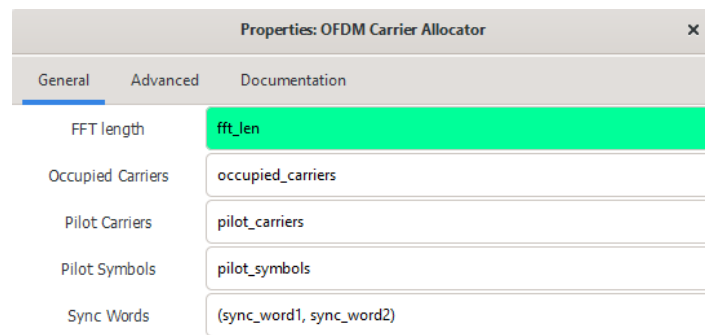


Figura 3.16 Configuración del bloque *OFDM Carrier Allocator*.

Transformada Inversa de *Fourier*

Una vez que la señal está en paralelo ingresa al bloque FFT que se observa en la Figura 3.17 y convertir los símbolos al dominio de la frecuencia. La configuración de este bloque solo abarca el modo de la función, que para este proyecto se lo realizó en “*Reverse*” ya que solo así trabaja en el modo IFFT, a este bloque se debe añadir la longitud y la cantidad de subportadoras que maneja la simulación mediante la inserción de la variable: “`packet_len`” y evitar errores en la simulación [12].

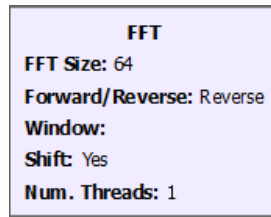


Figura 3.17 Bloque FFT

Para agregar el prefijo cíclico la señal debe pasar por el bloque OFDM *Cyclic Prefixer* donde asigna la longitud del prefijo cíclico el cual es necesario para que sirva de guarda para los símbolos. El bloque se puede apreciar en la Figura 3.18 [12].

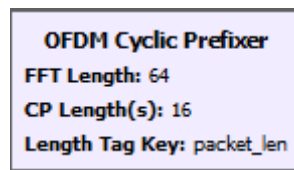


Figura 3.18 Bloque OFDM *Cyclic Prefixer*

A la salida, la señal está lista para transmitirse. La misma se halla en el dominio del tiempo.

3.3 Construir un diagrama de bloques para simular un receptor OFDM en GNU Radio.

El receptor OFDM se ve en la Figura 3.19, en este bloque es de vital importancia el sincronismo, pues mediante este proceso de sincronización se analiza la información y de esa manera se delimita las tramas. Otra ventaja de este proceso es el rastreo de un desfase de la señal ejecutando una corrección.

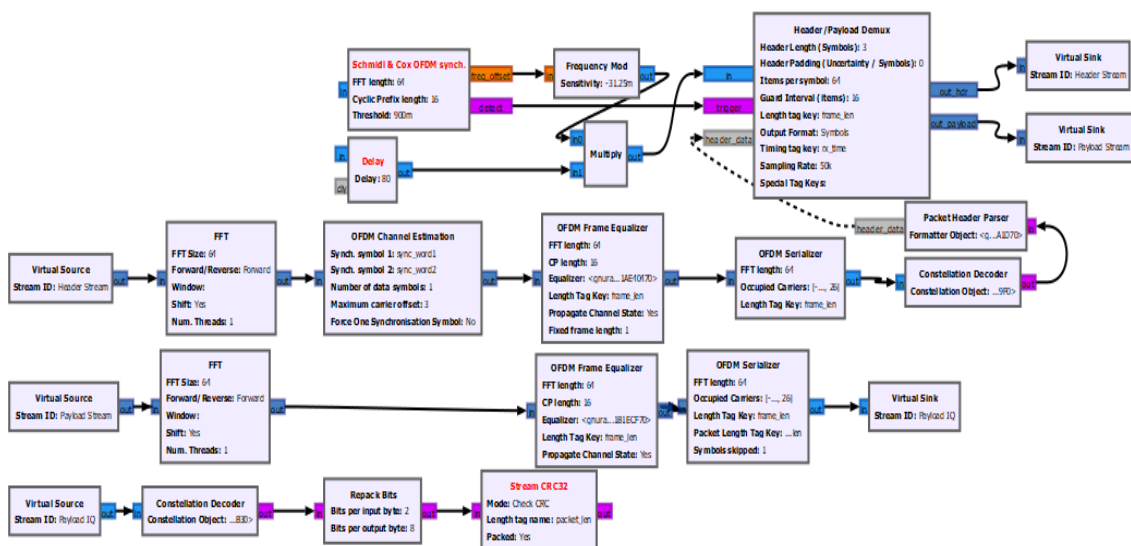


Figura 3.19 Receptor OFDM

En GNU Radio se halla el bloque *Schmidl & Cox OFDM Sync* como se observa en la Figura 3.20 que ejecuta el algoritmo homónimo. Este bloque tiene 2 salidas, la primera determina si está presente un desplazamiento de frecuencia y la segunda da información de detección de las palabras usadas en el momento de la sincronización, además fija los símbolos, carga útil y cabecera [11].

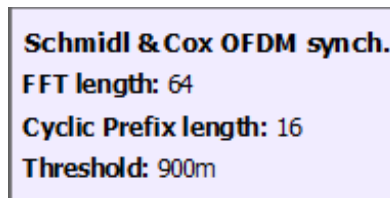


Figura 3.20 Bloque sincronizador.

A continuación del bloque de sincronización consta el bloque demultiplexor tal como se observa en la Figura 3.21 tiene 3 entradas: la información de la carga útil, los datos de control del bloque de sincronismo y la información de control de la cabecera después de ser desarrollada en la fase correspondiente. Este bloque tiene como función principal demultiplexar las cabeceras de la carga útil y a la vez entregar dos flujos de información y entregar dos flujos de información en serie y el tiempo [11] [12].

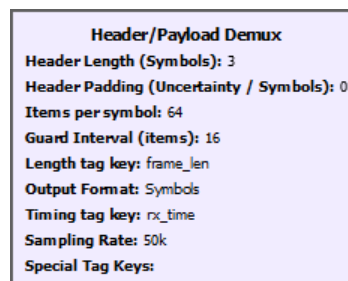


Figura 3.21 Bloque demultiplexor.

Procesamiento de la cabecera

Una vez que la información llega desde el bloque demultiplexor hasta los bloques OFDM debe ser procesada. Es fundamental mencionar que el primer bloque de procesamiento de la cabecera es el bloque FFT que convierte la señal del dominio del tiempo al dominio de la frecuencia tal bloque se observa en la Figura 3.22, para ello es necesario previamente seleccionar el modo *forward* para el respectivo procesamiento de información en el bloque [12].

Además, se debe usar la variable: "fft_len" con el mismo valor que se usó en el bloque del transmisor. Posteriormente la información de la cabecera sale de forma paralela del bloque FFT estos valores ingresan al bloque de estimación de canal.

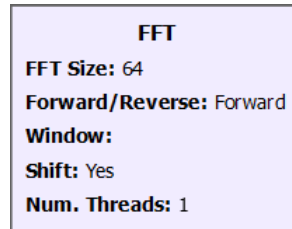


Figura 3.22 Bloque FFT para procesamiento de la cabecera.

Estimación de canal

El bloque de estimación de canal tiene la función de evaluar el comportamiento de la señal sobre el canal y asimismo cómo este afecta a la señal que está en curso. Además en el prototipo estos símbolos de entrenamiento son las palabras de sincronismo, las cuales se encargan del proceso de estimación de canal para el sistema OFDM denominado *OFDM Channel Estimation* que se puede observar en la Figura 3.23 [12].

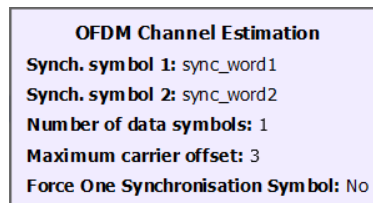


Figura 3.23 Bloque para la estimación de canal OFDM.

Por consiguiente la información que se obtiene a partir del proceso de estimación de canal ingresa al bloque *OFDM Frame Equalizer*. El bloque ecualizador dependerá de la longitud de la "fft_len" y del prefijo cíclico tal como se observa en la Figura 3.24. Este bloque ecualizador está encargado de tomar la información de la estimación del canal y ayudar a contrarrestar las anomalías del canal y enviar la información de control para poder. Además tiene la función de enviar información de control para procesar de una manera eficaz la información de la carga útil [11].

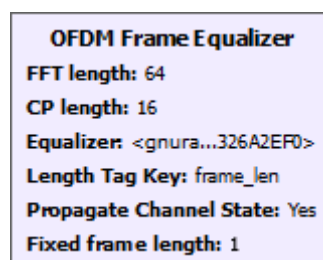


Figura 3.24 Bloque *Frame Equalizer*

Conversión Paralelo a Serie

El bloque *OFDM Serializer* como se observa en la Figura 3.25 tiene la función de convertir los flujos en paralelo en un solo flujo en serie. De esta manera también obtener la información de las cabeceras que se adjuntó en las subportadoras de datos que llegan como flujos paralelos y las convierte en un flujo en serie [4].

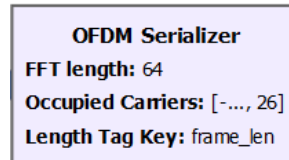


Figura 3.25 Bloque serializador.

Demodulación de la cabecera

En esta fase se utiliza el bloque *Constellation Decoder* como se observa en la Figura 3.26 definida en GNU Radio para el proceso de demodulación QPSK, la cual se encarga de realizar el proceso inverso al bloque “*chunks to symbols*” mediante la función *Constellation Object* que se utiliza la variable “*payload_mod.base()*” la cual permite desmapear los bits de la modulación BPSK . Además, con el proceso del desmapeo, se consigue que la información cambia del dominio de la frecuencia al dominio del tiempo [11].

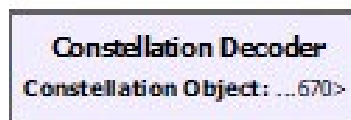


Figura 3.26 Bloque decodificador de la constelación.

FFT y Conversión Serial a Paralelo

Después del proceso del envío de: símbolos de entrenamiento, palabras de sincronismo y el proceso de estimación de canal comienza el procesamiento de la carga útil la cual consiste en la conversión serie a paralelo y uso de la FFT en modo “*forward*” para proceder de esa manera con la Transformada de *Fourier*, este bloque del receptor depende de la variable “*fft_len*”, tal como se observa en la Figura 3.27. Por consiguiente, para el paso de información la salida se conecta al bloque ecualizador [11].

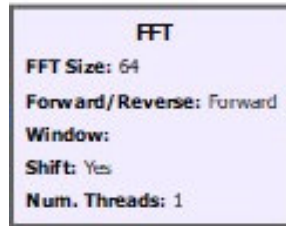


Figura 3.27 Bloque FFT.

Ecuador

Al finalizar el proceso del bloque de la FFT se conecta un bloque ecualizador con la finalidad de mitigar las anomalías de canal. En este bloque se utiliza la información sobre la estimación de canal que se logró obtener en el proceso de la cabecera realimentado al demultiplexador. Es fundamental importancia identificar que en este proceso el ecualizador es diferente tal como se observa en la Figura 3.28 porque depende directamente de una variable tipo objeto que llama al ecualizador propio de carga útil de OFDM [11].

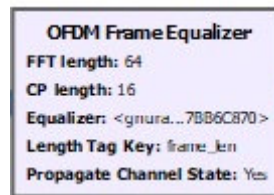


Figura 3.28 Ecuador para la carga útil.

Convertor Paralelo a Serial

El bloque *OFDM Serializer* tiene la función convertir los flujos de símbolos a un solo flujo en paralelo para ello se encarga de tomar la información de cada subportadora de datos tal como se observa en la Figura 3.29. De esta manera logra entregar este flujo de símbolos en serie listo para ser demodulado [12].

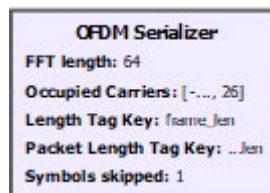


Figura 3.29 Convertor Paralelo/Serie de la carga útil.

Demodulación

El flujo de los símbolos de los bloques OFDM están preparados para el proceso de desmapeo para lo cual utilizan el bloque *Constellation Decoder* para asociar la tabla de desmapeo con los símbolos que ingresan tal como se observa en la Figura 3.30. Adicional se conectan con la variable “*constellation_object*” [12].

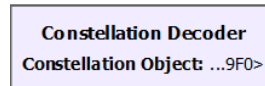


Figura 3.30 Bloque demodulador de la carga útil.

Luego del proceso de demodulación de los símbolos se procede a entregar un flujo de bits en serie empaquetados y etiquetados en el dominio del tiempo.

Desempaquetado

En este proceso se utiliza el bloque *Repack bits* con la respectiva descripción de entrada y salida. La entrada del bloque de desempaquetado se utiliza la misma variable de transmisor, es decir que en este proceso se agrupó a los bits para formar 4 subgrupos de 2 bits. Esta salida de los bits representa un símbolo desmapeado y posteriormente se eliminan los subgrupos y grupos bits para generar un solo flujo desempaquetado tal como se observa en la Figura 3.31 [4].

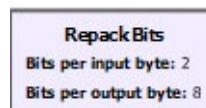


Figura 3.31 Bloque Desempaquetador de carga útil.

Retiro de CRC32

Cómo último componente del receptor es el bloque CRC32 que se puede apreciar en la Figura 3.32. El cual se encarga de verificar si hubo errores o cambios en la transmisión y luego retira el código de detección de errores establecido en el transmisor. Para ello en la configuración de este bloque únicamente se lo coloca en modo “*Check CRC*” y se debe colocar la longitud, en este punto únicamente se hace uso de la variable: “*packet_len*”, esto se debe al hecho de que para retirar el código de verificación de errores correctamente se debe ingresar ser la misma longitud que se usó en el transmisor [12].

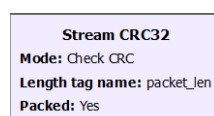


Figura 3.32 Bloque CRC32.

3.4 Investigar técnicas para reducir el PAPR.

Para la contrarrestar el PAPR se han creado varias técnicas las cuales se las puede clasificar en tres familias grandes. Cada familia tiene una característica que la diferencia de las demás. La primera familia se centra en modificar la fase y la amplitud de la señal. La segunda familia agrega subportadoras que carezcan de información útil y la tercera familia reordena la información útil en las subportadoras [14].

Técnicas de la primera familia

- ***Channel Coding***

Es un método que tiene como función localizar las secuencias en la señal que provocan niveles muy altos y bajos del PAPR, esto con el propósito de evitar ocupar partes de la señal que provocan picos de potencia elevado. Para implementarla esta técnica se debe cifrar las secuencias de datos que se quiere transmitir. Por consiguiente, para aplicar esta técnica es imprescindible: ordenar las secuencias, cifrar los datos y transmitir [14].

- ***Clipping and filtering***

Es un método simple de reducción de PAPR pero se fundamenta en recortar los picos que presenta la amplitud de la señal en el dominio del tiempo. Consiste en fijar un límite y recortar todos los niveles que sobrepasen ese valor, en otras palabras, se le conservará la fase, sin embargo se establecerá un límite de amplitud configurado con anterioridad. Las muestras que no excedan dicho límite no serán alteradas [14].

Este procedimiento contiene un obstáculo debido que al recortar las muestras se agrega una distorsión, la misma distorsión puede llegar a interpretarse como una fuente de ruido. El ruido generado puede alterar la señal dentro o incluso fuera del ancho de banda. La desviación interna es imposible de suprimir pero la desviación externa se podrá mitigar en base del filtrado. Este procedimiento no solicita un alto requerimiento de recursos computacionales aunque esto no es eficiente debido a que deforma la señal. Para reforzarla a menudo se mezcla con otras técnicas [14].

- ***Partial Transmit Sequence (PTS)***

El símbolo de OFDM que se encuentra en el dominio de la frecuencia en este método es dividido en secciones para luego multiplicar cada una de ellas por un valor fijo. De tal manera se logra variar en la señal que se desea transmitir la amplitud y la fase. Gracias a esto se cambia la señal que está en el dominio del tiempo y su nivel de PAPR. Cabe mencionar que existen resultados de la multiplicación antes mencionada que se

encargan de elevar el nivel del PAPR y otros que lo restan. Una solución que evita los potenciales problemas es usar múltiples valores fijos. Gracias a esto es posible seleccionar varias mezclas entre los subbloques y los valores fijos permitiendo escoger mediante de un selector la mezcla que tenga un bajo PAPR [14].

- ***Selective Mapping (SLM)***

Es un método semejante a PTS la diferencia radica en que la secuencia no se subdivide. Sino que los símbolos OFDM que están en el dominio de la frecuencia son ampliados gracias a la multiplicación de un vector conformado por constantes con longitud uniforme, dicho resultado permitirá oscilar la amplitud, la fase y el nivel de PAPR de la señal temporal a enviar [14].

Esta técnica logra generar un conjunto de vectores constituidos por constantes y por medio de un selector se elegirá un conjunto que refleje el resultado con menor PAPR, aunque el receptor no conocerá el vector escogido a menos que sea notificado. Por ende será elemental transmitir a la par con la señal de información un indicador que notifique el vector elegido. Esta es la razón por la que se aprecia una disminución de la eficiencia del envío y un aumento de potencia [14].

- ***Active Constellation Extension***

Consiste en desplazar hacia el exterior algunos puntos de la constelación de cada una de las subportadoras con el propósito de disminuir el PAPR. De la misma forma que se reduce el BER en un sistema de comunicación en el que los puntos de su constelación están alejados entre sí ya que facilita el trabajo del decisor también puede utilizarse para reducir el PAPR [14].

Para la modulación OFDM los símbolos que pertenecen al diagrama de constelación son modificados y que pertenecen a una subportadora en especial y probablemente la fase de la señal sinusoidal correspondiente. Por lo tanto, de darse el caso de que los desplazamientos son adecuados para las constelaciones que intervienen algunas subportadoras como en OFDM es posible ejecutar la modificación de las señales sinusoidales que se adicionan entre las mismas para la reducir los picos de la señal en el dominio del tiempo. Cabe mencionar que se puede efectuar modulaciones como BPSK, QPSK, 16QAM, etc [14].

Técnicas de la segunda familia

- ***Flipping***

Esta técnica tiene como propósito añadir señales del tipo sinusoides que carecen de información útil, a estas se las conoce como: “*dummy subcarriers*”, esto se lo realiza con el fin de que se adicione destructivamente con las señales sinusoidales resultantes en los instantes en que se manifiestan picos de potencia en el dominio del tiempo. Por tal motivo el transmisor está obligado a no utilizar ciertas subportadoras con el propósito de agregar las señales del tipo sinusoides [14].

- ***Swapping***

Este procedimiento se fundamenta en establecer un valor que sirva como una referencia máxima requerida para la señal a tratar y estimar el error en todos los valores que crucen ese valor máximo. Por otro lado, es fundamental mencionar que aunque esta técnica es idéntica al de *Flipping*. Sin embargo, se diferencia porque las denominaciones de las subportadoras se incorporan en el dominio de la frecuencia. Además en esta técnica se han logrado hallar más valores óptimos gracias a un algoritmo [14].

Técnica de la tercera familia

- ***Interleaving***

Esta técnica tiene como objetivo principal cambiar los lugares de los coeficientes del símbolo OFDM que se encuentran en el dominio de la frecuencia. Lo que quiere decir es que se canjean las entradas del bloque IFFT. Por ende, la consecuencia de la multiplicación de los coeficientes por las exponenciales al ejecutar la IFFT son distintos. Además se encarga de cambiar símbolo OFDM por otro por medio de intercambiar sus partes, lo cual impulsa el progreso de la referencia de PAPR o no [14].

Por medio del selector se elige la que tenga los resultados más altos y será retransmitida. Por consiguiente, es fundamental enviar un código por el que se facultará reconstruir el símbolo OFDM enviado. Asimismo, en el receptor se debe conocer los distintos cambios ejecutados, los mismos que se identificarán como los que han sido utilizados a través del código y se les dará el uso de recobrar palabra OFDM [14].

3.5 Elegir una técnica para reducir el PAPR.

La sección anterior mostró los tipos de familias y métodos que existen para tratar el PAPR. Para este proyecto de titulación se escoge una técnica de la primera familia para evitar la agregación de subportadoras y el reordenamiento de la carga útil que es la manera como operan las familias dos y tres, de esa manera se previene la saturación y pérdida de información.

De entre la familia uno se eligió la técnica del *Channel Coding* ya que solo codifica la señal en lugar de limitar la señal a parámetros establecidos así como las demás técnicas de la familia uno. Esta codificación es suficiente para afrontar el problema del PAPR y su implementación en GNU Radio es sencilla, por lo que se decide insertarla en la simulación de este proyecto.

La codificación de canal se dedica a estudiar las técnicas y procedimientos para llevar a cabo la detección y corrección de errores producidos como consecuencia del ruido y distorsión introducidos por los sistemas de transmisión en el canal de comunicación o en medios de grabación. Conviene señalar que la codificación de canal no tiene que ver con la codificación de fuente.

El codificador de canal tiene como entrada una señal digital procedente del codificador de fuente. El codificador de canal es una secuencia de bits cuya integridad debe proteger para que pueda ser reproducida fielmente en el receptor [15]. Esta técnica se implementa inmediatamente a la salida de la fuente y antes de la modulación de la señal para limitar los picos presentes y evitar posibles saturaciones en el dispositivo transmisor.

3.6 Implementar *Channel Coding* en GNU Radio.

Para agregar esta técnica de reducción de PAPR, se deben agregar 3 bloques antes de la etapa de etiquetado en el receptor. El primer bloque *Scrambler* que se puede apreciar en la **Figura 3.33** cumple la función de codificador para el flujo de entrada, la razón de colocarlo antes del etiquetado de OFDM es que solo funciona en flujo desempaquetados.

Los parámetros configurados vienen por defecto para este bloque de GNU Radio contemplado en la Figura 3.33. Estos parámetros los cuales se conocen como: “*mask*” el cual es el tamaño en este caso 138 que permite cifrar la información de la señal a transmitir y se lo escribe de manera hexadecimal en la configuración del bloque [9] [12].

El segundo parámetro es “*seed*” que almacena un registro inicial de desplazamiento para este proyecto tiene un valor de 127, este parámetro depende de la señal inicial para luego ir cambiando y de esa manera administrar la forma en cómo se cargan y acceden a la información de la señal que está almacenada, todo esto de acuerdo con la longitud del desplazamiento. Por último, se configura la longitud de desplazamiento del registro que permite la administración del segundo parámetro [9] [12].

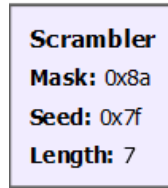


Figura 3.33 Bloque *Scrambler*.

El segundo bloque ejecuta una codificación adicional del tipo convolucional para el tipo de palabras código a utilizarse y facilitar el despliegue correcto de la codificación, pues añade redundancia y de esa manera durante la codificación las codificaciones adquieran una guarda. Este bloque se conoce como: *Encode CCSDS27* y se puede apreciar en la Figura 3.34. La razón de usar este bloque es que *Channel Coding* busca codificar varias veces para reducir el PAPR. Este bloque recibe los en su entrada bits en forma de flujo que han sido empaquetados y a la salida de este bloque se obtiene flujos de símbolos con codificación binaria, puede ser 0 o 1 [9] [12].

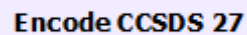


Figura 3.34 Bloque CCSDS27.

Como último bloque se presenta *Interleave* el cual permite una intercalación de las entradas o flujos de las palabras o información que ha sido codificada. Se puede apreciar en la Figura 3.35. Este bloque se lo agrega con el propósito de mantener un flujo uniforme de los símbolos y no exista errores [9].

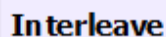


Figura 3.35 Bloque *Interleave*.

A la salida de este bloque se puede continuar con el proceso de las etapas del transmisor antes descritas para OFDM.

3.7 Realizar pruebas de funcionamiento de OFDM y *Channel Coding* en GNU Radio.

Para la verificación del funcionamiento de la simulación de OFDM se agrega bloques visualizadores que funcionan a manera de osciloscopios para observar la señal y poder medir la potencia de esta. Cabe recalcar que los bloques de GNU Radio tienen

diferentes tipos de salidas de datos, en este proyecto se usaron del tipo: *complex*, *byte* y *float*.

Para poder apreciarlas se usaron bloques que convirtieron esos formatos en el tipo de salida requerido para que pueda ser visualizada la señal. Por ejemplo: *Complex to Float* y *Uchart to float*, para convertir de complejo a real y de byte a real respectivamente. Esta conversión permite conectar la señal a los bloques visualizadores de GNU Radio, los mismos que pueden monitorear la señal en el dominio de la frecuencia y el tiempo. Para este proyecto usaremos el bloque que permite visualizar la señal en el dominio de la frecuencia [8].

El bloque visualizador se denomina *QT GUI Sink* mostrado en la Figura 3.36. A este bloque se le puede colocar el nombre que se desea, de preferencia el lugar o alguna característica que permita identificarlo al momento de ejecutar la simulación. El parámetro *FFT Size* es el valor de las muestras que se quieren observar y de esa manera contemplar la señal con una alta o baja precisión.

Se configura la frecuencia central para apreciar la señal con mayor fidelidad, pero por defecto se usa cero. El ancho de banda se colocó con el valor de la frecuencia en la que trabaja la variable del proyecto: "sample_rate" la cual viene por defecto en GNU Radio, y las actualizaciones se dejan en 10 para no mostrar muchos cambios en la señal y poder estudiarla mejor [8].

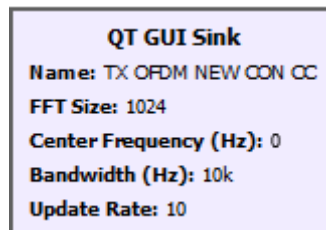


Figura 3.36 Bloque de Análisis de la señal.

Verificación de funcionamiento del Transmisor OFDM

Como primer punto se analiza mediante los visualizadores si se ha podido simular correctamente OFDM. Para ello el bloque visualizador se conectó a la salida de la fuente. El bloque que genera la señal inicial es de tipo aleatoria sin ningún tipo de modificación y esto se puede apreciar en la Figura 3.37 la señal aleatoria generada inicialmente.

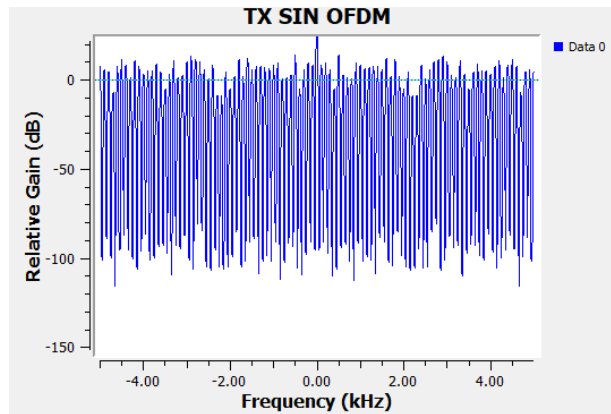


Figura 3.37 Señal aleatoria sin OFDM.

Luego se conecta la salida del transmisor al bloque visualizador para corroborar que se ha modificado la señal para ser del tipo OFDM como se aprecia en la **Figura 3.38**. Esta señal presenta el espectro ensanchado que caracteriza a las señales OFDM. La señal tiene, a modo de comparación, la forma de un pulso binario característica de las señales OFDM lo cual demuestra que se ha logrado modular en OFDM cumpliendo con el objetivo que se tenía previsto.

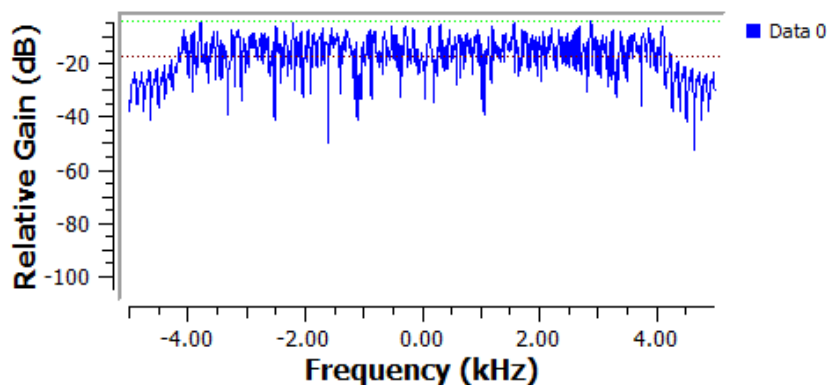


Figura 3.38 Señal OFDM a la salida transmisor.

Verificación de funcionamiento del Receptor OFDM

Para la verificación del funcionamiento del receptor OFDM se debe colocar los bloques visualizadores a la salida de la fuente de la señal y a la salida del proceso de demodulación del receptor. Esto se debe a que al final del proceso de recepción si la señal obtenida después de todo el proceso en el receptor no es la misma que la señal inicial, entonces el proceso tiene errores.

Sin embargo, como se observa en la **Figura 3.39** la señal a la salida de la fuente (en la derecha) no presenta demasiadas alteraciones, salvo las causadas por la agregación de subportadoras de guarda y el procesa para retirarlas, comparadas a la señal a la

salida del receptor (en la izquierda). Aparentemente es la misma señal que la original lo que permite concluir que el procesamiento de la señal en el receptor funciona correctamente y se ha logrado recuperar la señal original después de la modulación OFDM.

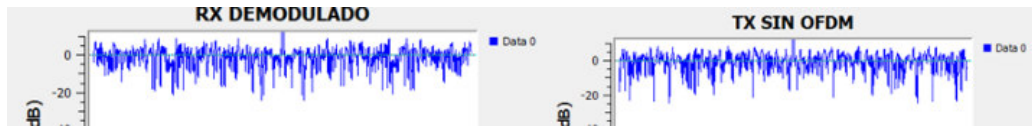


Figura 3.39 Comparación de la señal al inicio y fin de OFDM.

Verificación de funcionamiento de *Channel Coding*

Para verificar la reducción del PAPR junto al diagrama de bloques del transmisor se construyó otro transmisor OFDM idéntico al original con la diferencia que se agregaron los bloques que permiten simular la técnica *Channel Coding*. Ambos transmisores se conectaron a la misma fuente.

Para medir la reducción del PAPR se utiliza la **¡Error! No se encuentra el origen de la referencia.**, para ello es importante conseguir la potencia pico y la potencia media. La potencia pico se consigue midiendo la señal en el bloque visualizador y aplicando la **Ecuación 3.1**.

$$\text{Potencia} = \text{Pico máx} - \text{Pico mín}$$

Ecuación 3.1 Potencia

Donde:

- Potencia : (db) Potencia de la señal.
- Pico máx : (db) Pico máximo de la señal.
- Pico mín : (db) Pico mínimo de la señal.

Para obtener el valor de la potencia promedio también se requiere usar el bloque RMS mostrado en la Figura 3.40 el cual solo tiene un parámetro: *Alpha*, este parámetro es la ganancia del filtro promedio móvil que para este proyecto solo se lo coloca en 100u, suficiente para detectar la potencia media que se requiere. A la salida de este bloque se conecta un visualizador para medir el pico máximo y el pico mínimo de la potencia media. La potencia media se consigue midiendo la señal promedio en el bloque visualizador y aplicando la **Ecuación 3.1**.

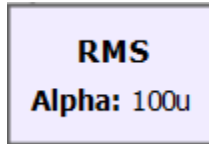


Figura 3.40 Bloque RMS.

La señal de la potencia promedio de la modulación OFDM se puede observar en la **Figura 3.41**.

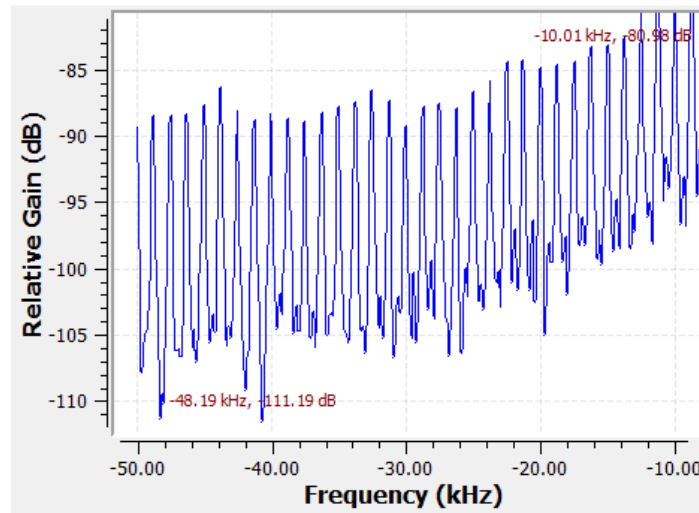


Figura 3.41 Potencia promedio OFDM sin *Channel Coding*.

Usando la Ecuación 3.1 se obtiene:

$$\text{Potencia promedio} = 30.21(\text{db})$$

La señal OFDM resultante sin el *Channel Coding* se puede observar en la **Figura 3.42**.

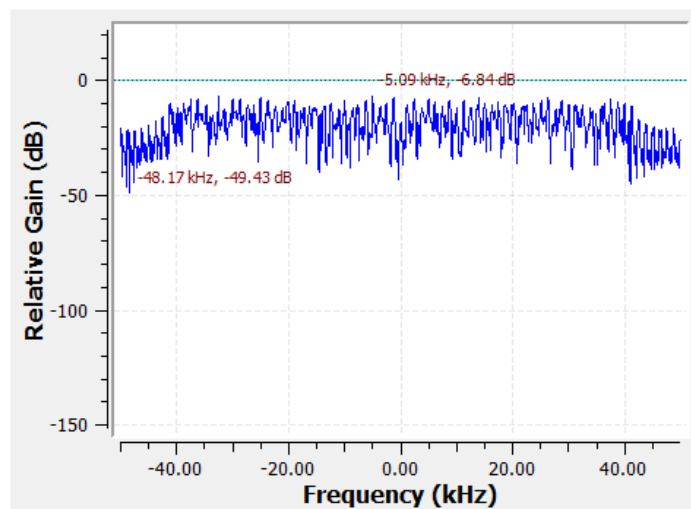


Figura 3.42 Potencia pico OFDM sin *Channel Coding*.

Usando la Ecuación 3.1 se obtiene:

$$\text{Potencia pico} = 42.59(\text{db})$$

La señal de la potencia promedio de la modulación OFDM se puede observar en la **Figura 3.4343**

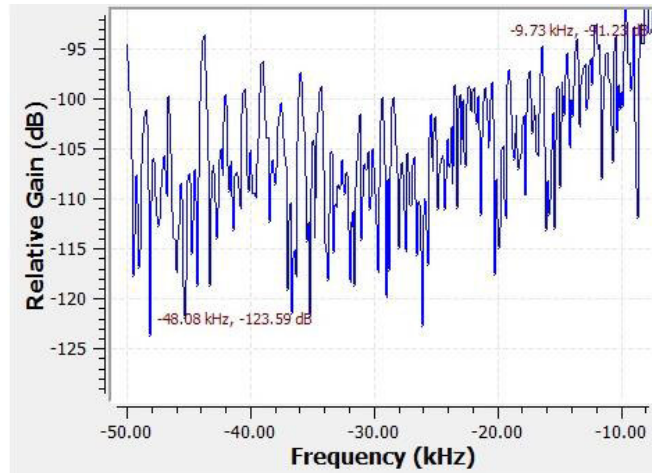


Figura 3.4343 Potencia promedio OFDM con *Channel Coding*.

Usando la Ecuación 3.1 se obtiene:

$$\text{Potencia promedio} = 32.36(\text{db})$$

La señal OFDM resultante con el *Channel Coding* se puede observar en la **Figura 3.44**.

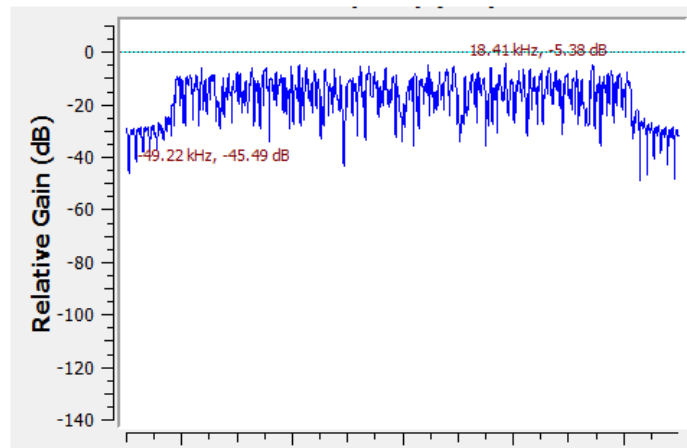


Figura 3.4444 Potencia pico OFDM con *Channel Coding*.

Usando la Ecuación 3.1 se obtiene:

$$\text{Potencia pico} = 40.11(\text{db})$$

Los valores calculados nos permiten determinar la potencia promedio, la potencia máxima y realizar un análisis de la reducción del PAPR en OFDM como se muestra en la Tabla 3.1.

Tabla 3.1 Resultados de la reducción del PAPR en la modulación OFDM.

MODULACIÓN	Potencia Promedio [db]	Pico Máximo [db]	PAPR
OFDM SIN CC	30.21	42.59	1.98
OFDM CON CC	32.36	40.11	1.53

Gracias a los valores obtenidos en los cálculos de las potencias se puede determinar que se ha reducido el PAPR al restar los valores de PAPR antes y después de usar *Channel Coding*. Estos valores muestran una disminución de 0.45 de PAPR en el sistema OFDM. Cumpliendo el objetivo planteado.

4 CONCLUSIONES

- Investigar acerca del proceso para realizar OFDM en GNU Radio permitió identificar que es fundamental dividir la aplicación en dos partes (transmisor y receptor) para ejecutar en un equipo de bajas prestaciones de hardware y, asimismo estudiar la reducción de PAPR sin fallas en el programa y de una manera sencilla sin muchas complicaciones.
- Mediante la investigación de las técnicas para contrarrestar el PAPR en GNU Radio se concluye que el método de *Channel Coding* con su bloque de codificación y corrección de errores CCSDS27 es fundamental para optimizar el sistema de comunicación, ya que no se requiere de una retransmisión de la información, realizando la corrección de errores directamente en el receptor. Asimismo, evita posibles saturaciones en el dispositivo transmisor.
- Las pruebas de funcionamiento de OFDM y *Channel Coding* realizadas permitieron evidenciar y verificar que la reducción del PAPR en el sistema. Sin duda alguna, para obtener el valor de la potencia promedio, se requiere usar el bloque RMS para mayor facilidad de cálculos.
- Al Implementar la técnica *Channel Coding* se puede concluir que es una técnica eficaz en la reducción del PAPR en GNU Radio, ya que al final de la modulación OFDM se demostró mediante una comparación la reducción en el PAPR de la señal .
- Las técnicas de la primera familia para reducir el PAPR son más sencillas de implementar al no modificar de forma sustancial la información útil de la señal, sin embargo para una reducción eficiente las técnicas de las familias restantes son las ideales aunque requieren de mayor procesamiento y presentan alta complejidad.
- Al ser GNU Radio un programa con una licencia pública general muestra ciertos inconvenientes, en concreto a la hora de mostrar los resultados, estos no se visualizan de una forma tan sencilla y amigable con el usuario. Debido a esto el diagrama desplegado presenta una complejidad al momento de realizar las mediciones por las limitaciones que presentan las herramientas de medición.
- Al terminar con la simulación de la modulación OFDM en GNU Radio se concluye que esta modulación tiene una complejidad media alta a comparación de otros programas informáticos, esto se debe a que requiere cierto grado de configuración de funciones en ciertos bloques y variables, especialmente para la

modulación. Sin embargo, esto le permite variar más parámetros que otros programas no poseen.

5 RECOMENDACIONES

- Se recomienda instalar GNU Radio en un computador con una buena capacidad de cómputo, ya que consume muchos recursos, de lo contrario presenta congelamiento de pantalla y puede fallar el programa cerrándose y perdiendo la información y las configuraciones sino no se guardaron con antelación.
- Para evitar perder información se recomienda cada vez que se termine de configurar un bloque presionar las teclas: Ctrl + s para guardar el progreso durante el desarrollo de proyectos en GNU Radio.
- Al momento de agregar variables en las configuraciones de los bloques se recomienda usar el mismo nombre de los parámetros para evitar confusiones cuando se coloquen más variables.
- Es recomendable usar los bloques sincronizadores para mantener un orden y una jerarquía al desplegar los bloques, de lo contrario el apilamiento de bloques puede confundir al usuario en el desarrollo de nuevos diagramas.
- Si se desea reducir de manera más profunda el PAPR en la modulación OFDM se recomienda usar dos métodos combinados para conseguirlo.
- En las configuraciones si se desea agregar más de una palabra se recomienda usar guion bajo como una manera de separar las palabras, ya que si se presiona la tecla de espacio, inmediatamente se genera un error en el bloque.
- Se recomienda colocar nombres específicos en los bloques visualizadores para identificar que se está midiendo, de esa manera se pueden realizar las correspondientes mediciones con facilidad, esto es importante ya que GNU Radio en sus últimas versiones coloca los resultados en una misma ventana indiscriminadamente.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] J. Herrera y V. Rodríguez, «RIUTPL,» 2018. [En línea]. Available: <https://dspace.utpl.edu.ec/bitstream/20.500.11962/21683/1/Herrera%20Bustamante%20Jos%c3%a9%20Enrique%20y%20Rodr%c3%adguez%20Lude%c3%b1a%20Vanessa%20Stefan%c3%ada.pdf>. [Último acceso: 05 Septiembre 2022].
- [2] MathWorks, «MathWorks,» [En línea]. Available: <https://la.mathworks.com/discovery/ofdm.html>. [Último acceso: 27 Julio 2022].
- [3] J. Armstrong, «La Trobe University,» 2002. [En línea]. Available: https://d1wqtxts1xzle7.cloudfront.net/32214201/02_armstrong_ofdm-with-cover-page-v2.pdf?Expires=1658939748&Signature=KWMJxev471IY3jdP0BR6E6Un3vIRKQLTEyoWHKB0TIUuqzLrghrm994BgiFuFl6vM0QDrjMmt2uaNguUUYxaGrmqtG0UthwiNzryBBnb2IVRYA7PfJRmhJeq1AM-MSc0RD8rhV2QVN. [Último acceso: 27 Julio 2022].
- [4] F. Lincango, «Repositorio Digital - EPN,» Octubre 2018. [En línea]. Available: <https://bibdigital.epn.edu.ec/handle/15000/19827>. [Último acceso: 27 Julio 2022].
- [5] «Biblius,» [En línea]. Available: <https://biblus.us.es/bibing/proyectos/abreproy/11983/fichero/Capitulo+3+-+OFDM.pdf>. [Último acceso: 05 Septiembre 2022].
- [6] M. Paredes, «Dialnet,» Diciembre 2014. [En línea]. Available: https://e-archivo.uc3m.es/bitstream/handle/10016/20549/tesis_martha_cecilia_paredes_paredes_2014.pdf?sequence=1&isAllowed=y. [Último acceso: 27 Julio 2022].
- [7] D. Simba, «Repositorio Digital - EPN,» Febrero 2018. [En línea]. Available: <https://bibdigital.epn.edu.ec/bitstream/15000/19179/1/CD-8563.pdf>. [Último acceso: 27 Julio 2022].
- [8] R. S. Goswami, «IJAREEIE,» July 2017. [En línea]. Available: https://www.ijareeie.com/upload/2017/july/71_GNU.pdf. [Último acceso: 27 Julio 2022].
- [9] «GNU RADIO,» GNU RADIO, 2022. [En línea]. Available: <https://www.gnuradio.org/about/>. [Último acceso: 27 Julio 2022].

- [10] V. Tripathi y K. Bansal, «EE IITB,» [En línea]. Available: <https://www.ee.iitb.ac.in/course/~vishrant/ofdm-tranmission-reception.pdf>. [Último acceso: 05 Septiembre 2022].
- [11] R. Erreyes y M. López, «Repositorio Digital - EPN,» Octubre 2019. [En línea]. Available: <https://bibdigital.epn.edu.ec/handle/15000/20512>. [Último acceso: 27 Julio 2022].
- [12] «Wiki GNU Radio,» Junio 2022. [En línea]. Available: <https://wiki.gnuradio.org/index.php/InstallingGR>. [Último acceso: 27 Julio 2022].
- [13] L. Jiménez, J. Parrado, C. Quiza y C. Suárez, «Universidad Distrital Francisco José de Caldas,» Noviembre 2000. [En línea]. Available: [https://revistas.udistrital.edu.co/index.php/reving/article/view/2699/3891#:~:text=OFDM%20\(Multiplexaci%C3%B3n%20por%20Divisi%C3%B3n%20de%20Frecuencia%20Ortogonal\)%20es%20un%20m%C3%A9todo,y%20se%20multiplexa%20en%20frecuencia.](https://revistas.udistrital.edu.co/index.php/reving/article/view/2699/3891#:~:text=OFDM%20(Multiplexaci%C3%B3n%20por%20Divisi%C3%B3n%20de%20Frecuencia%20Ortogonal)%20es%20un%20m%C3%A9todo,y%20se%20multiplexa%20en%20frecuencia.) [Último acceso: 27 Julio 2022].
- [14] R. Quijada, «Universidad Politécnica de Catalunya,» 04 Julio 2008. [En línea]. Available: <https://upcommons.upc.edu/bitstream/handle/2099.1/5078/memoria.pdf?sequence=1>. [Último acceso: 23 Agosto 2022].
- [15] P. Vila, «Albentia Systems,» Albentia Systems, 5 Septiembre 2013. [En línea]. Available: <https://blog.albentia.com/2013/09/05/modulacion-ofdm-wimax-madrid/>. [Último acceso: 27 Julio 2022].

7 ANEXOS

ANEXO I: Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 12 de septiembre de 2022

De mi consideración:

Yo, FERNANDO VINICIO BECERRA CAMACHO, en calidad de Director del Trabajo de Integración Curricular titulado SIMULACIÓN DE OFDM MEDIANTE GNU RADIO elaborado por el estudiante PATRICIO SEBASTIAN VALLEJO GUALACEO de la carrera en TECNOLOGÍA SUPERIOR EN REDES Y TELECOMUNICACIONES, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 9%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el link del informe generado por la herramienta Turnitin.

[https://epnecuador-my.sharepoint.com/:b:/g/personal/fernando_becerrac_epn_edu_ec/Ee-rPmzyQNBjk7qLJ4v7EJ4BtbWYfAdfsITg6t8MLzpSYQ?e=gS6Gf1Atentamente,](https://epnecuador-my.sharepoint.com/:b:/g/personal/fernando_becerrac_epn_edu_ec/Ee-rPmzyQNBjk7qLJ4v7EJ4BtbWYfAdfsITg6t8MLzpSYQ?e=gS6Gf1Atentamente)



Fernando Vinicio Becerra Camacho

Docente

Escuela de Formación de Tecnólogos

ANEXO II: Video de evidencia y prueba de funcionamiento

