

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**SIMULACIÓN DE LA TÉCNICA DE COMUNICACIÓN
INALÁMBRICA NOMA**

**SIMULACIÓN DE CFO (CARRIER FREQUENCY OFFSET) EN LA
TÉCNICA NOMA**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

JAIRO MARCELO VIRACOCCHA TOAPANTA

jairo.viracocha@epn.edu.ec

DIRECTOR: DR. DIEGO JAVIER REINOSO CHISAGUANO

diego.reinoso@epn.edu.ec

DMQ, febrero 2022

CERTIFICACIONES

Yo, JAIRO MARCELO VIRACOCCHA TOAPANTA 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.

JAIRO MARCELO VIRACOCCHA TOAPANTA

Certifico que el presente trabajo de integración curricular fue desarrollado por JAIRO MARCELO VIRACOCCHA TOAPANTA, bajo mi supervisión.

DR. DIEGO JAVIER REINOSO CHISAGUANO
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) 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.

JAIRO MARCELO VIRACOCCHA TOAPANTA

DR. DIEGO JAVIER REINOSO CHISAGUANO

DEDICATORIA

Dedico este trabajo con todo mi corazón a mis padres Agustín y Diocelina quienes son el principal motor de mi vida ya que siempre me han brindado su apoyo incondicional y han sabido formarme con buenos valores, motivándome siempre para alcanzar cada una de mis metas.

AGRADECIMIENTO

A Dios por brindarme salud, permitirme cumplir una más de mis metas, ser el soporte en los días más difíciles y por mantener a mis padres y familia junto a mí.

A mis padres Agustín y Diocelina por guiarme y apoyarme en cada uno de mis días, por no permitir que desista de esta meta, siempre estaré agradecido por todo el sacrificio que han hecho por brindarme una buena educación e impulsarme a ser cada día mejor. Gracias por ser quienes son y por confiar en que lo lograría.

A mis abuelitos Feliciano, Pedro, Manuela, Luisa, Diocelina por haber sido mis segundos padres, por brindarme sus consejos, apoyo, cariño y haberme cuidado cuando más lo necesitaba.

A mis tíos Víctor y Mercedes que siempre han estado pendientes de mi desde pequeño, me han cuidado y me han brindado su apoyo incondicionalmente.

A mis primos Juan, Roberto, Cesar, Mario y Edgar como también para sus esposas por haberme apoyado en cada uno de los momentos que atravesé.

A mi tío Genaro y tías Bertha, Luz María, Marisol, Marlene y Carolina por cada uno de sus consejos y su apoyo durante toda mi etapa universitaria.

A cada uno de mis tíos y primos por brindarme sus consejos y motivarme en los días más difíciles.

A la Escuela Politécnica Nacional por haberme permitido formarme en esta grandiosa institución, así como a cada uno de los docentes que me han impartido sus conocimientos. De manera especial al Dr. Diego Reinoso por su apoyo, paciencia y consejos que siempre han sido valiosos para que este trabajo culmine cumpliendo lo esperado.

A Gaby por brindarme su apoyo durante toda la carrera, por sus palabras de aliento en los momentos más difíciles y siempre estar pendiente de mí.

A los de siempre, Karolina Goyes, Esteban Machado, Arianna Rivas, Sebastián Mullo, Alan Crespo, Yarik Ramos y Leo Cando; por brindarme esa grandiosa amistad, soy creyente de que el éxito en la universidad va de la mano con los amigos que uno tiene y ustedes me lo han comprobado, gracias por compartir experiencias que jamás olvidaremos y también por estar en los momentos buenos y malos.

Gracias a todos.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT.....	VIII
1.INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO	3
1.4.1 COMUNICACIONES INALÁMBRICAS.....	3
1.4.2 OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING).4	
1.4.3 NOMA (NON-ORTHOGONAL MULTIPLE ACCESS).....	7
1.4.4 SINCRONIZACIÓN PARA NOMA-OFDM.....	8
2. METODOLOGÍA.....	11
2.1 INICIALIZACIÓN.....	13
2.2 TRANSMISOR OFDM-NOMA.....	14
2.2.1 GENERACIÓN DE BITS ALEATORIOS.....	15
2.2.2 CODIFICACIÓN FEC.....	15
2.2.3 MODULACIÓN.....	15
2.2.4 SUMA DE SEÑALES E INSERCIÓN DEL FACTOR DE INYECCIÓN.....	16
2.2.5 MODULACIÓN OFDM.....	16
2.2.6 GENERACIÓN DEL PREAMBULO E INSERCIÓN A LA SEÑAL A TRANSMITIRSE.....	17
2.3 CANAL RICIAN.....	18
2.4 RECEPCIÓN.....	20
2.4.1 SINCRONIZACIÓN CFO.....	21
2.4.2 RECEPTOR DE CADA CAPA.....	26

2.5	CÁLCULO DE BER.....	29
2.6	GENERACIÓN DE ARCHIVO DE DATOS.....	29
2.7	GENERACIÓN DE GRAFICA.....	30
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	31
3.1	RESULTADOS.....	31
3.1.1	ESCENARIO DE SIMULACIÓN.....	31
3.1.2	PRUEBAS DE FUNCIONAMIENTO.....	31
3.1.3	RESULTADOS CON SINCRONIZACIÓN DEL CFO.....	33
3.2	CONCLUSIONES	40
3.3	RECOMENDACIONES	42
4	REFERENCIAS BIBLIOGRÁFICAS.....	43
5	ANEXOS	44
	ANEXO I	44

RESUMEN

El presente trabajo lleva a cabo un análisis de las técnicas de sincronización para un sistema de comunicación NOMA (*Non-Orthogonal Multiple Access*) OFDM (*Orthogonal Frequency Division Multiplexing*), enfocándose en la sincronización del CFO (*Carrier Frequency Offset*). Para este fin, se implementa una simulación en el software MATLAB del sistema de comunicación mencionado anteriormente en donde el proceso de sincronización se realiza en la sección del receptor. Esta sincronización se lleva a cabo mediante tres técnicas de estimación del CFO, dos de estas trabajan en el dominio del tiempo y la restante trabaja en el dominio de la frecuencia.

El capítulo 1 presenta las características del sistema de comunicación a implementarse así también las características y afectaciones que sufren las comunicaciones inalámbricas. Se examina OFDM y la técnica de acceso NOMA para obtener un sistema en conjunto y posteriormente se presentan las técnicas de sincronización del CFO.

El capítulo 2 muestra el desarrollo paso a paso del código implementado en el software MATLAB en cual está formado por bloques o secciones, mismos que son: transmisor, canal y receptor, considerando que en este último se realiza el proceso de sincronización. Con el objetivo de presentar detalladamente el código implementado, se describen cada una de la funciones y comandos usados para que el sistema de comunicación pueda funcionar de la manera esperada.

El capítulo 3 presenta los resultados obtenidos mediante graficas de BER (*Bit Error Rate*) Vs. E_b/N_0 (*Energy per bit to the Spectral Noise Density*) de cada una de las simulaciones ya que cada una de estas presenta distinta configuración de capas, diferente factor de inyección y diferente técnica de estimación de CFO. Finalmente, se presentan las conclusiones y recomendaciones obtenidas en base al desarrollo del presente trabajo.

PALABRAS CLAVE: NOMA, OFDM, CFO, BER.

ABSTRACT

The present work carries out an analysis of the synchronization techniques for a NOMA (Non-Orthogonal Multiple Access) OFDM (Orthogonal Frequency Division Multiplexing) communication system, focusing on the synchronization of the CFO (Carrier Frequency Offset). For this purpose, a simulation is implemented in the MATLAB software of the communication system where the synchronization process is carried out in the receiver section, this synchronization is carried out through three CFO estimation techniques, two of these work in the time domain and the rest works in the frequency domain.

Chapter 1 shows the characteristics of the communication system to be implemented as well as the characteristics and effects suffered by wireless communications. OFDM and the NOMA access technique are examined to obtain an overall system, and then the CFO synchronization techniques are presented.

Chapter 2 shows the step-by-step development of the code implemented in the MATLAB software in which it is made up of blocks or sections, which are: transmitter, channel and receiver, considering that the synchronization process is carried out in the latter. In order to present the implemented code in detail, each of the functions and commands used so that the communication system can work as expected are described.

Chapter 3 presents the results obtained through graphs of BER (Bit Error Rate) Vs. E_b/N_0 (Energy per bit to the Spectral Noise Density) of each of the simulations, since each of these presents a different configuration of layers, different injection factor and different CFO estimation technique. Finally, the conclusions and recommendations obtained based on the development of this work are presented.

KEYWORDS: NOMA, OFDM, CFO, BER.

1 INTRODUCCIÓN

Las redes móviles se han visto involucradas en un proceso evolutivo con un crecimiento exponencial debido a esto se ha provocado alta demanda por parte de los usuarios, ya que éstos buscan un servicio más eficiente con altas velocidades de transmisión de datos. En los últimos años el número de aplicaciones van incrementando diariamente lo que motiva a un mayor desarrollo de tecnologías que puedan soportar los nuevos requerimientos, además la gran expansión en el número de dispositivos conectados provoca una enorme explosión del tráfico de datos por lo que es evidente que la próxima generación tiene que soportar altas velocidades de transmisión, por esta razón la industria de aplicaciones y el mundo académico coinciden en la necesidad de tecnologías de acceso por radio nuevas y flexibles.

Las técnicas de acceso múltiple juegan un papel muy importante en el proceso de comunicación general, desde la primera generación hasta la cuarta generación las técnicas de acceso han distribuido ortogonalmente a los usuarios sobre los recursos disponibles como por ejemplo OFDMA (*Orthogonal Frequency Division Multiple Access*), sin embargo, los investigadores se han alejado de utilizar los recursos de forma ortogonal a uno no ortogonal, lo conocido como acceso múltiple no ortogonal NOMA (*Non-Orthogonal Multiple Access*), también conocida como multiplexación por división de capas la cual provee una transmisión simultánea para servicios móviles, así también como para los servicios fijos en el mismo canal de frecuencia. NOMA trabaja en el dominio de la potencia lo que permite una transmisión simultánea de diferentes usuarios variando la potencia de uno de estos que generalmente se le da prioridad de potencia al usuario que pueda tener más problemas en el canal.

NOMA-OFDM (*Orthogonal Frequency Division Multiplexing*) presenta características robustas frente a los problemas que puede presentar el canal inalámbrico, uno de los problemas principales comparados con los sistemas mono portadora es que estos son muy sensibles a los errores que pueden ser producidos por desfases de tiempo o de frecuencia, de tal manera que es importante agregar a todo este sistema una fase en la que el receptor se pueda sincronizar al transmisor tanto en tiempo como en frecuencia, consiguiendo de esta manera una eficiencia alta de todo el sistema.

En el presente trabajo se realiza una simulación del sistema NOMA-OFDM, en donde se minimiza los efectos causados por la sincronización. Específicamente, los problemas generados por el desplazamiento de frecuencia de portadora, o conocida por sus siglas en

inglés como CFO (*Carrier Frequency Offset*). Para este fin se utilizan tres técnicas de estimación, dos en el dominio del tiempo y una en el dominio de la frecuencia.

1.1 OBJETIVO GENERAL

Implementar una simulación de CFO (*Carrier Frequency Offset*) en la técnica NOMA-OFDM empleando la plataforma de programación MATLAB.

1.2 OBJETIVOS ESPECÍFICOS

1. Detallar un sistema NOMA-OFDM y sus características de funcionamiento.
2. Describir las técnicas de sincronización en frecuencia que pueden ser aplicadas en el receptor
3. Realizar pruebas de funcionamiento con diferentes parámetros de tasa de codificación, modulación y factor de inyección.
4. Analizar los resultados obtenidos en las simulaciones de CFO en el dominio del tiempo y frecuencia y comparar estas técnicas de estimación.

1.3 ALCANCE

Se implementa una simulación en el software MATLAB de un sistema NOMA en el que se acopla dos flujos de información o datos a los que se les conoce como capas (capa 1 y capa 2), las mismas que tienen como destino dos usuarios. Este sistema emplea un codificador convolucional que tiene la capacidad de generar tasas de $1/2$, $2/3$ y $3/4$. Asimismo, se puede usar esquemas de modulación tales como QPSK, 16QAM o 64QAM y el factor de inyección (g) es un parámetro que puede ser configurado en el código. En cuanto a características del canal, este tendrá ruido AWGN y desvanecimiento Rician. Con el fin de poder estimar el CFO en el receptor y luego proceder con la respectiva compensación se transmitirá un preámbulo. Para recuperar cada una de las capas o flujos de datos de cada uno de los usuarios, los receptores trabajarán independientemente en donde el receptor del usuario 1 recuperará la capa 1, mientras que el receptor del usuario 2 tendrá el trabajo de recuperar en primer lugar la capa 1 para luego poder recuperar la capa 2. Como parte de resultados obtenidos por la simulación se obtendrá la curva de BER

(*Bit Error Rate*) Vs. E_b/N_0 (*Energy per-bit to the Spectral Noise Density*) para cada uno de los usuarios.

Los parámetros que se establecerán para cada una de las simulaciones constan de configurar dos diferentes tasas de codificación, así también, para los esquemas de modulación y estas deben ser elegidas basándose en los resultados de las simulaciones, con respecto al factor de inyección (g) se ensayará con dos valores. Una vez obtenidos los resultados se procederá a analizarlos con el fin de establecer las conclusiones y recomendaciones del presente trabajo.

El desarrollo del componente consta de tres fases, estas son: fase de diseño o planteamiento, fase de implementación y por último la fase de pruebas y/o análisis de resultados. En la primera fase se planteará el diagrama de bloques del sistema de comunicación inalámbrica NOMA-OFDM, en donde se detallarán cada una de las secciones que se desarrollan en el componente, como son: el transmisor, canal de comunicación y receptor. Debido a que este componente se enfoca en la sincronización de CFO, dentro del diagrama de bloques se especifica la sección en donde se realizará el proceso de inserción, estimación y compensación que corresponden a cada una de las etapas de la sincronización. En la fase de implementación se llevará a la práctica mediante programación de código en el software MATLAB cada una de las secciones mencionadas en la fase de diseño, en esta fase se establecerán los parámetros de simulación y también se crearán funciones que permitan comprender mejor el desarrollo del código. Por último, se tiene la fase de pruebas y/o análisis de resultados en donde se realizarán las pruebas necesarias con el fin de comprobar la correcta implementación y funcionamiento de la simulación, en esta fase se definirán los parámetros de modulación, tasa de codificación y factor de inyección que muestren un buen rendimiento del sistema en general. Así también, se realizará un análisis de los resultados obtenidos, de esta manera se podrá determinar cuál técnica obtiene un mejor rendimiento.

1.4 MARCO TEÓRICO

1.4.1 COMUNICACIONES INALÁMBRICAS

Las comunicaciones inalámbricas son consideradas como una de las más antiguas formas de comunicación, los gritos o el sonido mismo de los tambores en lugares remotos no necesitaba de un medio guiado tales como cables para poder funcionar. Así también las comunicaciones ópticas más antiguas son consideradas inalámbricas en donde

encontramos las señales de humo, mismas que se basan en la propagación de señales ópticas a lo largo de una conexión de línea de visión. No obstante, las comunicaciones inalámbricas tal como las conocemos actualmente comenzaron con el trabajo de Hertz y Maxwell, quienes ayudaron a que podamos entender el principio de la transmisión de ondas electromagnéticas [1].

Desde la década de los 60's, ha sido una de las áreas más interesantes en el campo de la comunicación actual, hoy en día son la sección en la industria de las comunicaciones con mayor crecimiento debido a la gran cantidad de aplicaciones. El aumento explosivo en la demanda de conectividad inalámbrica impulsado por la telefonía celular ha sido uno de los factores que ha contribuido el crecimiento. Las redes WLAN (*Wireless Local Area Network*) son un complemento indispensable en ciertos campus o negocios, inclusive pueden llegar a ser un reemplazo para las redes cableadas conocidas como LAN (*Local Area Network*).

Con la evolución de las comunicaciones inalámbricas, aplicaciones como telemedicina remota, sensores inalámbricos e incluso hogares inteligentes han podido ser llevados de ideas a sistemas concretos que hoy por hoy pueden ser usados. No obstante, aunque se ha presentado un progreso rápido, se han ido encontrando varios desafíos técnicos que dificultan el diseño de una red inalámbrica con características robustas que brinden un buen rendimiento con el fin de admitir aplicaciones emergentes [2].

El canal inalámbrico es el medio de transmisión usado en este tipo de comunicaciones y de este depende el rendimiento de los sistemas inalámbricos. A diferencia de un medio guiado, el canal inalámbrico es variante con el tiempo, dinámico e impredecible; lo que da como resultado un difícil análisis de una comunicación inalámbrica. Uno de los problemas es la sincronización entre el transmisor y el receptor como consecuencia de las características del canal mencionadas anteriormente, es así como en este trabajo se presentarán ciertas técnicas de estimación y compensación que estarán enfocadas en la sincronización en frecuencia.

1.4.2 OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING)

En los últimos años, se ha tratado de centrar las investigaciones y trabajos en la posibilidad de extender los servicios que ya están disponibles para los usuarios de redes públicas cableadas hacia los usuarios de telecomunicaciones móviles. Una de las características del canal inalámbrico es la recepción multitrayecto, en donde la señal que recibe el equipo receptor no solamente contiene una onda de radio dada por la línea de vista, sino también, una gran cantidad de ondas reflejadas y estas llegan al receptor con diferentes intervalos de tiempo. Estas ondas o señales atrasadas son la consecuencia de las características del

terreno por donde viajan la información en donde se puede encontrar montañas, colina, arboles, edificios o incluso vehículos. Las ondas atrasadas y reflejadas generan interferencia con la onda directa y producen ISI (*Intersymbol Interference*) y esta provoca una degradación significativa del rendimiento de la red.

Como solución a los inconvenientes ya mencionados, se presenta el esquema de transmisión de multiplexación por división de frecuencia ortogonal (OFDM) y es una de las aplicaciones de un esquema de transmisión de datos en paralelo ampliamente utilizado en comunicaciones inalámbricas. Esta ayuda a reducir la influencia del desvanecimiento por múltiples trayectos y hace innecesario el ocupar ecualizadores muy complejos.

Con el fin de incrementar la robustez de la señal frente a problemas del desvanecimiento selectivo en frecuencia, OFDM a más de permitir una multiplexación de frecuencia, también permite la modulación con múltiples portadoras consiguiendo de esta manera que sólo un porcentaje de estas sean afectadas por el canal. De igual forma las portadoras que se vean afectadas por características mismas del canal pueden ser corregidas haciendo uso de códigos de corrección de errores [3].

OFDM define que la frecuencia base es f_b y esta es la portadora que posee la frecuencia más baja. Las subportadoras que la acompañan son múltiplos enteros de la frecuencia base, es decir: $2f_b, 3f_b, \dots, nf_b$, donde n es un numero entero positivo, tal como se indica en la Figura 1.1. Las subportadoras se encuentran superpuestas de tal manera que distinguirlas es muy fácil, a esto se lo llama ortogonalidad [4].

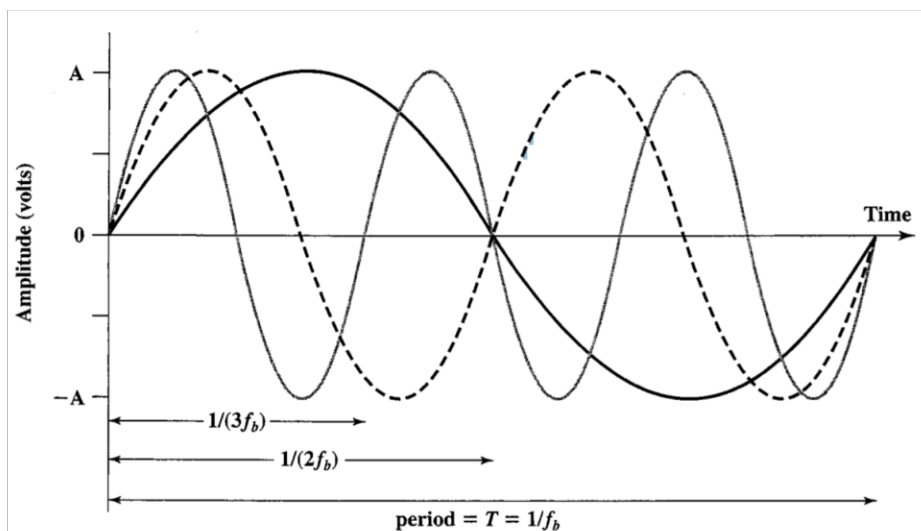


Figura 1.1. Tres subportadoras múltiplos de la frecuencia base f_b en el dominio del tiempo [4].

1.4.2.1 Ventajas y desventajas de OFDM

El esquema de transmisión OFDM tiene las siguientes ventajas clave:

- OFDM es una manera eficaz de lidiar con los problemas dados por el multitrayecto, para una dispersión de retardo dada, la complejidad de implementación del sistema de comunicación es menor que la de un sistema de una sola portadora con un ecualizador complejo.
- OFDM presenta la característica de flexibilidad permitiendo de esta manera el uso de técnicas avanzadas, como por ejemplo la diversidad en el transmisor y receptor con el fin de mejorar la eficiencia del sistema.
- OFDM hace posible las redes de frecuencia única, lo que es un segmento atractivo para aplicaciones de radiodifusión [3].
- Presenta inmunidad a canales con características de desvanecimiento selectivo en frecuencia dado que las subportadoras usadas solo experimentan el desvanecimiento plano.
- El solapamiento de portadoras permite que el espectro de frecuencias sea utilizado eficientemente.
- OFDM posee un tiempo de guarda el cual no ayuda a que el sistema no se vea afectado por ISI dado por los ambientes con múltiples trayectorias propios del canal.
- OFDM es computacionalmente eficiente gracias al uso de IFFT (*Inverse Fast Fourier Transform*) y FFT (*Fast Fourier Transform*) para la modulación y demodulación respectivamente.
- Brinda protección contra el ruido impulsivo, así también, contra la interferencia co-canal.

Por otro lado, OFDM también presenta algunos inconvenientes o desventajas.

- OFDM es más sensible al desplazamiento de frecuencia y también al ruido de fase, lo que genera errores de sincronización en el receptor [3,5].
- El transmisor de alta potencia requiere de linealización [5].
- El PAPR (*Peak to Average Power Ratio*) es alto.

1.4.3 NOMA (NON-ORTHOGONAL MULTIPLE ACCESS)

El acceso múltiple no ortogonal es una técnica que promete superar los esquemas de acceso convencionales en varios puntos, esta técnica de acceso usa SC (*Superposition Coding*) que tiene como objetivo el transmitir dos mensajes simultáneamente codificándolos en una sola señal para de esta manera poder compartir los recursos disponibles entre los usuarios y también utiliza SIC (*Successive Interference Cancellation*) para la detección multiusuario. La detección es procesada en el dominio de la potencia y puede ser establecida con la asignación de un valor de potencia adecuada, en este dominio NOMA utiliza SC para la transmisión y SIC para la recepción [6].

La capacidad de transmitir información haciendo uso del mismo recurso de frecuencia entre diferentes usuarios simultáneamente con diferentes niveles de potencia, como se logra observar en la Figura 1.2, representa un gran impacto en el valor de latencia durante una transmisión simultánea. Este valor decae significativamente y a consecuencia de esto se tiene un mejor uso del ancho de banda asignado y de esta manera mejorar la eficiencia espectral.

Si se compara con las generaciones anteriores de redes móviles que se han estado basando en el dominio de frecuencia, código y tiempo, como por ejemplo OFDMA (*Orthogonal Frequency-Division Multiple Access*) que es utilizado en LTE, es que esta técnica presenta un problema principal en donde su eficiencia espectral es baja cuando algunos recursos del ancho de banda se asignan a usuarios con CSI (*Channel State Information*) deficiente. Por otra parte, NOMA da la posibilidad de que cada usuario tenga acceso a todos los canales de la subportadora lo que mejora significativamente la eficiencia espectral [7].

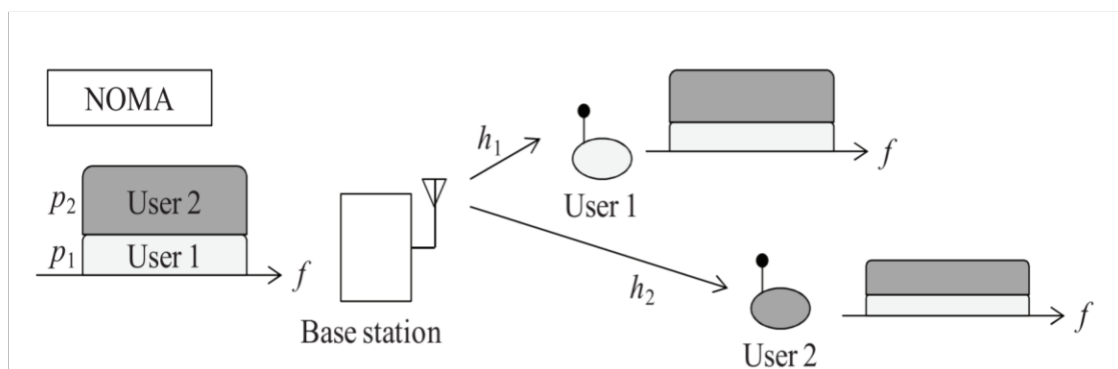


Figura 1.2. Técnica de acceso NOMA [8].

1.4.3.1 Beneficios de NOMA

Dado la alta demanda de usuarios conectados simultáneamente NOMA brinda beneficios en su aplicación en las redes de comunicaciones inalámbricas, algunas que se pueden detallar son:

- Eficiencia espectral alta como resultado de que NOMA permite que un bloque comparta recursos que pueden ser en tiempo, frecuencia o código, de esta manera todo el canal es aprovechado por los múltiples usuarios conectados simultáneamente.
- NOMA revisa cual de sus usuarios presenta peores condiciones en el canal y es al que asigna la mayor potencia, garantizando de esta manera que la información transmitida entre usuarios presente un buen rendimiento y equilibrio del sistema, este último nos permite mantener una equidad entre usuarios.
- Presenta baja latencia como consecuencia de la transmisión simultanea debido a que el usuario no debe esperar por un intervalo o *slot* de tiempo para transmitir la información.
- Buena compatibilidad debido a que NOMA puede complementarse con cualquier técnica OMA existente [9].

1.4.4 SINCRONIZACIÓN PARA NOMA-OFDM

Uno de los principales inconvenientes en comparación con los sistemas de portadora única es que el sistema OFDM es muy sensible a errores provocados por desfases de tiempo y frecuencia por lo que es recomendable añadir una fase de sincronización en la parte de recepción, para el caso frecuencial como para el temporal y de esta manera receptor de manera correcta la información que fue transmitida. Este trabajo se centra en el problema de sincronización de frecuencia, este inconveniente es generado por la diferencia existente entre la frecuencia del oscilador en el transmisor con relación a la frecuencia del oscilador en el receptor conocido tradicionalmente como *Carrier Frequency Offset* (CFO).

1.4.4.1 CFO y sus efectos

Como se mencionó anteriormente, el CFO puede darse como consecuencia de la discrepancia de frecuencia entre los osciladores del transmisor y el receptor obteniendo un desplazamiento de fase, además también puede darse por el efecto Doppler como resultado del movimiento de los usuarios y sus dispositivos o de la propia estación base. El *offset* de frecuencia dado genera una pérdida de ortogonalidad en las portadoras que

integran el símbolo OFDM, obteniendo así algunos efectos negativos en el rendimiento del sistema, uno de estos es la ICI (*Intercarrier Interference*) impidiendo que la información tenga una demodulación de forma correcta.

Otra de las consecuencias producidas por el CFO es que genera una rotación en la fase de la subportadora, de tal manera que esta desviación debe ser estimada y corregida para que exista una correcta recepción de la información transmitida, por último, el CFO provoca una atenuación de la señal enviada desde el receptor [10].

Por lo regular, al desplazamiento de frecuencia se lo puede expresar como f_{offset} y se lo puede obtener mediante la diferencia entre la frecuencia de la portadora del transmisor f_c y la frecuencia portadora del receptor f_c' .

$$f_{offset} = f_c - f_c' \quad (1.1)$$

Se expresa el CFO normalizado ε , como la relación entre el CFO y el espaciado de subportadoras Δ_f , dicho de otra manera $\varepsilon = \frac{f_{offset}}{\Delta_f}$. Igualmente, se expresa ε_i y ε_j como componentes del CFO normalizado, siendo la parte entera y fraccional, respectivamente.

1.4.4.2 Técnicas de estimación de CFO

Estimación de CFO en el dominio del tiempo utilizando el prefijo cíclico.

Si se tiene una sincronización perfecta en función del símbolo un CFO de ε genera una rotación de fase de $\frac{2\pi n \varepsilon}{N}$ en la señal recibida, en este punto se puede asumir que el efecto dado por el canal es despreciable, por tal motivo el CFO puede ser calculado haciendo uso del ángulo de fase dado por el producto entre el CP y su correspondiente al final del símbolo OFDM, como se expresa en la Ecuación 1.2.

$$\hat{\varepsilon} = \frac{1}{2\pi} \arg \left\{ \sum_{n=-N_G}^{-1} y^* [n] y [n + N] \right\} \quad (1.2)$$

Estimación de CFO en el dominio del tiempo utilizando símbolos de entrenamiento.

Con el fin de aumentar el rango de estimación del CFO, se busca disminuir la distancia existente entre los bloques de muestras para la correlación. Y esto es posible usando el preámbulo ya que este contiene símbolos repetitivos dentro de un periodo más corto. Se define D como el número de patrones repetitivos en el símbolo de entrenamiento o preámbulo de N muestras. Es decir, cada símbolo repetitivo contiene N/D muestras, de tal manera que es posible estimar el CFO mediante la Ecuación 1.3.

$$\hat{\varepsilon} = \frac{D}{2\pi} \arg \left\{ \sum_{n=0}^{\frac{N}{D}-1} y * [n] y [n + \frac{N}{D}] \right\} \quad (1.3)$$

Estimación de CFO en el dominio de la frecuencia utilizando símbolos de entrenamiento.

Si se tienen dos símbolos de entrenamiento iguales y estos son transmitidos uno después de otro, las señales con CFO de ε están relacionadas una con otra, considerando que se trabaja en el dominio de la frecuencia, tal como se muestra en la expresión de la Ecuación 1.4.

$$\hat{\varepsilon} = \frac{1}{2\pi} \arg \left\{ \sum_{k=0}^{N-1} [Y_1 * [k] Y_2 [k]] \right\} \quad (1.4)$$

2 METODOLOGÍA

En este capítulo se describe el código implementado para realizar las simulaciones del sistema NOMA-OFDM, el mismo que se ejecutará sobre el software de simulación MATLAB. El esquema consta de diferentes secciones: configuración de parámetros de simulación, transmisor NOMA-OFDM, canal Rician, ruido AWGN, sincronización con las diferentes técnicas de estimación CFO, receptor NOMA-OFDM, cálculo de errores, generación de archivo con resultados y generación de grafica. Así mismo, se detallará las funciones, variables y comandos usados para la creación del script principal.

El enfoque del presente trabajo es cuantitativo ya que requiere de la estimación del CFO que es una magnitud que afecta a la señal transmitida, de esta manera poder determinar el rendimiento de cada una de las técnicas presentadas en el capítulo 1, basándose en los errores cometidos en la transmisión y el valor de potencia requerido para obtener cierta cantidad de errores. Además de que se ha realizado una investigación que tiene un orden secuencial, partiendo desde el transmisor, pasando por el canal inalámbrico y terminando en el receptor en el que incluye el proceso de sincronización. El alcance es de tipo mixto, ya que combina tres tipos de alcance que son descriptivo, correlacional y explicativo. Descriptivo porque permite describir como afecta el CFO a la señal transmitida. Correlacional porque permite determinar la relación que existe entre dos variables, en este caso los datos generados en la etapa de transmisión y los datos recibidos por cada una de las capas, de esta manera obtener los errores presentes y de esta manera determinar el comportamiento del sistema en general. Y explicativo porque permite explicar por qué se da el CFO y las diferentes técnicas de estimación y compensación.

La Figura 2.1 presenta el diagrama de flujo de todo el sistema NOMA-OFDM implementado en MATLAB. La técnica de acceso NOMA permite que dos o más usuarios compartan el mismo recurso de red, por lo que se consideran dos usuarios denominados como capa1 y capa2, respectivamente. El código implementado consta de dos lazos, en el primero se realizan tantas iteraciones como elementos en el vector SNR_dB existan, mientras que en el segundo lazo hace referencia al número de repeticiones que el programa se ejecutara para un valor de SNR_dB . Es decir, para cada valor dentro de la variable SNR_dB se ejecutarán tantas veces como se establezca la variable $iter$ y de este último depende la precisión de los resultados.

Una vez codificados, modulados y multiplexados los datos de cada uno de los usuarios en el transmisor NOMA-OFDM, la señal resultante se verá afectada por los efectos del canal. En este punto se tiene dos señales, debido a que tenemos dos usuarios o capas como

están definidos en el presente trabajo. En la sección de recepción de datos, en primera instancia, las señales son afectadas por el CFO, por esta razón se realiza un proceso de inserción de este a cada una de las señales.

Posteriormente, se tiene la etapa de sincronización en donde se utiliza la técnica de estimación de CFO elegida en la sección de parámetros de simulación. Una vez estimado el valor de CFO, se procede a la etapa de compensación. En este punto ya se tiene las señales sincronizadas y pueden ser receptadas por cada una de las capas. Además, es importante mencionar que cada uno de los receptores realizan un proceso diferente para obtener los datos que le corresponden y en cada uno de estos se realiza el proceso de equalización.

Finalmente, se realiza el cálculo de BER, se guardan los datos necesarios para un posterior análisis y se genera las grafica de dicha simulación, en donde se compara el BER Vs. Eb/No. La Figura 2.1 representa específicamente el desarrollo del programa principal, por tal razón es importante mencionar que el programa posee varias funciones que permite el funcionamiento de todo el sistema NOMA-OFDM y estas serán explicadas en el presente capitulo.

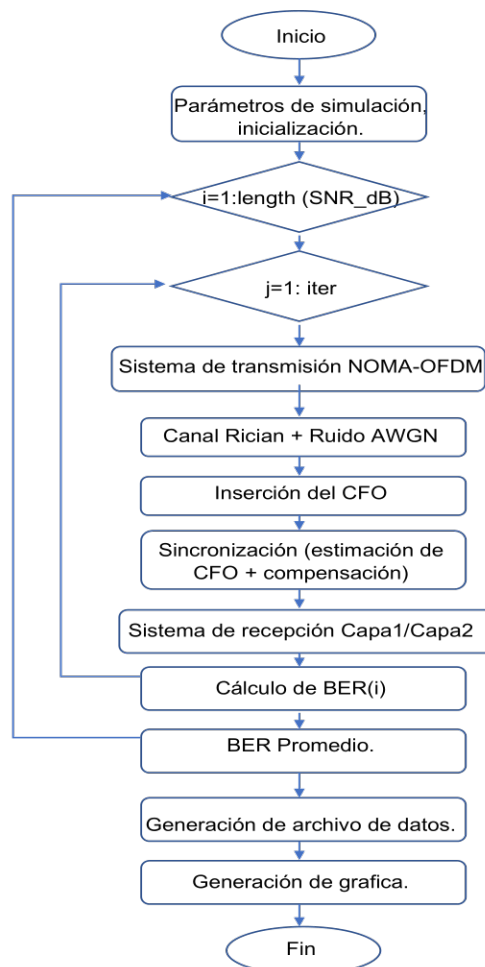


Figura 2.1. Diagrama de flujo del programa principal.

2.1 INICIALIZACIÓN

En primer lugar, se procede a definir el transmisor OFDM con la ayuda del *Communications Toolbox* presente en el software de simulación. En el segmento de Código 2.1 se muestra la configuración del transmisor OFDM bajo los parámetros dados por el estándar IEEE 802.11p. Es importante aclarar que el número de símbolos transmitidos son 6 debido a que el preámbulo cuenta como un símbolo más y este es agregado a la señal *txSig* correspondiente a los 5 símbolos transmitidos por las capas, por tal motivo la variable *NumSymbols* tiene el valor n_sym+1 , en donde n_sym es igual a 5. De igual forma, una vez creado el modulador, se define el demodulador en función del primero.

A continuación, se establece el número de bits para cada una de las capas, en función de su tasa de codificación, esquema de modulación y el número de símbolos que transmitirá. Finalmente se establece un valor de CFO, que consiste en la suma del CFO entero y CFO fraccional, tal como se presenta en el capítulo 1. Además de eso se elige la técnica de estimación que será usada en la simulación, en donde la variable *tec_CFO* puede tomar 4 valores: 0 para una sincronización perfecta en donde el CFO no afecta a la señal transmitida, 1 para usar la técnica de estimación de CFO en el dominio del tiempo utilizando el PC (prefijo cíclico), 2 para usar la técnica de estimación de CFO en el dominio del tiempo utilizando el preámbulo y por último 3 para usar la técnica de estimación de CFO utilizando el preámbulo.

```
OFDM_mod=comm.OFDMModulator('FFTLength',64, ...
                             'PilotInputPort',true, ...
                             'PilotCarrierIndices',[12;26;40; 54], ...
                             'InsertDCNull',true, ...
                             'CyclicPrefixLength',16, ...
                             'NumSymbols',n_sym+1);
OFDM_demod = comm.OFDMDemodulator(OFDM_mod);

num_bits_c1=48*k1*m1*n_sym; %Numero de bits para la capa 1
num_bits_c2=48*k2*m2*n_sym; %Numero de bits para la capa 2

% Datos para el CFO
IFO = 0; %CFO entero
FFO = 0.21; %CFO fraccional
CFO = IFO + FFO;
tec_CFO = 0; %Técnica de estimación
           %[0]: (Sincronización Perfecta)
           %[1]: (Est CFO dom-tiempo PC)
           %[2]: (Est CFO dom-tiempo Pream)
           %[3]: (Est CFO dom-freq Pream)
```

Segmento de Código 2.1. Configuración del transmisor OFDM.

2.2 TRANSMISOR OFDM-NOMA

El transmisor será el encargado de generar los bits aleatorios que serán procesados en cada iteración y estos formarán parte de la señal OFDM que será transmitida. En la Figura 2.2 se muestra el diagrama de bloques de los procesos que se llevan a cabo en el transmisor NOMA-OFDM.

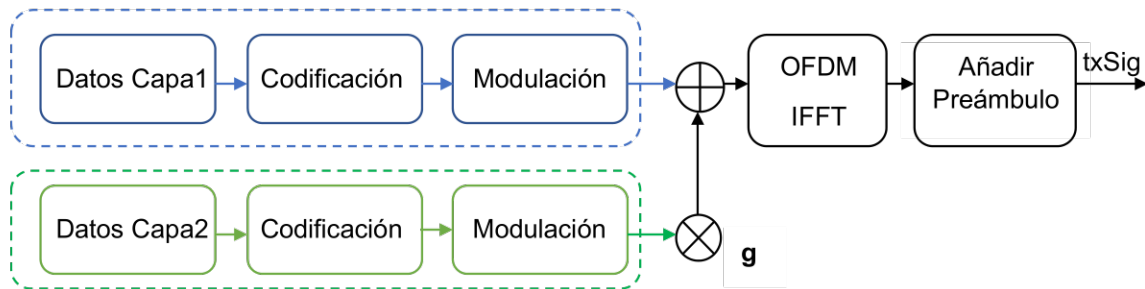


Figura 2.2. Diagrama de bloques del transmisor NOMA-OFDM.

El establecimiento de parámetros que serán utilizados para la simulación del sistema de transmisión se configura de forma independiente para cada una de las capas como muestra el segmento de Código 2.1. Además, en este mismo segmento de Código se presentan los parámetros para generar la señal OFDM que será transmitida, en los comentarios es posible identificar los posibles parámetros que pueden ser establecidos como también el significado de las variables dentro de la simulación.

```

%% Parámetros de simulación.

iter    = 100;           %número de iteraciones simulación
M1      = 16;           %CAPA 1 --- 64->64QAM,16->16QAM,4->QPSK
M2      = 16;           %CAPA 2 --- 64->64QAM,16->16QAM,4->QPSK
k1      = 1/2;          % tasa de codificación 1/2, 2/3, 3/4
k2      = 1/2;          % tasa de codificación 1/2, 2/3, 3/4
fac_dB  = -6;           %factor de inyección en dB -6
EbNo_dB=0:1:30;        %intervalo para simulación
PC      = 1/4;          %tamaño del prefijo cíclico

n_sym   =5;             %número de símbolos OFDM por paquete
Nfft    =64;            %tamaño fft
data_sub =48;           %número subportadoras de datos
pilot_sub =4;           %número subportadoras piloto

```

Segmento de Código 2.2. Parámetros utilizados para la simulación de sistema NOMA-OFDM.

La sección de transmisor se representa bajo la función implementada *Tx_NOMA_OFDM*, en donde los variables de entrada son las descritas anteriormente. A continuación, se presenta el desarrollo de la función en MATLAB.

2.2.1 GENERACIÓN DE BITS ALEATORIOS

La función *Tx_NOMA_OFDM* en primer lugar genera los bits de información que serán transmitidos. Esto lo hace mediante el comando `randi`, el mismo que brinda la posibilidad de generar una matriz de valores aleatorios entre 0 y 1. La dimensión de esta depende del número de bits requerido por cada capa y este valor se encuentran en las variables `num_bits_c1` y `num_bits_c2` pertenecientes a la capa 1 y capa 2, respectivamente. Este valor corresponde al número de filas, ya que la variable `data_bit` solamente posee una fila. En el segmento de Código 2.3 se presenta lo antes mencionado.

```
data_bit1 = randi([0 1],num_bits_c1,1); %generación bits usuario 1
data_bit2 = randi([0 1],num_bits_c2,1); %generación bits usuario 2
```

Segmento de Código 2.3. Generación de bits aleatorios.

2.2.2 CODIFICACIÓN FEC

La codificación se realiza bajo la técnica de procesamiento de señal digital FEC (*Forward Error Correction*, en español: Corrección de errores hacia adelante), con el objetivo de incrementar la confiabilidad de los datos. Se empieza por definir las características del codificador y decodificador, esto se observa en las dos primeras líneas del segmento de código 2.4, en donde se definen dos codificadores dado que se tiene dos capas o usuarios y ambos operan de manera independiente. Posteriormente, se procede a la codificación de la información de cada una de las capas.

```
%Codificador y decodificador.
[convEncoder1, vitDecoder1]=conf_cod1(k1,traceBack);
[convEncoder2, vitDecoder2]=conf_cod2(k2,traceBack);
dataEnc1 = convEncoder1(data_bit1); %codificación FEC
dataEnc2 = convEncoder2(data_bit2); %codificación FEC
```

Segmento de Código 2.4. Codificación FEC para cada una de las capas.

2.2.3 MODULACIÓN

Para el proceso de modulación se usa el comando *qammod*, presente en el *Communication Toolbox*. Cada señal es modulada de acuerdo con el orden de modulación establecido en las variables *M1* y *M2*, teniendo de esta manera 3 posibles ordenes de modulación que

son: 4 para QPSK, 16 para 16QAM y 64 para 64QAM, este es el primer paso para obtener el símbolo OFDM. Las variables de entrada que acepta el comando *qammod* son la señal por modularse, en este caso *dataEnc1* y *dataEnc2* correspondiente a cada usuario, el orden de modulación, también se especifica mediante *InputType* que los elementos a modularse son bits y mediante el comando *UnitAveragePower* se configura la potencia de la señal de salida a 1W. Como resultado se tiene 240 subportadoras de datos que corresponden a los 5 símbolos que transmiten cada una de las capas y estas son almacenadas en la variable *data_sym*. Esto se presenta en el segmento de Código 2.5.

```
data_sym1 = qammod(dataEnc1,M1,'InputType','bit', ...
                  'UnitAveragePower',true);
data_sym2 = qammod(dataEnc2,M2,'InputType','bit', ...
                  'UnitAveragePower',true);
```

Segmento de Código 2.5. Modulación de las capas.

2.2.4 SUMA DE SEÑALES E INSERCIÓN DEL FACTOR DE INYECCIÓN

En esta fase se procede a sumar las señales, en primera instancia se convierte a matriz los datos modulados, teniendo como resultados dos matrices de 48x5. Los mismos que corresponden a los 5 símbolos de datos con 48 subportadoras de información de cada una de las capas. Posteriormente, se procede a multiplicar la matriz de la capa 2 por un factor de inyección y al resultado de esta se le suma la matriz de la capa 1, obteniendo así una única señal que se guarda en la variable *data_sym_tot*. Esto se encuentra en el segmento de Código 2.6.

```
%convierte en matriz
data_sym_mat1=reshape(data_sym1,data_sub,n_sym); %Capa 1
data_sym_mat2=reshape(data_sym2,data_sub,n_sym); %Capa 2
data_sym_tot=data_sym_mat1+data_sym_mat2*factor; %suma 2 usuarios
```

Segmento de Código 2.6. Suma de señales e inserción del factor de inyección.

2.2.5 MODULACIÓN OFDM

En la fase de inicialización se creó el objeto *OFDM_mod* el mismo que está configurado bajo los parámetros del estándar IEEE 802.11p presente en los sistemas de acceso inalámbrico en entornos vehiculares. En el segmento de Código 2.7 se presenta la aplicación del objeto para la modulación y las respectivas variables con la que trabaja. Para este fin se utiliza el comando *step*, en donde se ordena que el modulador OFDM definido

anteriormente, se aplique sobre la señal de datos *data_sym_tot* y los pilotos establecidos en etapas anteriores.

```
txSig = step(OFDM_mod,data_sym_tot,pilotos); %se aplica OFDM
```

Segmento de Código 2.7. Modulación OFDM.

2.2.6 GENERACIÓN DEL PREAMBULO E INSERCIÓN A LA SEÑAL A TRANSMITIRSE

El preámbulo es uno de los símbolos más importantes a transmitirse ya que es muy útil para el proceso de sincronización y debe ser posicionado al inicio de la señal OFDM generada en la sección anterior. La función *get_pream* es la que se encarga de entregar el preámbulo listo para ser insertado en la señal OFDM, es decir, la señal ya pasó previamente por una *IFFT* y de esta manera es posible añadirla y formar una nueva señal que será transmitida, obteniendo así la señal de 6 símbolos, 5 símbolos de información y 1 símbolo de preámbulo. El segmento de Código 2.8 presenta el llamado de la función, obteniendo la longitud del preámbulo y finalmente se concatenan las señales obteniendo de esta manera la señal final que será enviada por el canal.

```
%*****Preambulo*****
[pream]=get_pream(Nfft); %se obtiene el preambulo
preambulo=pream;
length_pream=length(preambulo); %guardamos la longitud de
%la variable.

%*****Señal transmitida*****
txSig=[preambulo; txSig]; % Señal transmitida
```

Segmento de Código 2.8. Creación e inserción del preámbulo en la señal a transmitirse.

La función *get_pream* acepta como único parámetro de entrada el valor total de las subportadoras, en este trabajo este valor se encuentra almacenado en la variable *Nfft* con valor 64 y como salida nos retorna el preámbulo. Este se forma partiendo de una PN (*pseudonoise sequence*, en español: secuencia de pseudo-ruido), posteriormente, se lo multiplica por $2\sqrt{2}$ buscando mantener la energía de la señal constante. Considerando que el símbolo tiene una estructura repetitiva se elige utilizar un tiempo de guarda con valores de cero. Finalmente, se concatena el periodo de guarda con el símbolo formado por la PN y se duplica este último con el signo cambiado con el fin de mejorar la eficiencia de cada una de las técnicas de estimación que se utilizan, debido a que poseen dos valores de referencia [11]. La Figura 2.3 presenta como está formado el preámbulo.

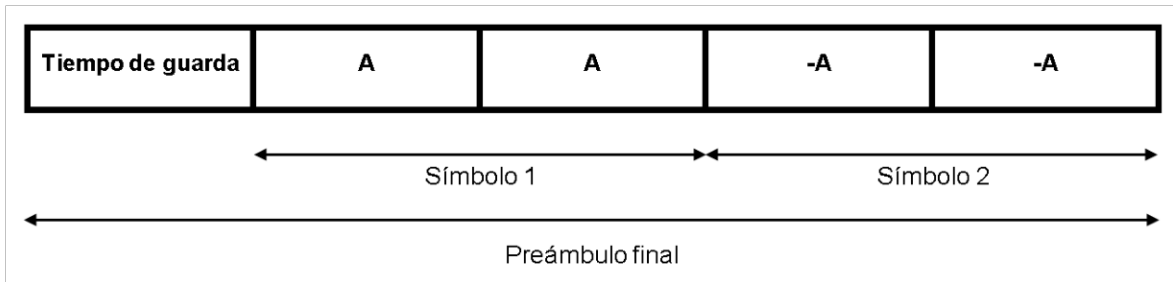


Figura 2.3. Estructura final del preámbulo [11].

El código de la función *get_pream* se muestra en el Segmento de Código 2.9.

```
function [pream] = get_pream (Nfft)

%Secuencia de PseudoRuido
PN_seq = [1 1 -1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 -1 -1 -1 -1 -1 ...
          -1 1 -1 -1 -1 1 0 1 -1 -1 -1 1 1 1 -1 1 -1 -1 -1 1 -1...
          -1 -1 -1 1 1 1 -1 1 1 -1 1 1];

PN_seq = PN_seq*2*sqrt(2);
PN_seq(2:2:end) = 0;      %Tx de ceros en frecuencia impares

%Mapeo de subportadoras.
PN_seq_ord = zeros(Nfft,1);
PN_seq_ord(39:64) = PN_seq(1:26);
PN_seq_ord(1) = PN_seq(27);
PN_seq_ord(2:27) = PN_seq(28:53);

prel = ifft(PN_seq_ord, Nfft);      %Dominio del tiempo

T_guarda = 0*prel(33:64);          %Tiempo de guarda
pream = [T_guarda; prel; -prel];    %Preámbulo de forma [A A -A -A]

end
```

Segmento de Código 2.9. Función *get_pream* [11].

2.3 CANAL RICIAN

Una vez que el transmisor genera la señal que será enviada por el medio de transmisión, el siguiente paso es definir el canal. En este trabajo se utiliza el canal tipo Rician el mismo que está definido bajo el comando *comm.RicianChannel* presente en el *Communication Toolbox*. En el segmento de Código 2.10 se muestra el código de la función *Channel*, el mismo que devuelve el objeto canal y la variable *sampIdx* que indica las posiciones de los rayos en el vector. Para la configuración del objeto canal se procede a ingresar los parámetros en el comando *comm.RicianChannel*, el primero es el factor K que contiene información de la potencia de la señal con línea de vista con respecto a las otras sin línea de vista, la frecuencia de muestreo dado por la variable *sRate*, *Delay* que es el retardo de

cada rayo, la variable *Pot* que da la información sobre la potencia en dB de cada rayo, *fd* que corresponde a la frecuencia Doppler que en este trabajo tiene el valor de cero debido a que los usuarios no se encuentran en movimiento y por último se activa el puerto de salida del *path* de ganancias.

```
function [ canal,sampIdx ] = Channel( )
%Pot Potencia en dB de cada rayo.
%Delay Retardo de cada rayo.

Pot = [0 -14 -17];
Delay = [0 100 200]*1e-9;

K=10; %10

% Establecer la frecuencia Doppler máxima
fd = 0;
% Definir la frecuencia de muestreo según el AB del estándar
sRate = 10*1e6;
% Crear el canal
canal = comm.RicianChannel (...
    'KFactor', K, ...
    'SampleRate', sRate, ...
    'PathDelays', Delay, ...
    'AveragePathGains', Pot, ...
    'MaximumDopplerShift', fd, ...
    'PathGainsOutputPort', true ...
);

% Calcular sampIdx
sampIdx = round(Delay/(1/sRate)) + 1;

end
```

Segmento de Código 2.10. Función Channel para obtener el objeto de canal tipo Rician.

Ahora que la función Channel está creada, se procede a definir dos canales en el programa principal, esto se realiza en la sección de inicialización, en el segmento de Código 2.11. se muestra cómo se generan los canales. La razón por la que se generan dos canales es por la presencia de dos usuarios y cada uno de estos posee su receptor independiente, de tal manera que la señal que reciba el usuario 1 se verá afectada de forma diferente por el canal que la señal recibida por el usuario 2.

```
% Definimos el canal
[canal_fading1,sampIdx1]=Channel(); %genera canal con función
[canal_fading2,sampIdx2]=Channel(); %genera canal con función
```

Segmento de Código 2.11. Se generan los canales con la función creada.

A continuación, en el lazo de iteraciones se debe reiniciar cana uno de los canales, luego, con el comando *step* se le aplica el desvanecimiento a la señal enviada por el transmisor, obteniendo de esta manera la señal con desvanecimiento, definida en el programa como *Y1* y *Y2*, para el usuario 1 y 2, respectivamente. Así también, el *path* de ganancias. Finalmente, a la señal anterior se le aplica ruido con el comando *awgn* y como resultado final se obtiene dos señales, *txSig1* que es receptada por el usuario 1 y *txSig2* con el usuario 2. La aplicación de los canales se puede ver implementado en el segmento de Código 2.12.

```
%% *****CANAL RICIAN*****

%-----Usuario1-----
reset(canal_fading1);           %Resetear el objeto canal
[y1,pathgains1] = step(canal_fading1,txSig); %Filtrar la
                                     %señal con canal
txSig1=awgn(y1,SNR_dB(i),'measured'); % se aplica ruido

%----Usuario2-----
reset(canal_fading2);           %Resetear el objeto canal
[y2,pathgains2] = step(canal_fading2,txSig); %Filtrar la
                                     %Señal con canal
txSig2=awgn(y2,SNR_dB(i),'measured'); % se aplica ruido
```

Segmento de Código 2.12. Canal Rician más ruido AWNG.

2.4 RECEPCIÓN

En la Figura 2.4 se muestra el diagrama de bloques del sistema de recepción NOMA-OFDM, al tener dos usuarios o capas, la recepción de la información se la efectúa de manera independiente, es decir, cada usuario tiene su propio receptor ya que uno es más complejo que el otro para recuperar la señal. Se puede observar que en cada usuario algunos de los procesos son similares; por ejemplo, ambos realizan el proceso de sincronización, extracción del preámbulo, demodulación OFDM y ecualización. A partir de este punto el proceso para recuperar la señal de cada usuario es diferente, en el primero caso solo se recupera la señal de la capa 1, mientras que para el usuario 2 se debe primero recuperar la señal del usuario 1 para que después esta sea restada de la señal total recibida y de esta manera recuperar la señal de la capa 2. En esta sección se explicará cada uno de los bloques presentados en la Figura 2.4.

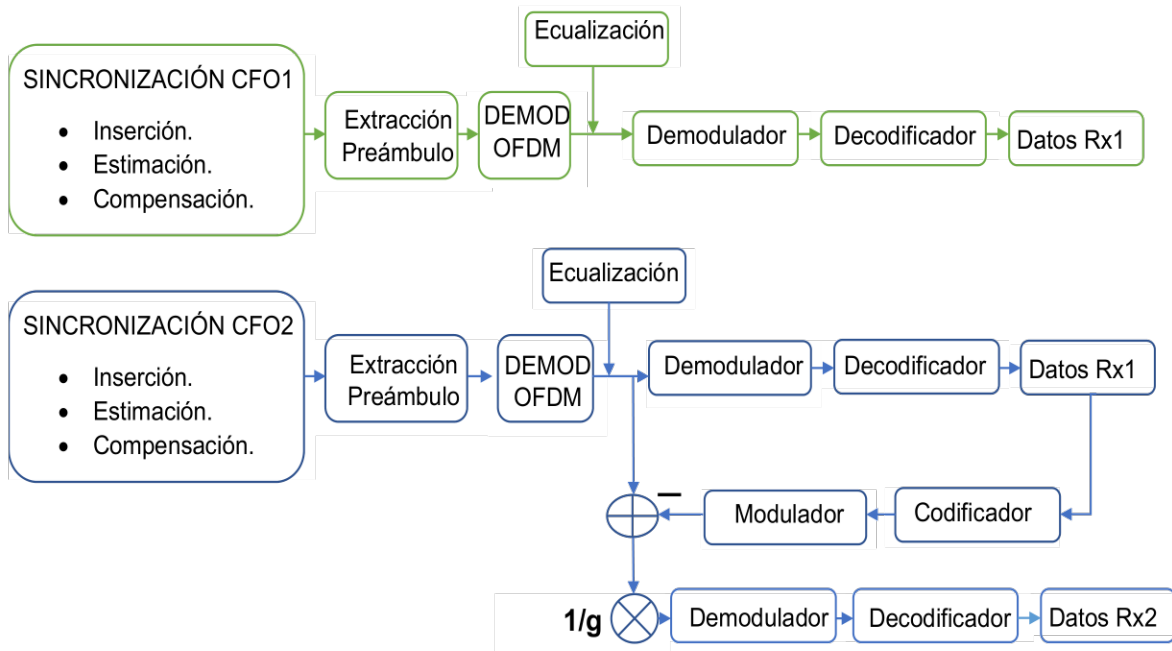


Figura 2.4. Diagrama de bloques del sistema de recepción NOMA-OFDM.

2.4.1 SINCRONIZACIÓN CFO

En esta etapa del receptor se procede a realizar el proceso de sincronización que está formado por tres secciones que son: el proceso de inserción, estimación y finalmente la compensación del CFO. En el segmento de Código 2.13 se presentan el llamado de cada una de las funciones para las diferentes secciones de la etapa de sincronización en el programa principal. Posteriormente, se detallarán las funciones, variables y comandos utilizados en cada una de estas secciones.

```

%% *****[SINCRONIZACION]*****

%*****[INSERCIÓN DEL CFO]*****
sym_CF01=ins_CF0(rxSigu1, CF0, Nfft);
sym_CF02=ins_CF0(rxSigu2, CF0, Nfft);

%*****Estimación*****
CF0_est1=tec_est(tec_CF0, sym_CF01, Nfft, PC, length_pream);
CF0_est2=tec_est(tec_CF0, sym_CF02, Nfft, PC, length_pream);

%*****[Compensación]*****
Sym_CF0_comp1=comp_CF0(sym_CF01, CF0_est1, Nfft);
Sym_CF0_comp2=comp_CF0(sym_CF02, CF0_est2, Nfft);

```

Segmento de Código 2.13. Etapa de sincronización con funciones implementadas para cada una de las secciones.

2.4.1.1 Inserción del CFO

En esta sección se añade el efecto de CFO a la señal OFDM recibida, esto se realiza al multiplicar la señal OFDM por el factor $e^{j2\pi k\epsilon/N}$, en donde ϵ es el valor del CFO que fue establecido en la sección de inicialización y k es un vector de longitud igual a la señal recibida. Para realizar la inserción se define una función llamada *ins_CFO*, misma que acepta los siguientes parámetros de entrada: *rxSigu* que corresponde a la señal OFDM recibida, el valor del CFO y por último el valor de las subportadoras totales, el mismo que sirve para la creación del vector k . El segmento de Código 2.13 presenta la implementación de la función.

```
function [sym_CFO]=ins_CFO(rxSigu, CFO, Nfft)
%insercion del CFO
    nn=0:length(rxSigu)-1;
    nn=reshape(nn,[length(rxSigu),1]);
    sym_CFO = rxSigu.*exp(1i*2*pi*nn*CFO/Nfft);

end
```

Segmento de Código 2.13. Implementación de la función *ins_CFO*.

2.4.1.2 Estimación del CFO

Para realizar la estimación del CFO se utilizan tres técnicas, estas son: estimación del CFO en el dominio del tiempo utilizando el prefijo cíclico, estimación en el dominio del tiempo utilizando el preámbulo o también conocido como símbolos de entrenamiento y por último estimación en el dominio de la frecuencia utilizando el preámbulo. En el segmento de Código 2.14 se muestra la implementación de una función que tiene como objetivo el de elegir la técnica de estimación establecida en el proceso de inicialización y pasarle a dicha técnica los variables que esta necesita para ejecutarse.

```
function CFO_est=tec_est(tec_CFO,sym_CFO,NsubT,PC,length_pream)
%Estimacion del CFO
switch tec_CFO
    case 0 %Sin estimación
        CFO_est=0;
    case 1 %Estimación CFO dominio tiempo utilizando PC
        CFO_est=CFO_tiempo_PC(sym_CFO, NsubT, PC);
    case 2 %Estimación CFO dominio tiempo utilizando preámbulo
        CFO_est=CFO_tiempo_pream(sym_CFO, length_pream);
    case 3 %Estimación CFO dominio frecuenc. utilizando preámbulo.
        CFO_est=CFO_frecuencia_pream(sym_CFO,NsubT, length_pream);
end
```

Segmento de Código. 2.14. Función que permite elegir la técnica de estimación a ejecutarse.

- **CFO_tiempo_PC**

En el caso 1 se tiene a la técnica de estimación en el dominio del tiempo utilizando el prefijo cíclico, esta técnica estima este valor a partir del cálculo del ángulo de fase que corresponde al producto entre el prefijo cíclico y su correspondiente al final del símbolo OFDM. El *delay* establecido nos indica la posición del prefijo cíclico dentro de la señal recibida, de tal manera que al tener 5 símbolos se debe realizar el cálculo mencionado anteriormente en cada uno de los símbolos y al final se realiza un promedio de todos los valores CFO calculados. La estimación se la realiza con ayuda de la Ecuación 1.2 y la implementación se la presenta en el segmento de Código 2.15.

```
function CFO_est=CFO_tiempo_PC(sym_CFO,NsubT,PC)
%Estimación CFO dominio tiempo utilizando PC

% Simbolo 1
Ng = NsubT*PC;           %Tamaño del prefijo cíclico
delay = 160;             %Retraso simbolo 1
nn = delay+(1:Ng);
tmp = conj(sym_CFO(nn)).*sym_CFO(nn+NsubT);
CFO_est1 = 1/(2*pi)*angle(sum(tmp)); %Ecuacion 1.2

% Simbolo 2
delay = 240;             %Retraso símbolo 2
nn = delay+(1:Ng);
tmp = conj(sym_CFO(nn)).*sym_CFO(nn+NsubT);
CFO_est2 = 1/(2*pi)*angle(sum(tmp));

% Simbolo 3
delay = 320;             %Retraso símbolo 3
nn = delay+(1:Ng);
tmp = conj(sym_CFO(nn)).*sym_CFO(nn+NsubT);
CFO_est3 = 1/(2*pi)*angle(sum(tmp));

% Simbolo 4
delay = 400;             %Retraso símbolo 4
nn = delay+(1:Ng);
tmp = conj(sym_CFO(nn)).*sym_CFO(nn+NsubT);
CFO_est4 = 1/(2*pi)*angle(sum(tmp));

% Simbolo 5
delay = 480;             %Retraso símbolo 5
nn = delay+(1:Ng);
tmp = conj(sym_CFO(nn)).*sym_CFO(nn+NsubT);
CFO_est5 = 1/(2*pi)*angle(sum(tmp));

CFO_est=(CFO_est1+CFO_est2+CFO_est3+CFO_est4+CFO_est5)/5;

end
```

Segmento de Código 2.15. Función *CFO_tiempo_PC*

- **CFO_tiempo_pream.**

En el caso 2 se presenta la técnica de estimación en el dominio del tiempo utilizando el preámbulo, esta es similar a la anterior, pero disminuye la distancia entre los bloques de muestras para la correlación buscando de esta manera incrementar el rango de estimación. La Figura 2.3 indica como está formado el preámbulo y se observan dos símbolos idénticos positivos y dos símbolos idénticos negativos, es decir el preámbulo tiene la siguiente composición: [A A -A -A], en base a esta información se define la variable D misma que indica el número de símbolos idénticos en el preámbulo. La estimación se realiza en función de la Ecuación 1.3 y la implementación de la función se la presenta en el segmento de Código 2.16.

Al igual que en la anterior función, con el fin de reducir los errores se calculan 2 valores de CFO y posteriormente se realiza su promedio. El primer calculo corresponde a los segmentos de [A A] y el segundo a los segmentos de [-A -A]. El *delay* tiene valor de 32 e indica la posición en la que inicia el primer símbolo idéntico y este se correlaciona con su correspondiente, mientras que para encontrar el primer símbolo negativo idéntico se tiene que esperar 3 *delay*, por tal motivo la variable *delay1* tiene este valor y este símbolo estaría relacionado a su correspondiente negativo.

```
function CFO_est=CFO_tiempo_pream(sym_CFO, length_pream)
%Estimación CFO dominio tiempo utilizando preámbulo

%[A A]
N = 2*length_pream/5;           %Longitud del patron repetitivo = 64
D = 2;                          %Numero de simbolos identicos.
delay = 32;                      %retraso
nn = delay+(1:32);
tmp = conj(sym_CFO(nn)).*sym_CFO(nn+(N/D));
CFO_est1 = D/(2*pi)*angle(sum(tmp));

%[-A -A]
delay1=3*delay;
nn1 = delay1+(1:32);
tmp = conj(sym_CFO(nn1)).*sym_CFO(nn1+(N/D));
CFO_est2 = D/(2*pi)*angle(sum(tmp));

CFO_est=(CFO_est1+CFO_est2)/2;

end
```

Segmento de Código 2.16. Función *CFO_tiempo_PC*.

- **CFO_frecuencia_pream**

El último caso pertenece a la estimación del CFO en el dominio de la frecuencia utilizando el preámbulo, en este proceso se usa la relación que existe entre los dos símbolos de entrenamiento consecutivos, ya que como se mencionó en el caso anterior estos son idénticos. Se trabaja en el dominio de la frecuencia por lo que cada símbolo debe ser pasado a este dominio para luego estimar el valor del CFO basándose en la Ecuación 1.4.

En esta función se procede a extraer el preámbulo de la señal recibida para trabajar con este, una vez que se tiene el preámbulo guardado en la variable *preamt* se extrae el tiempo de guarda para de esta manera obtener los dos símbolos idénticos. La estructura del preámbulo es [A A -A -A], por lo que la segunda mitad del preámbulo debe ser multiplicada por menos uno para que ambos símbolos sean idénticos. La implementación de la función se muestra en el segmento de Código 2.17.

```
function CFO_est=CFO_frecuencia_pream(sym_CFO, NsubT,length_pream)

Ng = NsubT;
delay = 32;           %Retraso
Sym=reshape(sym_CFO, [1,length(sym_CFO)]);
preamt=Sym(1:length_pream);
preamt=preamt(delay+1:end);
Y = zeros(2,NsubT); %Inicializacion de la variable Y
Y(1,:) = fft(preamt(1:Ng), NsubT); %Primer simbolo OFDM
Y(2,:) = fft(-1.*(preamt(Ng+1:Ng*2)), NsubT); %Segundo simbolo
OFDM

tmp = conj(Y(1,:)).*Y(2,:);
CFO_est = (1/(2*pi))*angle(sum(tmp));

end
```

Segmento de Código 2.17. Función CFO_frecuencia_pream.

2.4.1.3 Compensación del CFO

Una vez que el valor del CFO haya sido estimado por cualquiera de las técnicas descritas, se procede a compensar el efecto generado por el mismo, que en este caso resulta un desfase de frecuencia. De tal manera, se realizan las operaciones contrarias a la efectuada en la sección de inserción del CFO. Dicho de otra manera, para la compensación se divide la señal por el factor $e^{j2\pi k\varepsilon/N}$, en donde ε es igual al CFO_est.

Este proceso se realiza por la función implementada llamada *comp_CFO*. Esta función admite las siguientes variables como parámetros de entrada: *sym_CFO* que corresponde a la señal afectada por el CFO, *CFO_est* que guarda el valor del CFO estimado y por último el valor de las subportadoras de datos. La función nos retorna la variable *Sym_CFO_comp*, misma que está lista para pasar al proceso de demodulación OFDM. La implementación de la función se presenta en el segmento de Código 2.18.

```
function [Sym_CFO_comp]=comp_CFO(sym_CFO, CFO_est, Nfft )

%Compensacion de CFO
nn=0:length(sym_CFO)-1;
nn=reshape(nn, [length(sym_CFO),1]);
Sym_CFO_comp=sym_CFO./exp(1i*2*pi*CFO_est*nn/Nfft);

end
```

Segmento de Código 2.18. Función de compensación del CFO.

2.4.2 RECEPTOR DE CADA CAPA

Como ya se mencionó anteriormente, cada una de las capas realiza un proceso diferente para obtener la información que se les envió desde el transmisor, por tal motivo se crearon dos funciones que trabajarán con cada una de las capas. Dentro de cada una de las funciones también se realiza el proceso de ecualización, mismo que permite combatir el desvanecimiento de cada subportadora. Para este fin se utiliza el ecualizador de una etapa (*one-tap equalization*) y este divide la señal recibida para la respuesta del canal, ambas deben estar en el dominio de la frecuencia. En el segmento de Código 2.19 se presentan el llamado de cada función en el programa principal y los parámetros de entrada que acepta cada una.

```
%*****[ Receptor -- usuario 1]*****
rx_bit_dec1=Rx_User1(rxSig1, length_pream, OFDM_demod, n_sym,...
    pathgains1, sampIdx1, data_sub, M1, vitDecoder1);

%*****[ Receptor -- usuario 2]*****
rx_bit_dec2=Rx_User2(rxSig2, length_pream, OFDM_demod, ...
    n_sym, pathgains2,sampIdx2, data_sub, ...
    k1, traceBack, M1, M2, factor, vitDecoder2);
```

Segmento de Código 2.19. Función receptor de cada una de las capas llamadas desde el programa principal.

2.4.2.1 Receptor Capa 1

La función *Rx_User1* permite recuperar los datos enviados por la capa 1, este receptor solamente procesa la señal que le corresponde. Inicialmente se retira el preámbulo ya que el objeto *OFDM_demod* tiene definido 5 símbolos transmitidos. Según el diagrama de bloques planteado en la figura 2.4, el siguiente paso es la demodulación OFDM en donde se retira el prefijo cíclico y las subportadoras piloto, es decir, se realiza un proceso inverso al modulador OFDM, para este fin se utiliza el comando *step* al que se pasa el objeto *OFDM_demod* y la señal a ser demodulada. El siguiente paso consta de la ecualización para combatir los efectos del canal y finalmente los dos últimos pasos de demodulación con el respectivo orden de modulación realizado en el transmisor y decodificación con la misma tasa implementada en el transmisor. El final de todo este proceso se obtiene la señal de la capa 1. Esto se lo presenta en el segmento de Código 2.20.

```
.function [rx_bit_decl]=Rx_User1(rxSig1, length_pream, ...
    OFDM_demod, n_sym, pathgains1, sampIdx1, data_sub,...
    M1, vitDecoder1)
%Receptor del usuario 1.

rxSig1=rxSig1((length_pream+1):end); %extraemos el preambulo
rx_data_mat1= step(OFDM_demod,rxSig1); %proceso inverso OFDM
rx_data_eq1=one_tap(rx_data_mat1, n_sym, pathgains1, sampIdx1);
rx_data1=reshape(rx_data_eq1,n_sym*data_sub,1); % se coonvierte a
vector

rx_bit1 = qamdemod(rx_data1,M1,'OutputType','bit', ...
    'UnitAveragePower',true);
rx_bit_decl = vitDecoder1(rx_bit1); %decodificacion FEC

end
```

Segmento de Código 2.20. Función del receptor del usuario 1.

2.4.2.2 Receptor Capa 2

La función *Rx_User2* tiene el mismo funcionamiento que la función del receptor de la capa 1 hasta obtener la señal del usuario 1, una vez que la señal ha sido recuperada la codifica con la misma tasa de codificación y posteriormente la modula con el mismo orden que en la transmisión. La señal resultante se resta de la señal total recibida y luego se divide para el factor de inyección. En este punto ya se tiene la señal de la capa 2, ahora esta debe ser demoduladas y decodificada.

La figura 2.5 presenta el diagrama de bloques del receptor del usuario 2. Esta figura ayudara al mejor entendimiento de cómo funciona el receptor.

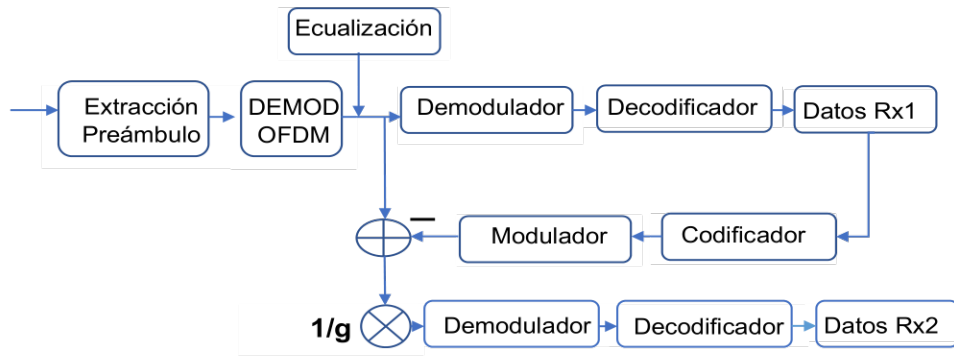


Figura 2.5. Proceso para obtener la señal de la capa 2.

El segmento de Código 2.21 presenta la implementación de la función.

```
function [rx_bit_dec2]=Rx_User2(rxSig2, length_pream, ...
    OFDM_demod, n_sym, pathgains2, sampIdx2,...
    data_sub, k1, traceBack, M1, M2, factor, ...
    vitDecoder2 )
%Receptor del usuario 2

%*****Primero obtenemos la señal de la capa 1*****
rxSig2=rxSig2((length_pream+1):end); %extraemos el preambulo
rx_data_mat2= step(OFDM_demod,rxSig2); %proceso inverso OFDM
rx_data_eq2=one_tap(rx_data_mat2, n_sym, pathgains2, sampIdx2);
rx_data2=reshape(rx_data_eq2,n_sym*data_sub,1);

[convEncoder1,vitDecoder1]=conf_cod1(k1,traceBack);
rx_bit1 = qamdemod(rx_data2,M1,'OutputType','bit', ...
    'UnitAveragePower',true);

%Se obtiene la señal de la capa 1.
rx_bit_dec1 = vitDecoder1(rx_bit1); %decodificacion FEC
%*****

[convEncoder1,vitDecoder1]=conf_cod1(k1,traceBack);

%codificacion FEC
rx_bit_cod1 = convEncoder1([rx_bit_dec1(traceBack+1:end); ...
    zeros(traceBack,1)]);
sig_tx_1 = qammod(rx_bit_cod1,M1,'InputType','bit', ...
    'UnitAveragePower',true);

%Resta de señales
sig_tx_2=(rx_data2-sig_tx_1)./factor;
rx_bit2 = qamdemod(sig_tx_2,M2,'OutputType','bit', ...
    'UnitAveragePower',true);
rx_bit_dec2 = vitDecoder2(rx_bit2); %decodificacion FEC

end
```

Segmento de Código 2.21. Función del receptor del usuario 2.

2.5 CÁLCULO DE BER

Para calcular los errores en el sistema NOMA-OFDM se compara la información transmitida con la información recibida de cada una de las capas, el comando *biterr* es el encargado de calcular la tasa de errores de bit o conocido como BER. El segmento de Código 2.22 muestra el uso del comando para los fines requeridos.

```
% BER----USUARIO 1
[number1, ratio1] =
biterr(data_bit1(1:end-traceBack), rx_bit_dec1((traceBack+1:end)));

BER1(i)=BER1(i)+ratio1;
% BER----USUARIO 2
[number2, ratio2] =
biterr(data_bit2(1:end-2*traceBack), rx_bit_dec2((traceBack+1:end-
traceBack))); %calculo BER

BER2(i)=BER2(i)+ratio2;
```

Segmento de Código 2.22. Cálculo de BER.

Debido a que para cada valor de SNR_{dB} se genera un numero de iteraciones, se procede a promediar todos los valores calculados de BER para ese valor de SNR. Tal como se muestra en el segmento de Código 2.23

```
BER1(i)=BER1(i)/iter; %BER promedio
BER2(i)=BER2(i)/iter; %BER promedio
```

Segmento de Código 2.23. Cálculo del BER promedio.

2.6 GENERACIÓN DE ARCHIVO DE DATOS

Una vez realizada toda la simulación, se procede a crear un archivo de datos y en este se guarda las variables necesarias para obtener la gráfica y así poder analizar posteriormente. La función *saveDatos* se encarga de generar un nombre para el archivo de forma automática, en función de las variables que se pueden observar en el segmento de Código 2.24. Una vez obtenido el nombre, se usa el comando *save* para crear un archivo con las variables antes mencionadas.

```
NombreArchivo=saveDatos(M1, M2, k1, k2, fac_dB, PC, CFO, tec_CFO,
iter);
save(NombreArchivo, 'EbNo_dB', 'BER1', 'BER2', 'M1', 'M2', ...
'k1', 'k2', 'fac_dB', 'PC', 'CFO', 'tec_CFO')
```

Segmento de Código 2.24. Archivo de datos con variables necesarias.

2.7 GENERACIÓN DE GRÁFICA

En esta parte de la simulación se crea una gráfica, la misma que permita obtener información sobre cuál de las combinaciones de orden de modulación, tasa de codificación, factor de inyección, entre otras. Como en la sección anterior los datos se guardaron en un archivo, se puede ir comparando posteriormente cuales son las configuraciones que permiten tener un mejor rendimiento y presentar estos resultados en el próximo capítulo. El segmento de Código 2.25 muestra cómo se genera la gráfica.

```
%% Grafica.
semilogy(EbNo_dB, BER1, EbNo_dB, BER2)
ylim([0.0001 0.5])
xlim([0 EbNo_dB(end)])
grid on
xlabel('E_b/N_0 [dB]');
ylabel('BER')
ParteNombre1=num2str(M1);
ParteNombre2=num2str(M2);
ParteNombre3=num2str(tec_CFO);
ParteNombre4=num2str(k1);
ParteNombre5=num2str(k2);
ParteNombre6=num2str(fac_dB);
ParteNombre7=num2str(iter);

leyenda1=['Usuario 1; ', ' k1=', ParteNombre4, '; M1=', ...
ParteNombre1,];
leyenda2=['Usuario 2; ', ' k2=', ParteNombre5, '; M2=', ...
ParteNombre2,];
titleLeg=['Tec. Est=', ParteNombre3, '; Fac. Iny.=', ...
ParteNombre6, 'dB; Iter=', ParteNombre7];
title({'BER Vs. Eb/No'; 'Sincronizacion del CFO'});
lgd=legend(leyenda1, leyenda2);
title(lgd, titleLeg);
```

Segmento de Código 2.25. Generación de grafica.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 RESULTADOS

3.1.1 ESCENARIO DE SIMULACIÓN

El sistema de comunicación NOMA-OFDM fue creado en base a la implementación de funciones y también gracias a la ejecución de comandos o algoritmos que vienen presentes en la librería de comunicaciones o conocido como *Communication Toolbox*. El sistema está diseñado en bloques o secciones, estas son: transmisor, canal inalámbrico y receptor. En esta última sección se realiza el proceso de sincronización en donde la señal atraviesa los procesos para este fin, tales como: inserción, estimación utilizando 3 técnicas diferentes y compensación del CFO. La señal resultante en la sección de transmisión viaja a través de un canal inalámbrico, este canal presenta un modelo de distribución Rician, además de que la señal es afectada por el ruido gaussiano blanco (AWGN).

Para cada una de las capas o usuarios se generan datos aleatorios y estos se codifican en base a la tasa de codificación elegida, ya sea $1/2$, $2/3$ y $3/4$. Posteriormente, estos datos son modulados bajo los diferentes esquemas presentados, tales como QPSK, 16-QAM y 64-QAM. La capa 2 es afectada por un factor de inyección estableciendo de esta forma la potencia de esta capa. Finalmente, NOMA suma las dos señales y la señal resultante esta lista para formar parte del símbolo OFDM a transmitir. En la etapa de recepción, una vez que la señal ha sido sincronizada se procede a recuperar la señal de cada una de las capas de manera independiente. En otras palabras, cada una de las capas presenta diferente método para obtener la señal que le corresponde.

3.1.2 PRUEBAS DE FUNCIONAMIENTO

Con el fin de comprobar el correcto funcionamiento del sistema de comunicación implementado se realiza una comparación con el trabajo citado en la sección de referencias bibliográficas en el numeral [9]. Si bien es cierto que ambos trabajos no comparten las mismas tasas de codificación y el canal inalámbrico en ambos casos es diferente, se trata de usar tasas similares a una de las pruebas realizadas por la autora en dicho trabajo, de esta manera observar si el comportamiento o tendencia de las curvas son similares.

Para dicho fin se ha utilizado como referencia el grafico mostrado en la Figura 3.1, mismo que se obtiene desde el trabajo antes mencionado. En dicha figura se puede observar que las tasas de codificación para la capa 1 y la capa 2 son 4/5 y 1/3, respectivamente. También se puede obtener el dato del factor de inyección que se usó para dicha simulación, teniendo de esta manera el valor de -8 dB.

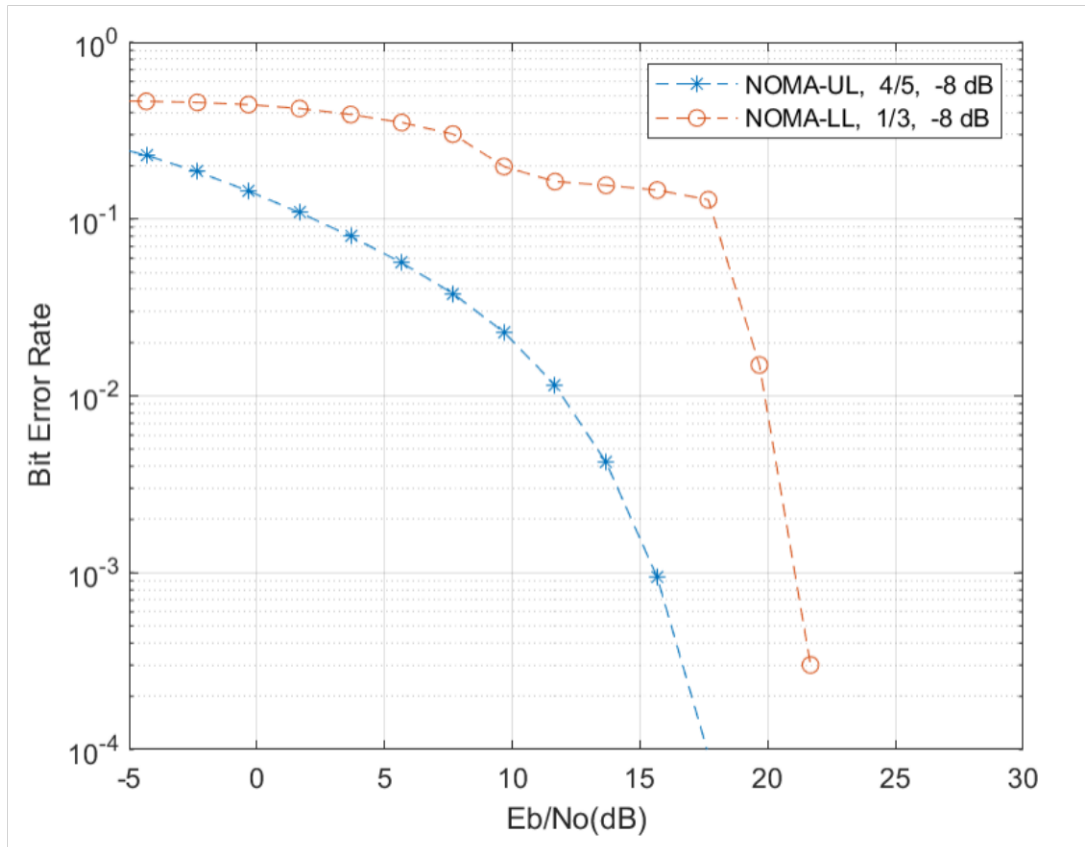


Figura 3.1. BER Vs Eb/No con un factor de inyección de -8 dB y tasas de codificación Capa1: 4/5 y Capa2: 1/3 [9].

Con los datos obtenidos a partir de la gráfica se procede a realizar una simulación con parámetros similares, por lo que se ha decidido usar tasas de codificación de 3/4 y 1/2 para la capa 1 y capa 2, respectivamente. De igual forma, se establece el factor de inyección a -8 dB. Debido a que el trabajo mencionado anteriormente no se enfoca en el tema de sincronización, la simulación realizada no utiliza ninguna técnica de estimación y tampoco se inserta en CFO a la señal, dicho en otras palabras, se considera una sincronización perfecta. Los resultados de la simulación se presentan en la Figura 3.2.

Al comparar ambas graficas se puede observar que la tendencia o comportamiento de las curvas es similar, por lo que el sistema de comunicación funciona de acuerdo con lo esperado.

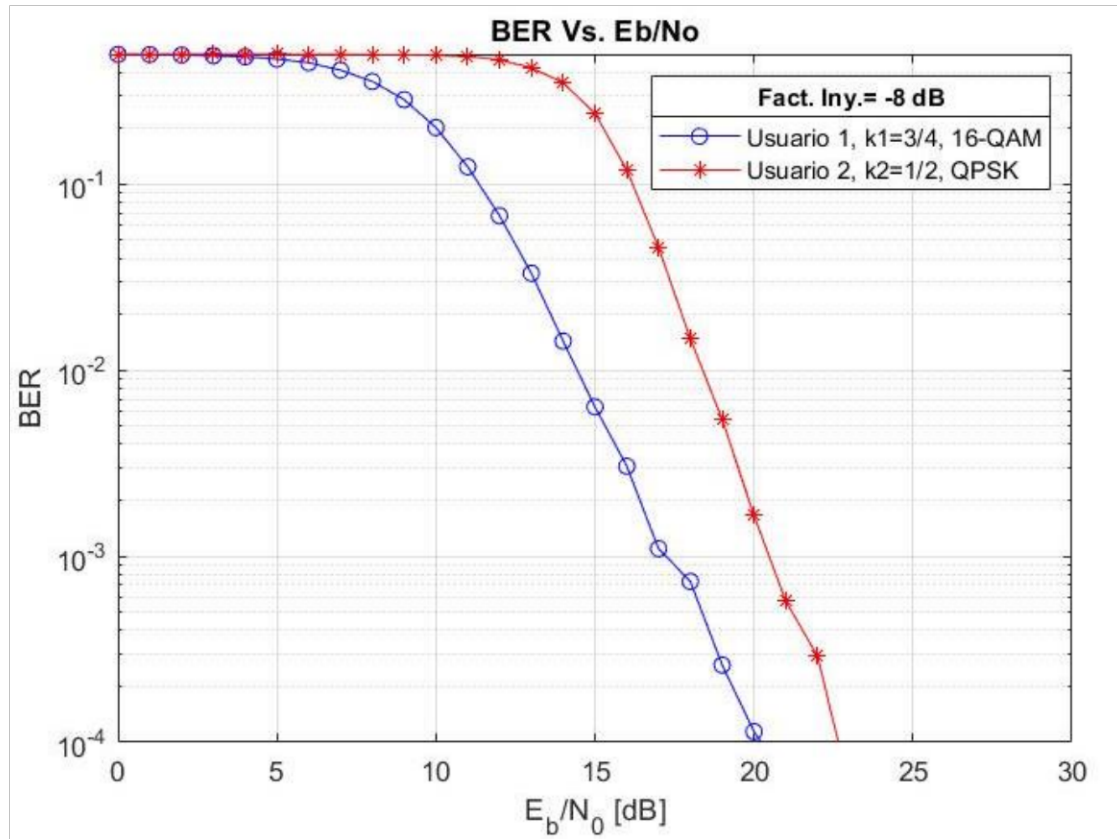


Figura 3.2. BER Vs E_b/N_0 con un factor de inyección de -8 dB y tasas de codificación Capa1: 3/4 y Capa2: 1/2.

3.1.3 RESULTADOS CON SINCRONIZACIÓN DEL CFO

A continuación, se presentan las simulaciones realizadas en el sistema de comunicación NOMA-OFDM que fue presentado en el capítulo 2, para este fin se ha utilizado el software de simulación MATLAB con la versión R2021b, dichas simulaciones poseen la configuración de parámetros presentados en la Tabla 3.1.

Tabla 3.1. Parámetros fijos de simulación.

Número de iteraciones.	4000
Número de símbolos	5
Tamaño de la FFT	64
Número de subportadoras de datos	48
Número de subportadoras piloto	4
Prefijo cíclico	16

El medio de transmisión es un modelo de canal multitrayecto Rician que posee el PDP (*Power Delay Profile*) presentado en la Tabla 3.2.

Tabla 3.2. PDP del canal Rician.

<i>Power Delay Profile</i>	
Retraso Relativo [ns]	Potencia Promedio [dB]
0	0
100	-14
200	-17

3.1.3.1 Técnica de estimación de CFO en el dominio del tiempo utilizando el prefijo cíclico

La primera técnica que se analiza es la de estimación de CFO en el dominio del tiempo haciendo uso del prefijo cíclico. La Figura 3.3 presenta la primera simulación de esta subsección, la cual posee las siguientes configuraciones de cada una de las capas: para la capa 1 se ha configurado una tasa de codificación (k_1) de $3/4$ y el esquema de modulación (M1) 16-QAM, mientras que para la capa 2 se establece una tasa de codificación (k_2) de $2/3$, el esquema de modulación (M2) QPSK, el factor de inyección (g) para esta simulación es de -8 y el valor de CFO = $0,21$ que se utiliza en toda la subsección. En la figura se puede observar el rendimiento de la técnica utilizada, ya que al compararla con las curvas entrecortadas que pertenecen a sincronización perfecta difieren en gran manera ya que se espera que las curvas obtenidas gracias a la técnica de estimación (curvas solidas con marcadores) sean semejantes o que tengan la misma tendencia a las curvas de sincronización perfecta. Por lo que para esta primera combinación de parámetros el rendimiento de la técnica no es bueno.

Al buscar mejorar el rendimiento de la técnica de sincronización, se modifica la configuración de las capas, incluyendo el factor de inyección. Es así como se establece la tasa de codificación para la capa 1 y capa 2 en $2/3$ y $1/2$, respectivamente, y ambas se modulan bajo el esquema QPSK, el factor de inyección se configura en un valor de -4 dB. Esto se presenta en la Figura 3.4, en donde se observa el rendimiento o comportamiento de ambas capas (curvas con marcadores) con respecto a la sincronización perfecta (curvas con líneas entrecortadas). Si bien es cierto que las curvas que representan los resultados

de la técnica de sincronización con las configuraciones de capas mencionadas presentan un pendiente más pronunciada hasta cierto valor de BER con respecto a las curvas presentadas en la Figura 3.3, no muestran una gran mejora en el rendimiento del sistema de comunicación ya que estas logran alcanzar un valor de $BER=10^{-2}$ y posteriormente marcan un piso de ruido, de tal manera que las curvas dadas por la técnica de estimación difieren totalmente con las curvas de sincronización perfecta.

Al revisar ambas graficas con las distintas configuraciones de capas y diferente factor de inyección se concluye que la técnica de estimación en el dominio del tiempo usando el prefijo cíclico no es muy buena para ambas capas, ya que en ambos casos la diferencia entre las curvas de sincronización perfecta y las curvas dadas por la técnica de estimación no muestran ninguna relación y esto se debe a la incorrecta estimación del valor de CFO por la técnica usada en esta sección.

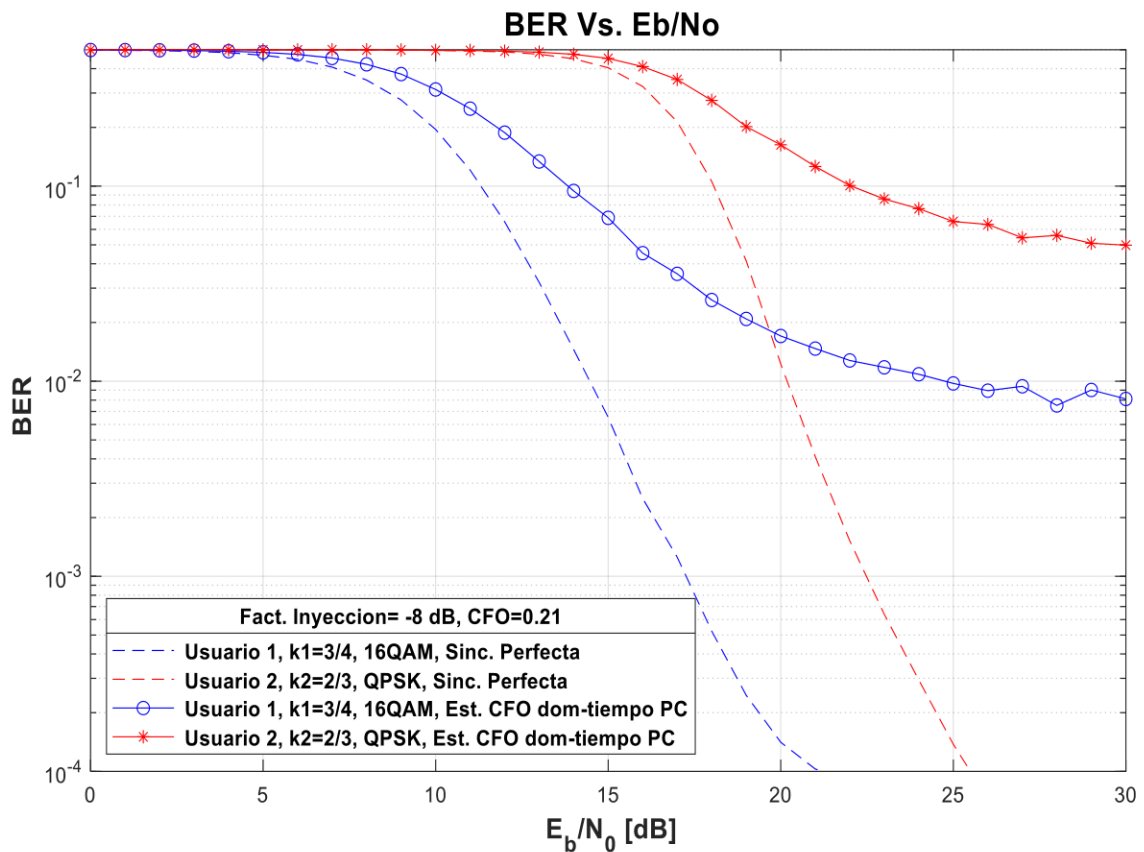


Figura 3.3. BER Vs Eb/No. Tec. de estimación del CFO dominio-tiempo con PC, $g = -8$ dB, CFO=0.21, $k_1=3/4$, $k_2=2/3$, M1=16QAM y M2=QPSK.

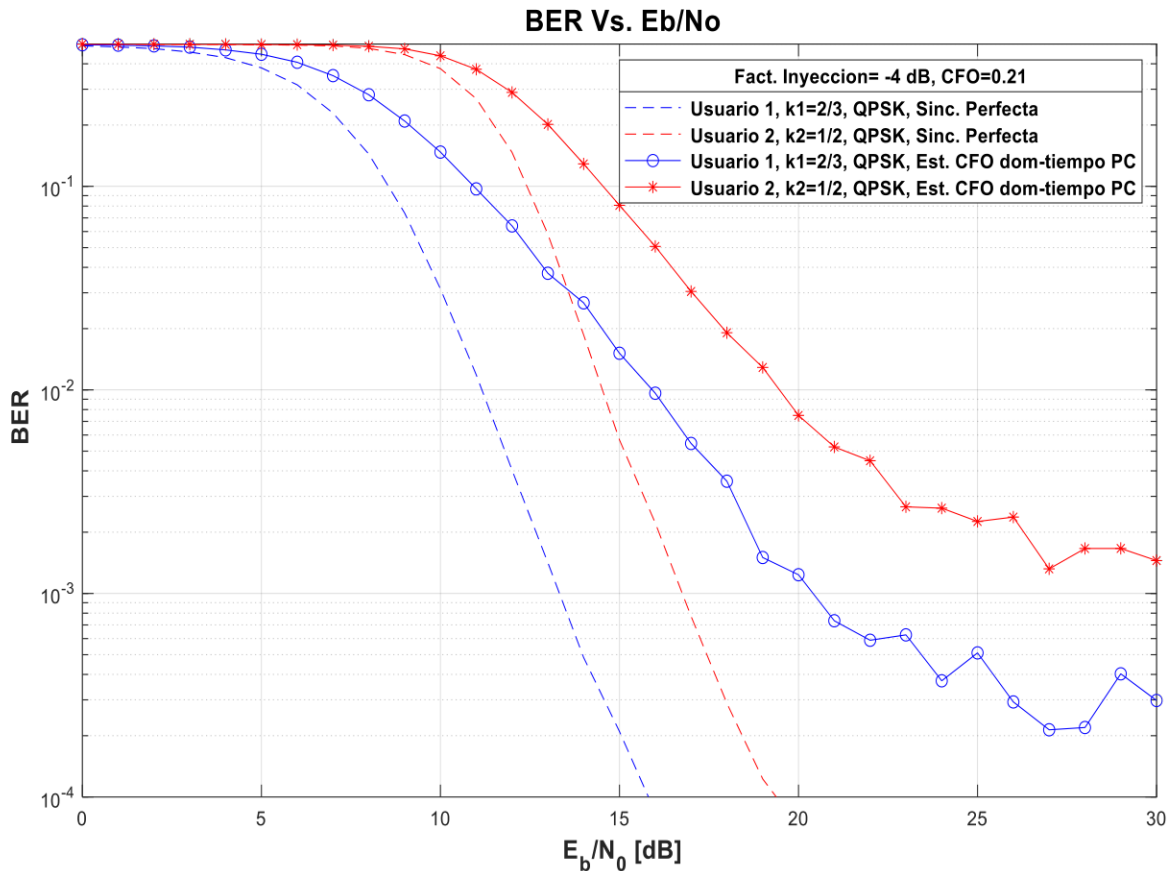


Figura 3.4. BER Vs E_b/N_0 . Tec. de estimación del CFO dominio-tiempo con PC, $g = -4$ dB, CFO=0.21, $k_1=2/3$, $k_2=1/2$, M1=QPSK y M2=QPSK.

3.1.3.2 Técnica de estimación de CFO en el dominio del tiempo utilizando el preámbulo

Con las pruebas realizadas para esta subsección, se encuentra que esta es la técnica menos eficiente de las tres presentadas en el capítulo 1 e implementadas en el capítulo 2, en este último capítulo en mención se trató de mejorar el rendimiento de la técnica añadiendo un algoritmo de comparación en donde se trabaja con las partes negativas del preámbulo, aun con estas modificaciones el resultado que entrega la técnica no es para nada el esperado. Para la simulación de esta subsección se ha configurado las tasas de codificación a 1/2 y esquema de modulación QPSK para ambas capas, además de que se establece un factor de inyección (g) de -4 dB y el valor de CFO = 0,21. La Figura 3.5 muestra el rendimiento explicado, en donde las curvas entrecortadas pertenecen a la sincronización perfecta y las curvas con marcadores a la técnica de estimación usada en esta subsección, en esta figura podemos observar que las curvas de la técnica de

estimación presentan un piso de ruido a muy temprano valor de E_b/N_0 de tal forma que la técnica usada en esta subsección no logra estimar adecuadamente el CFO.

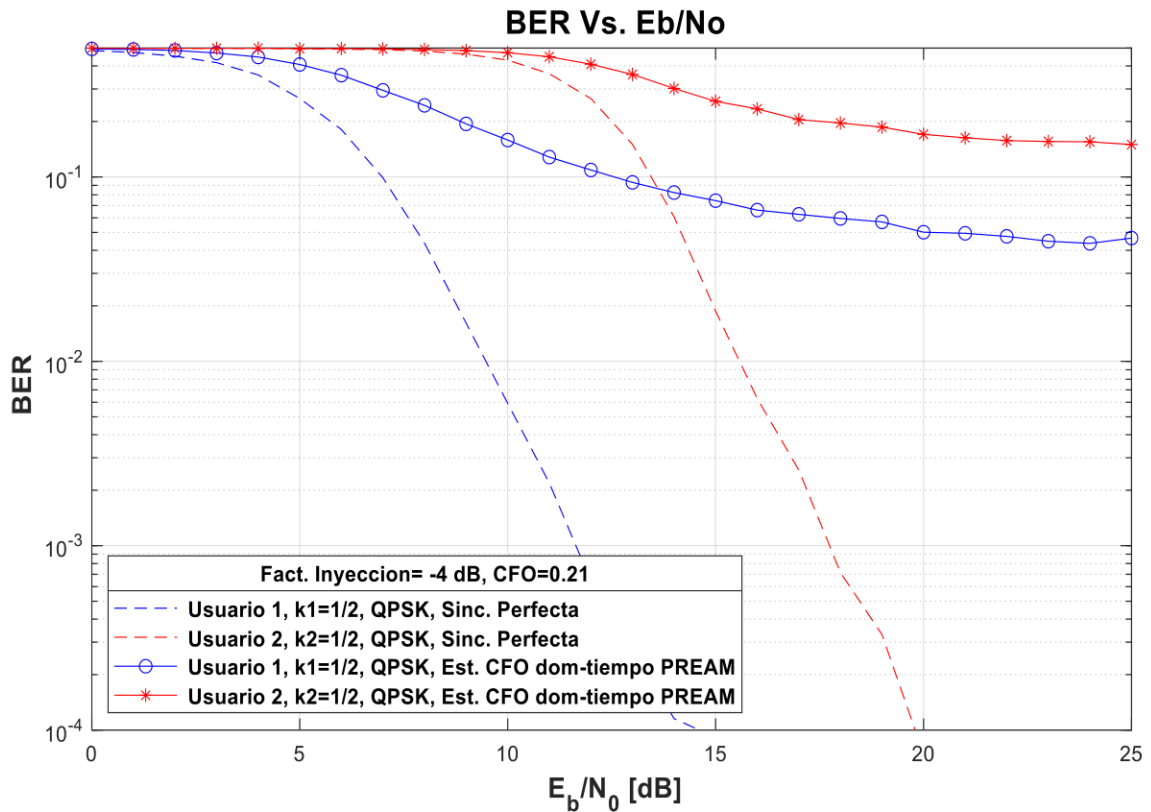


Figura 3.5. BER Vs E_b/N_0 . Tec. de estimación del CFO dominio-tiempo con Preámbulo, $g = -4$ dB, $CFO=0.21$, $k_1=1/2$, $k_2=1/2$, $M_1=QPSK$ y $M_2=QPSK$.

3.1.3.3 Técnica de estimación de CFO en el dominio de la frecuencia utilizando el preámbulo.

La última técnica en analizar es la de estimación del CFO en el dominio de la frecuencia usando el preámbulo, en la Figura 3.6 se puede observar los resultados de la primera configuración de capas para esta subsección, en donde se encuentra que las tasas de codificación son 1/2 para ambas capas, esquemas de modulación 16QAM y QPSK, para la capa 1 y capa 2, respectivamente, factor de inyección de -6 dB y el valor de $CFO = 0,21$. Al igual que en los casos anteriores, las curvas con marcadores pertenecen a los resultados de la técnica de estimación, mientras que las curvas con líneas entrecortadas pertenecen a la sincronización perfecta, la azul para la capa 1 y la roja para la capa 2. En la Figura se puede observar el rendimiento de la técnica de estimación con la configuración mencionada en donde las curvas llegan al valor de $BER=10^{-3}$ pero con valores de E_b/N_0 superiores a los presentados por la sincronización perfecta, por tal motivo existe mucha diferencia entre los pares de curvas mencionadas. Esta primera simulación con la técnica de estimación en el dominio de la frecuencia utilizando el preámbulo muestra mejores resultados que las

técnicas presentadas anteriormente, lo que indica que con la correcta configuración de las capas y el factor de inyección adecuado se puede obtener el mejor rendimiento del sistema de comunicación usando esta técnica de estimación.

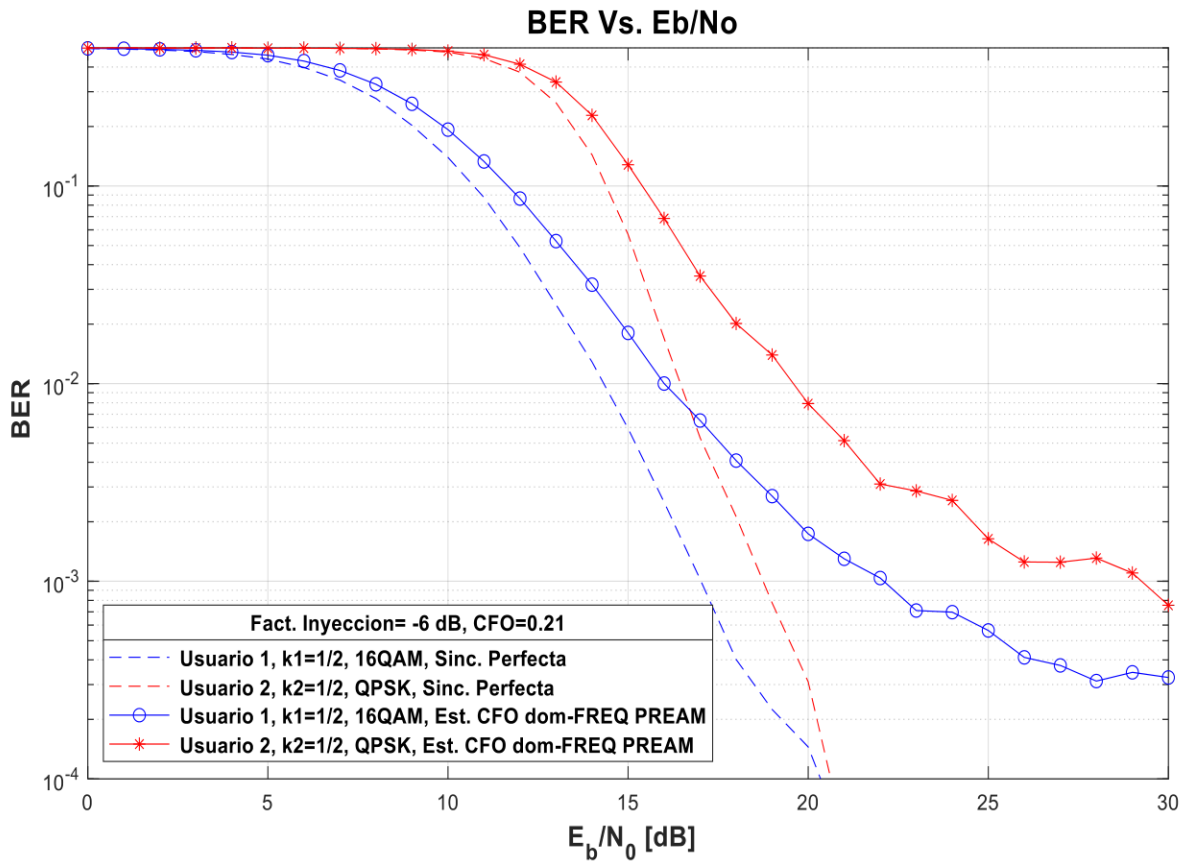


Figura 3.6. BER Vs Eb/No. Tec. de estimación del CFO dominio-frecuencia con Preámbulo, $g = -6$ dB, CFO=0.21, $k_1=1/2$, $k_2=1/2$, M1= 16QAM y M2=QPSK.

La configuración que se muestra en la Figura 3.7 permite tener el mejor rendimiento de la técnica de estimación analizada en esta subsección, las tasas de codificación son configuradas en 3/4 y 1/2 para la capa 1 y capa 2, respectivamente. Mientras que el esquema de modulación es el QPSK para ambas capas, el factor de inyección establecido es de -4 dB y el valor de CFO = 0,21. Las curvas con líneas entrecortadas sirven como referencias para determinar el rendimiento del sistema ya que estas pertenecen a la sincronización perfecta, por tal motivo se puede observar que las curvas con marcadores pertenecientes a la técnica de estimación tienen una tendencia o comportamiento muy similar a las de referencia, por lo que esta técnica de estimación con la configuración presentada trabaja de acuerdo con lo esperado.

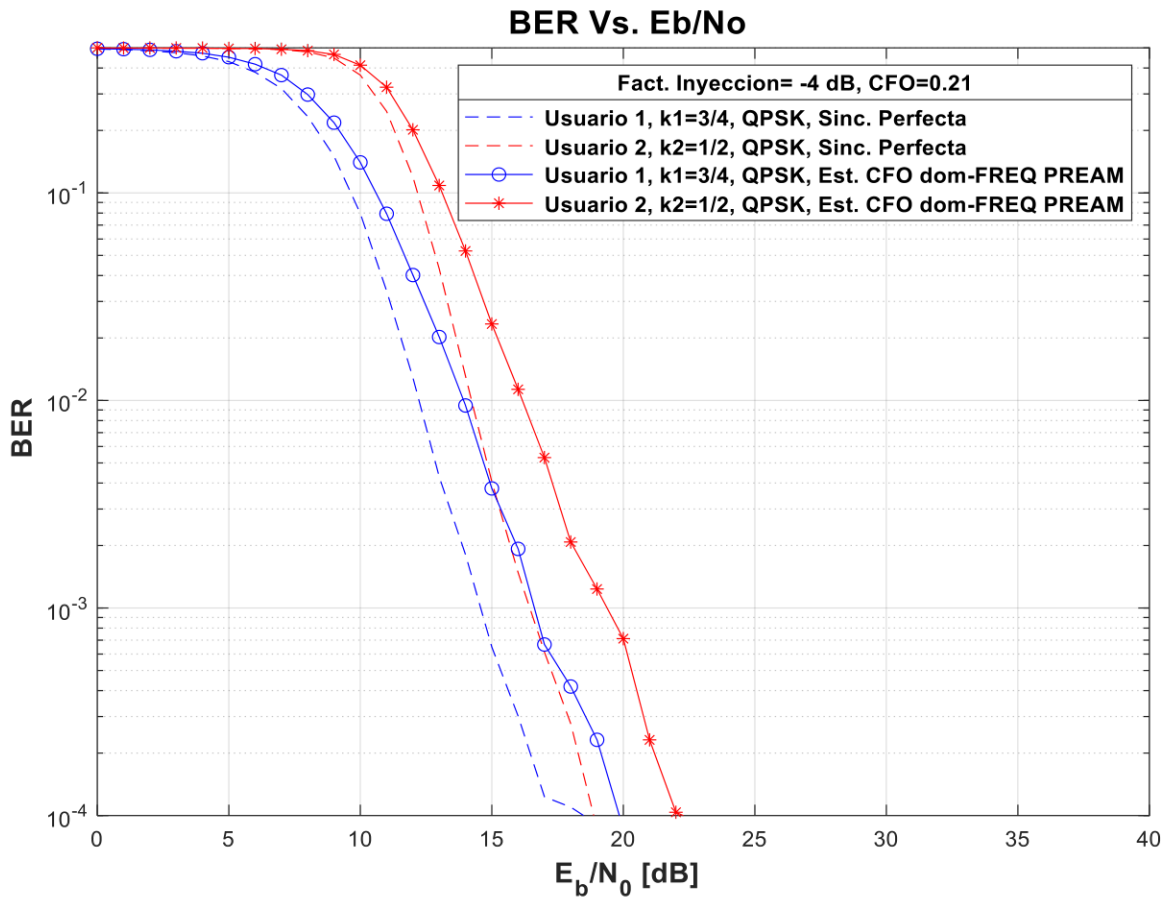


Figura 3.7. BER Vs Eb/No. Tec. de estimación del CFO dominio-frecuencia con Preámbulo, $g = -4$ dB, CFO=0.21, $k_1=3/4$, $k_2=1/2$, M1=QPSK y M2=QPSK.

Una vez analizadas las técnicas de estimación presentadas en los capítulos anteriores con las respectivas configuraciones que permiten su mejor rendimiento, se logra determinar que la técnica con mejor rendimiento es la tercera, es decir, la técnica de estimación en el dominio de la frecuencia utilizando el preámbulo con la configuración presentada en la Figura 3.7. A diferencia de las otras técnicas y configuraciones presentadas, esta última permite tener unas curvas con sus tendencias y comportamientos bien marcados y parecidos al de su referencia (sincronización perfecta), además de que se elimina el piso de ruido presentado en las figuras anteriores.

Con el fin de corroborar el rendimiento de la técnica de estimación en el dominio de la frecuencia utilizando el preámbulo se realiza una nueva simulación en la que se configura el valor del CFO insertado a 0.35 y se mantiene la configuración de las capas mencionadas anteriormente. El resultado de esta simulación se presenta en la Figura 3.8, en donde se puede confirmar el buen rendimiento de la técnica de estimación dentro del sistema de comunicación ya que el comportamiento de las curvas es el esperado.

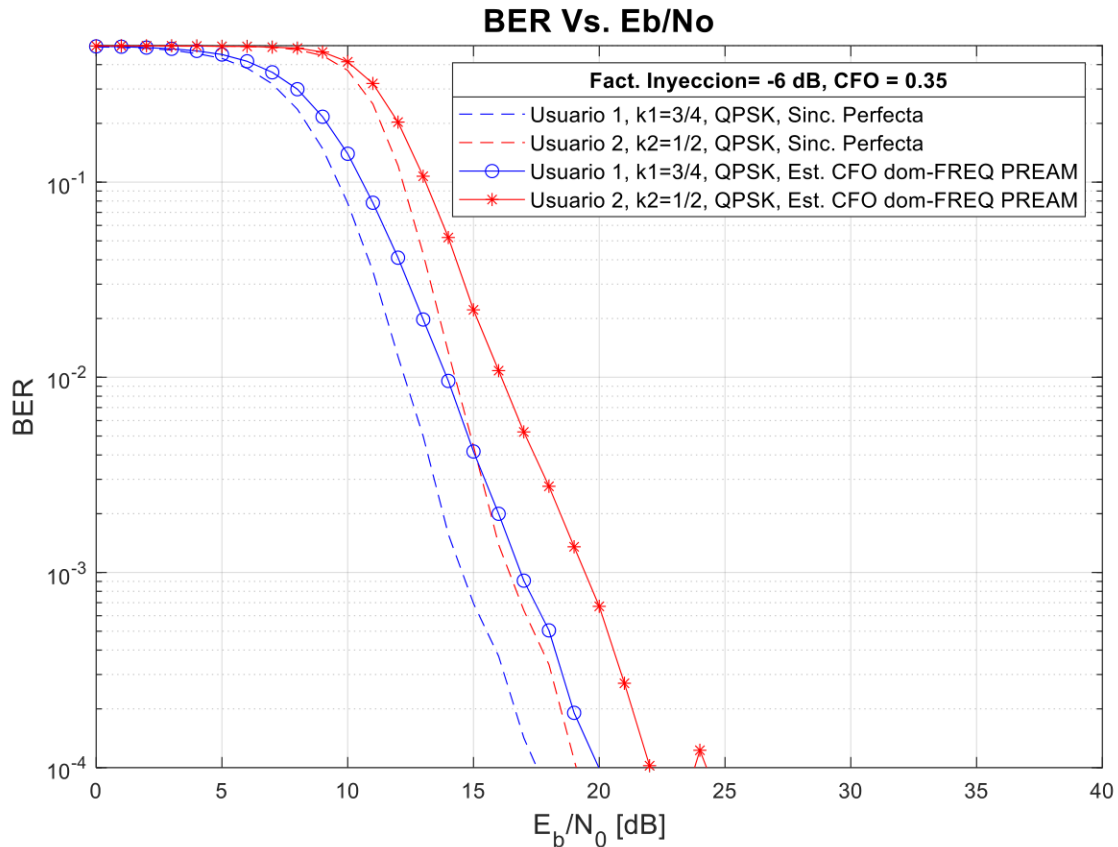


Figura 3.8. BER Vs Eb/No. Tec. de estimación del CFO dominio-frecuencia con Preámbulo, $g = -4$ dB, CFO=0.35, $k_1=3/4$, $k_2=1/2$, M1=QPSK y M2=QPSK.

3.2 CONCLUSIONES

En el desarrollo del presente trabajo se detalló y se implementó cada uno de los bloques o secciones que conforman el sistema de comunicación NOMA-OFDM, así también las características de funcionamiento de cada una de estas. Al ser un sistema bajo un orden secuencial, dicho en otras palabras, se tiene que seguir cierta disposición, se fueron implementado las secciones requeridas, como son: transmisor, canal inalámbrico y receptor en el que incluye la sincronización, en ese orden. El correcto diseño e implementación de cada uno de los bloques permite asegurar el éxito del bloque que viene a continuación, teniendo como resultado un sistema de comunicación con un comportamiento o rendimiento de acuerdo con lo esperado.

Las técnicas de estimación en el dominio del tiempo no brindan buenos resultados en la sincronización. La técnica de estimación de CFO en el dominio del tiempo utilizando el prefijo cíclico muestra una tendencia o comportamiento definido de sus curvas hasta alcanzar el valor de $BER=10^{-2}$, posteriormente ambas curvas marcan un piso de ruido. La

configuración de las capas para obtener dichos resultados es: tasas de codificación para la capa 1 y capa 2 en $2/3$ y $1/2$, respectivamente, esquema de modulación QPSK para ambas capas y factor de inyección en -4 dB. La técnica de estimación de CFO en el dominio del tiempo utilizando el preámbulo es la que peor rendimiento presenta ya que se marca un piso de ruido desde bajos valores de E_b/N_0 , la configuración de las capas para obtener los resultados presentados es: tasas de codificación a $1/2$ y esquema de modulación QPSK para ambas capas y factor de inyección en -4 dB.

Con el fin de mejorar el rendimiento de la técnica de estimación de CFO en el dominio de tiempo utilizando el prefijo cíclico, se procedió a incrementar las operaciones que se realizan dentro de la función implementada en MATLAB en base a la Ecuación 1.2, en donde se calcula el ángulo de fase que viene dado por el producto entre el PC y su correspondiente al final de cada símbolo OFDM, al tener 5 símbolos se realizan 5 operaciones dentro de la función. De esta manera se logró evidenciar una mejora significativa en el resultado de la técnica. En la técnica de estimación del CFO en el dominio del tiempo utilizando el preámbulo se realizó el mismo arreglo, obteniendo una respuesta no favorable ya que la técnica no logró estimar el valor de CFO aun con las modificaciones realizadas.

Se realizaron pruebas de funcionamiento del sistema de comunicaciones NOMA-OFDM implementado en el software MATLAB, en donde a cada una de las capas se configuró su tasa de codificación y esquema de modulación. Además, se estableció un factor de inyección para el sistema, todo esto buscando la configuración que permita tener el mejor rendimiento con cada una de las técnicas de estimación presentadas.

Una vez analizadas las técnicas con sus distintas configuraciones se puede concluir que el rendimiento del sistema depende netamente de la estimación de CFO, este valor debe ser muy cercano al valor que fue insertado en la señal ya que el sistema es muy sensible y pequeñas diferencias entre el valor insertado y calculado puede generar errores muy grandes.

Los resultados presentados en el capítulo 3 muestran que la mejor técnica de estimación del CFO es aquella que trabaja en el dominio de la frecuencia y utiliza el preámbulo, ya que presenta un buen rendimiento y es robusta frente los problemas por los que atraviesa la señal. La configuración de las capas para obtener el mejor rendimiento de la técnica en mención es: tasas de codificación configuradas en $3/4$ y $1/2$ para la capa 1 y capa 2, respectivamente, esquema de modulación QPSK para ambas capas, el factor de inyección en -4 dB.

3.3 RECOMENDACIONES

El número de iteraciones que se realiza dentro de la simulación para cada uno de los valores de E_b/N_0 es muy importante para obtener datos confiables, por lo que se recomienda usar 4000 iteraciones. También esto permite obtener gráficas con curvas definidas y que permitan identificar bien su tendencia o comportamiento.

Para el análisis de resultados se recomienda primeramente configurar las capas y una vez encontrado un buen rendimiento, proceder a modificar el factor de inyección del sistema y lograr que este sea el mejor dentro de cada escenario o técnica de estimación.

Como futuro trabajo, se puede analizar la técnica de estimación de CFO en el dominio del tiempo utilizando el preámbulo ya que por ahora es la técnica que menos rendimiento presenta, buscar una solución para que el valor estimado por la técnica sea muy cercano al insertado a la señal.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] A. F. Molisch, *Wireless Communications*, University of Southern California, 2011.
- [2] A. Goldsmith, *Wireless Communications*, Stanford University, 2014.
- [3] R. Prasad, *OFDM for Wireless Communication System*, Boston-London: Artech House, Inc., 2004.
- [4] W. Stallings, *Wireless Communications and Networks*, U.S.A: Pearson Education, Inc., 2005.
- [5] A. G. M. Bhardway, A Review on OFDM: Concept, Scope & its Applications, *IOSR Journal of Mechanical and Civil Engineering*, 2012.
- [6] T. Ayman y A. Huseyin, *NOMA for Multinumerology OFDM Systems*, USA, 2018.
- [7] D. Zhiguo, L. Yuanwei, C. Jinho, S. Qi, E. Maged, I. Chih-Lin y V. Poor, Application of Non-orthogonal Multiple Access in LTE and 5G Networks, *IEEE Communications Magazine*, 2016.
- [8] K. Higuchi, Non-orthogonal Multiple Access (NOMA) with Successive, *IEICE Transactions on Communications*, 2015.
- [9] C. Arrieta, Simulación de un sistema de comunicación inalámbrica usando la técnica de acceso NOMA, Quito: Escuela Politécnica Nacional, 2020.
- [10] G. Javier, Optimización algorítmica orientada a la implementación en dispositivos reconfigurables de esquemas multiestandar y multiusuario para sincronismo de frecuencia en sistemas de comunicaciones inalámbricas basados en OFDM, Madrid: Universidad Politécnica de Madrid, 2011.
- [11] K. Toromoreno, SIMULACIÓN DE TÉCNICAS DE SINCRONIZACIÓN EN SISTEMAS OFDM, Quito: Escuela Politécnica Nacional, 2020.

5 ANEXOS

ANEXO I. Scripts y funciones de la simulación en Matlab (Anexo Digital)