

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

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

**SIMULACIÓN DE STO (SYMBOL TIME OFFSET) EN LA TÉCNICA
NOMA**

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

STALIN DAVID CHICAIZA JAMI

stalin.chicaiza@epn.edu.ec

DIRECTOR: DIEGO JAVIER REINOSO CHISAGUANO

diego.reinoso@epn.edu.ec

DMQ, febrero 2022

CERTIFICACIONES

Yo, STALIN DAVID CHICAIZA JAMI 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.

STALIN DAVID CHICAIZA JAMI

Certifico que el presente trabajo de integración curricular fue desarrollado por STALIN DAVID CHICAIZA JAMI, bajo mi supervisión.

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.

STALIN DAVID CHICAIZA JAMI

DIEGO JAVIER REINOSO CHISAGUANO

DEDICATORIA

A mis padres Willam e Isabel quienes me brindaron apoyo incondicional desde el primer momento que inicié la carrera, ya que, sin su ayuda esto no sería posible.

A mis hermanos Kerly y Stiven quienes me motivaron a no rendirme, siendo para ellos espejo de responsabilidad y superación.

AGRADECIMIENTO

Quiero agradecer en primer lugar a Dios por bendecir a toda mi familia y mantenernos juntos con salud.

A mis padres Willam e Isabel por brindarme apoyo en los buenos y malos momentos, a mis hermanos por ser mi alegría, a mi familia que me apoyó cuando estuve lejos de casa durante la carrera y a mis amigos quienes estuvieron siempre que los necesité y me animaban a no rendirme.

Al Dr. Diego Reinoso por brindarme apoyo en la realización de este trabajo con su conocimiento y paciencia en cada una de las etapas.

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)	6
1.4.4 Sincronización NOMA-OFDM	8
2 METODOLOGÍA.....	12
2.1 Sistema de Transmisión NOMA-OFDM	13
2.1.1 Generación de bits aleatorios para cada usuario	14
2.1.2 Codificación	15
2.1.3 Modulación	16
2.1.4 Suma de señales y uso del factor de inyección	16
2.1.5 Multiplexación OFDM.....	16
2.1.6 Preámbulo	17
2.2 Canal.....	18
2.3 Sistema de Recepción NOMA-OFDM	20
2.3.1 Sincronización NOMA-OFDM	22
2.3.2 Extraer preámbulo	27
2.3.3 Demultiplexación OFDM	27
2.3.4 Ecuación	28
2.3.5 Capa uno	29
2.3.6 Capa dos	29

2.4	Cálculo del BER y generación de gráficas	30
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	31
3.1	Escenario de simulación	31
3.2	Pruebas del sistema NOMA.....	32
3.3	Resultados	34
3.3.1	Resultados de simulaciones con modulación 16 QAM para capa uno	37
3.4	Conclusiones	40
3.5	Recomendaciones	41
4	REFERENCIAS BIBLIOGRÁFICAS.....	42
5	ANEXOS	44

RESUMEN

En el presente trabajo se muestra la simulación de STO (*Symbol Time Offset*) en el sistema de comunicación NOMA (*Non Orthogonal Multiple Access*) basado en OFDM (*Orthogonal frequency-division multiplexing*). Por esta razón, se emplean dos técnicas de sincronización en el lado de receptor de cada usuario. Estas técnicas permiten estimar el STO para encontrar el inicio del símbolo con la finalidad de sincronizar el sistema. El sistema NOMA está compuesto de capas que permiten la asignación de recursos a múltiples usuarios desde el dominio de la potencia, cada capa es configurada de forma independiente.

En el primer capítulo, se presentan los conceptos generales teóricos del presente trabajo, se estudia las características de la técnica de acceso NOMA, la técnica de transmisión OFDM y se detallan las técnicas de estimación del STO que se aplican en el sistema.

En el segundo capítulo, se detalla el código implementado en Matlab del transmisor, canal y receptor de la técnica NOMA, dentro del receptor se explica las técnicas de sincronización implementadas.

Finalmente, en el tercer capítulo se presenta el análisis de los resultados de las simulaciones con las diferentes configuraciones de modulación, tasa de codificación y factor de inyección aplicando las técnicas de estimación de STO para la sincronización, mediante las gráficas generadas en función del BER (*Bit Error Rate*) vs E_b/N_0 (*Energy per-bit to noise power spectral density ratio*). Adicionalmente, se presentan las conclusiones y recomendaciones del trabajo realizado.

PALABRAS CLAVE: NOMA, OFDM, STO, factor de inyección.

ABSTRACT

This work shows the simulation of STO (Symbol Time Offset) in the NOMA (Non Orthogonal Multiple Access) communication system based on OFDM (Orthogonal frequency-division multiplexing). For this reason, two synchronization techniques are used on the receiver side of each user. These techniques allow to estimate the STO to find the start of the symbol in order to synchronize the system. The NOMA system is composed of layers that allow the allocation of resources to multiple users from the power domain, each layer is configured independently.

The first chapter presents the general theoretical concepts of this work, studies the characteristics of the NOMA access technique, the OFDM transmission technique and details the STO estimation techniques applied in the system.

In the second chapter, the code implemented in Matlab of the transmitter, channel and receiver of the NOMA technique is detailed, within the receiver the implemented synchronization techniques are explained.

Finally, the third chapter presents the analysis of the results of the simulations with different modulation configurations, coding rate and injection factor applying the STO estimation techniques for synchronization, through the graphs generated as a function of BER (Bit Error Rate) vs E_b/N_0 (Energy per-bit to noise power spectral density ratio). Additionally, the conclusions and recommendations of the work are presented.

KEYWORDS: NOMA, OFDM, STO, injection factor

1 INTRODUCCIÓN

Las comunicaciones inalámbricas han permitido un gran desarrollo en la humanidad, satisfaciendo la necesidad de comunicación entre usuarios sin la conexión física de los dispositivos usando un canal natural. Al pasar de los años han presenciado una gran cantidad de desafíos en el medio de transmisión al ser esta variante en el tiempo. Por esta razón, las nuevas tecnologías han ido reduciendo estos problemas con la finalidad de satisfacer la demanda de los servicios de la comunicación inalámbrica cumpliendo con las expectativas de cantidad de usuarios y tráfico de información.

Dado la necesidad de resolver estos problemas existentes en la transmisión inalámbrica, se han propuesto varias técnicas de las cuales OFDM (*Orthogonal frequency-division multiplexing*) permite la transmisión con múltiples beneficios debido a su alta eficiencia y robustez a los efectos que tiene el medio de transmisión por lo que mitiga los problemas existentes. Esta técnica se encuentra en varios estándares y sistemas de comunicación debido a sus características positivas, sin embargo, los sistemas OFDM son sensibles a los errores de sincronización [1].

En la actualidad, las comunicaciones inalámbricas presentan altos requerimientos generando necesidades en el desempeño de las redes debido al incremento de usuarios y el tráfico de datos, por ello se ha desarrollado un nuevo esquema de acceso al canal, el cual se conoce como NOMA (*Non Orthogonal Multiple Access*). Es una tecnología novedosa que permite una mayor optimización del canal y un alto flujo de datos en el medio inalámbrico, ya que, cuenta con una estructura multicapa las cuales son configuradas de forma independiente [2]. NOMA es un candidato para las redes de comunicación futuras, por lo cual se puede combinar con OFDM para lograr una mayor eficiencia espectral dando como resultado NOMA-OFDM.

Sin embargo, se obtiene una mayor complejidad en el receptor, además, la señal puede sufrir desfases en el tiempo o frecuencia lo cual genera pérdida en la ortogonalidad entre subportadoras complicando la sincronización de la señal y generando deficiencia en la transmisión correcta de datos [1], [2]. Por ello, se analiza en este componente el desfase en tiempo del símbolo conocido como STO (*Symbol Time Offset*) en la técnica NOMA.

Para resolver el problema de la ortogonalidad entre subportadoras se debe realizar un proceso de sincronización en el receptor, en donde, se debe estimar y compensar el desfase dado por STO. Para el análisis del componente, se presenta una simulación en Matlab con el sistema NOMA-OFDM en donde se aplicarán técnicas para estimar y compensar los efectos de la señal afectada por el STO.

1.1 OBJETIVO GENERAL

El objetivo general del componente es:

Simular el Symbol time Offset en la técnica NOMA utilizando Matlab.

1.2 OBJETIVOS ESPECÍFICOS

Los objetivos específicos del componente son:

1. Estudiar las técnicas de sincronización para una señal afectada por STO.
2. Realizar la simulación de un sistema NOMA-OFDM con las técnicas de sincronización que permitan la estimación y compensación del STO.
3. Realizar pruebas de la simulación con diferentes parámetros como: tasa de codificación, modulación y factor de inyección.
4. Analizar los resultados obtenidos en la simulación.

1.3 ALCANCE

En este componente se realizará una simulación en Matlab del sistema NOMA basado en OFDM el cual combina dos flujos de datos, es decir, está formado de una capa uno y capa dos que llegan a dos usuarios. El sistema está compuesto de un codificador convolucional que podrá generar tasas de $1/2$, $2/3$ y $3/4$. Seguido, se podrá utilizar modulación QPSK, 16 QAM o 64 QAM y el factor de inyección será un parámetro configurable. Se tendrá ruido AWGN (*Additive White Gaussian Noise*) y se considerará un canal con desvanecimiento Rician.

En la señal se inserta un desplazamiento para simular el STO, además, se transmitirá un preámbulo con el cual se podrá estimar el STO en el receptor. Se aplicarán técnicas de sincronización en tiempo para realizar la estimación y luego compensarlo. Para la recuperación de datos de los usuarios, en el receptor del usuario 1 únicamente se recupera la capa uno, mientras que para el usuario 2 se debe recuperar la capa uno para luego recuperar la capa dos.

En la simulación se obtendrá la curva BER (*Bit Error Rate*) vs E_b/N_0 (*Energy per-bit to noise power spectral density ratio*), para cada usuario. Para las pruebas se seleccionará dos configuraciones diferentes de tasas de codificación y modulación las cuales serán elegidas en base a los resultados obtenidos en las simulaciones, también, se probará con

dos factores de inyección. Posteriormente, se analizará los resultados obtenidos y finalmente se obtendrán las conclusiones y recomendaciones.

En la fase de diseño, se estudiará los conceptos generales de la técnica NOMA, multiplexación OFDM y las técnicas de sincronización para el STO, con la finalidad de plantear el diagrama de bloques del sistema NOMA-OFDM. En la fase de implementación como primer paso, se realizará en MATLAB el sistema NOMA basado en OFDM, en donde se implementará dentro del receptor las técnicas de sincronización para la señal con STO. Para luego, obtener las gráficas del BER de cada usuario con las técnicas de estimación STO utilizadas. En la última fase se realizarán las pruebas para verificar el correcto funcionamiento del sistema implementado y después, llevar a cabo el análisis de los resultados obtenidos de la simulación.

1.4 MARCO TEÓRICO

1.4.1 COMUNICACIONES INALÁMBRICAS

La primera transmisión sin la utilización de cables se dio en 1888 por el físico alemán Rudolf Hertz a través de ondas electromagnéticas, luego de este suceso las ondas de radio fueron consideradas como un medio de comunicación en 1894 [3]. En la actualidad el incremento de la demanda de servicios de comunicación inalámbrica se ha dado de manera exponencial, por lo cual, requieren un constante desarrollo de nuevas tecnologías, técnicas de transmisión y estándares que permitan mejorar la eficiencia en la comunicación al momento de enviar la información por medios de transmisión desfavorables [4].

El desafío que tienen los sistemas de comunicación inalámbrica son varios como la velocidad de transmisión, gran tráfico de datos y ofrecer servicio de calidad a la gran demanda de usuarios. En este sentido, es necesario la utilización de tecnologías que presenten alta eficiencia espectral y confiabilidad en la transmisión [4]. Por esta razón, se presenta el desarrollo de nuevas tecnologías, sistemas de comunicación como son los medios múltiples de transmisión y redes de múltiples capas [5].

Su popularidad se debe a que no necesita una conexión física entre el emisor y el receptor por lo cual le proporciona movilidad y comunicación instantánea en cualquier parte a todo momento en cualquier lugar del mundo [2]. Además, de generar grandes impactos en la sociedad como la economía, la industria, la educación, el turismo, el entretenimiento, entre otros [5]. Sin embargo, al ser su medio de transmisión el ambiente natural, también conocido como canal inalámbrico presenta varios inconvenientes en su diseño al ser impredecible e incierto en el tiempo [1].

En consecuencia, la señal transmitida hacia el receptor es afectada por fenómenos presentes en el canal, tales como la reflexión que ocurre cuando la señal se encuentra con un obstáculo provocando que sea reflejada a su origen. De forma similar, el fenómeno de difracción que crea ondas secundarias, otro fenómeno es la dispersión también conocida como *scattering*.

Cuando la señal atraviesa el canal inalámbrico es afectada por la atenuación que ocasiona pérdida de potencia debido a la distancia, también el ruido que está presente desde la transmisión es una señal no deseada, el desvanecimiento que consiste en la variación de amplitud de una señal a lo largo de tiempo y frecuencia [1].

1.4.2 OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING)

OFDM es considerado como una técnica de multiplexación que es ampliamente usada en muchos estándares de comunicación [6]. Es vista como un caso especial de transmisión multiportadora que emplea subportadoras ortogonales sobrepuestas como se muestra en la Figura 1.1, su idea es dividir el ancho de banda BW disponible en N subportadoras paralelas con el mismo ancho de banda. Cada subportadora es ortogonal al resto y para garantizar esto, la duración del tiempo de símbolo $T[s]$ está dado por la relación escrita en la ecuación 1.1, [7].

$$T = \frac{N}{BW} [s] \quad (1.1)$$

Las subportadoras o también conocidas como subcanales se modulan con un símbolo y se multiplexa en frecuencia, deben ser espaciadas un valor igual o mayor que el ancho de banda de cada subportadora, como se muestra en la ecuación 1.2, [8].

$$\Delta f \geq \frac{BW}{N} [Hz] \quad (1.2)$$

Esta técnica de transmisión al enviar los datos de forma paralela permite reducir los efectos de desvanecimiento por trayectos múltiples, ruido impulsivo y pérdida en la eficiencia del ancho de banda. Es importante notar en la Figura 1.1 la señal que se transmite, a través, de múltiples portadoras transformando el canal de banda ancha a múltiples subcanales de banda estrecha, permitiendo una alta eficiencia de la transmisión y el incremento en el uso del espectro [9].

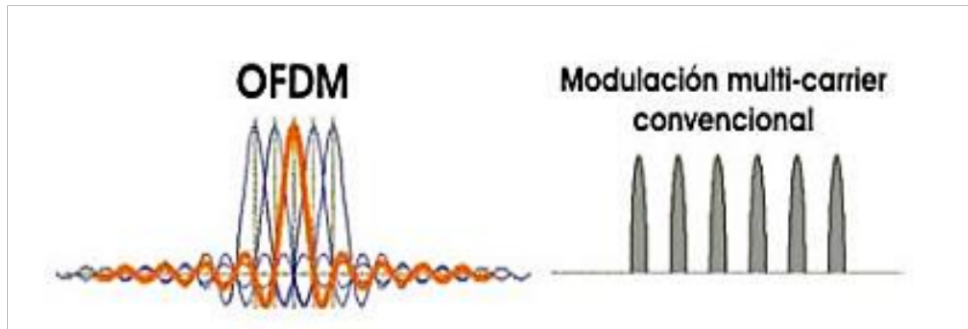


Figura 1.1. OFDM y Multiportadora tradicional [9].

1.4.2.1 Sistema OFDM

OFDM transmite la señal en partes, haciéndola más robusta al desvanecimiento y permitiendo la reconstrucción de la señal de forma simplificada. A continuación, se presenta un diagrama de bloques del sistema OFDM en la Figura 1.2 en el cual se muestra la utilización de procesamiento digital de señales como la transformada rápida de Fourier [10].

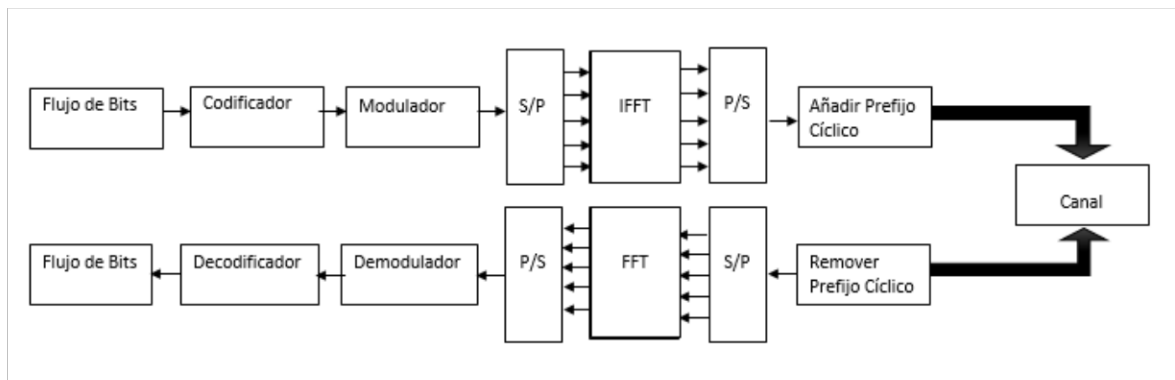


Figura 1.2. Diagrama de bloques de un sistema OFDM.

Para mayor eficiencia, los sistemas OFDM utilizan la IFFT (*Inverse Fast Fourier Transform*) en el transmisor y la FFT (*Fast Fourier Transform*) en el receptor. Como se muestra en la Figura 1.2, la señal que se genera es un flujo de bits en serie, los cuales se codifican para permitir la corrección de errores [9]. Estos datos pasan por un modulador tal como QPSK (*Quadrature Phase Shift Keying*), 16 QAM (*Quadrature Amplitude Modulation*) o 64 QAM también llamado mapeador. A la salida del modulador se obtiene una nueva serie de datos, para luego pasar a un formato en paralelo, ortogonales entre sí, a cada uno de estos se les asignará una subportadora por medio de la IFFT.

La IFFT se encarga de generar una variedad de portadoras con diferentes frecuencias ortogonales siendo capaz de dividir el ancho de banda total para el número N de

subportadoras. Una vez realizada la IFFT los datos son convertidos a un formato en serie, para luego agregar un intervalo de guarda conocido como prefijo cíclico en cada símbolo OFDM, esto se puede observar en la Figura 1.3 con el fin, de reducir la ISI (*Intersymbol Interference*), que es causado por la distorsión de trayectos múltiples [10].

En el canal, la señal está expuesta al ruido y desvanecimiento multitrayecto. Los sistemas OFDM son sensibles a los errores de sincronización, por lo cual existen técnicas para corregir los efectos causados por estos errores, por esta razón se lleva a un proceso de sincronización en el receptor. A continuación, se realiza el proceso contrario el cual consiste en remover el prefijo cíclico del símbolo OFDM para luego realizar la FFT. Finalmente, se demodula y decodifica la serie de datos para obtener los datos originales [10].

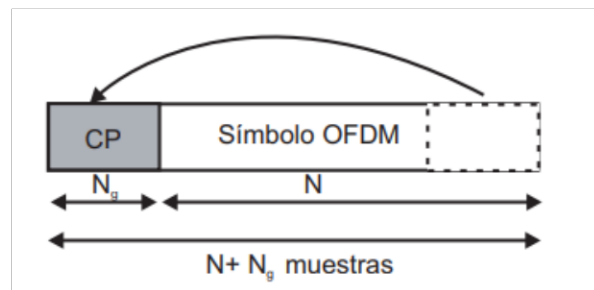


Figura 1.3. Representación del prefijo cíclico [7].

1.4.3 NOMA (NON-ORTHOGONAL MULTIPLE ACCESS)

NOMA es un candidato que promete mejoras en la eficiencia del sistema al contar con un esquema de acceso múltiple, que permite la multiplexación de usuarios desde el dominio de la potencia [11]. Los recursos son utilizados de manera simultánea de forma no ortogonal en una subportadora con niveles de potencia diferentes para realizar la transmisión de datos sobre un mismo recurso de frecuencia al mismo instante de tiempo [2].

Esta técnica de acceso permite la asignación de un canal de frecuencia a múltiples usuarios en un mismo intervalo de tiempo, lo que genera una eficiencia espectral mejorada y un mejor uso de todo el ancho de banda, de esta forma se aprovechan mejor los recursos como se muestra en la siguiente Figura 1.4.

En la Figura 1.4 OMA (*Orthogonal Multiple Access*) asigna varios usuarios con recursos de subportadoras que son ortogonales en el dominio de la frecuencia, de tiempo o código. Por esta razón idealmente no existe interferencia entre múltiples usuarios, pero no siempre puede alcanzar la máxima capacidad en los sistemas inalámbricos multiusuario.

Por otra parte, NOMA hace frente a los problemas que se menciona de OMA, básicamente permite interferencias controlables para la asignación de recursos y una complejidad tolerable en el receptor para lo cual usa SIC (*Successive Interference Cancellation*) que significa cancelación de interferencia sucesiva [12].

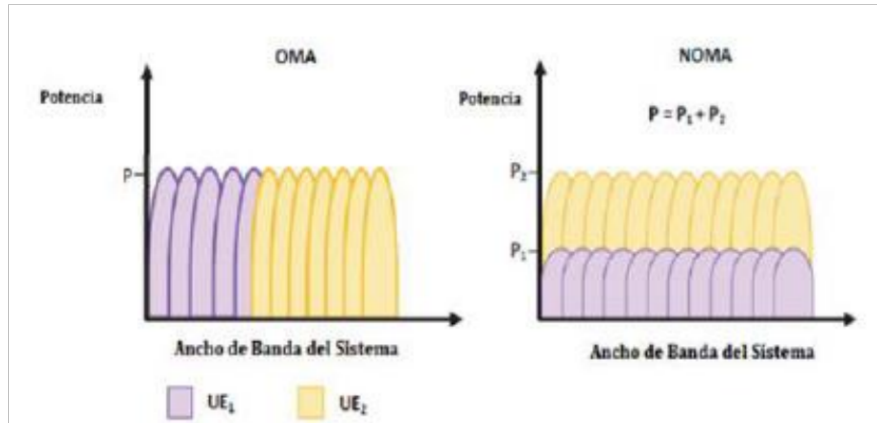


Figura 1.4. Recursos asignados a dos usuarios para técnica de acceso OMA y NOMA [2].

NOMA es una técnica que explota la capacidad del sistema en el dominio de la potencia que no se utilizó de manera eficiente en generaciones anteriores, lo que trae nuevos beneficios como [12], [13]:

- Mayor rendimiento del sistema con una alta mejora en la eficiencia espectral.
- No se requiere la utilización de nuevas antenas de transmisión, lo que beneficia en el tema de costos y limitados espacios para nuevas redes.
- Mediante el uso de los recursos no ortogonales NOMA tiene mayor ventaja sobre OMA al permitir una conectividad masiva.
- Baja latencia de transmisión, al ser simultánea no necesita esperar un intervalo de tiempo para enviar información.
- NOMA presenta una buena compatibilidad con técnicas OMA como OFDMA.

Existen varios desafíos que tienen los sistemas NOMA según Islam et al. [14].

- El emparejamiento dinámico de usuarios, dado que, varios usuarios comparten recursos de tiempo frecuencia y código, se presentan fuertes interferencias en los sistemas NOMA.
- Impacto en la distorsión de transmisión, al transmitir la información por el canal inalámbrico se presentan pérdidas y distorsión de la señal hasta llegar al receptor, para lo cual se presentan soluciones como codificaciones de fuente y codificación de canal para reducir la distorsión.

- Asignación de recursos, se debe, a la complejidad de asignación de recursos y determinación de tiempo para cada usuario, ya que, posee recursos limitados.

NOMA emplea un sistema de comunicación con transmisores y receptores complejos formados por dos capas en el transmisor, que únicamente, se diferencian por el factor de inyección en la segunda capa o capa inferior [2], la capa superior presenta mayor asignación de potencia que la capa inferior. Las capas están formadas de un conjunto de bloques similares, que son configurados de manera independiente. En estas se aplican diferentes modulaciones y tasas de codificación, con la finalidad de proporcionar la robustez necesaria a la señal en la primera capa, con el propósito que la señal llegue sin errores. Mientras que, la segunda capa se enfoca en las prestaciones del canal, con la finalidad de mostrar mejores velocidades en la transmisión [15].

El receptor realiza procesos por separado para la recuperación de la señal, mientras que para la capa superior se recupera la información del primer usuario de manera fácil al realizar el proceso inverso del transmisor, en la capa inferior se realiza un proceso más complejo para retirar la señal del primer usuario y eliminar el factor de inyección que se aplicó en el transmisor, para así, obtener la señal del segundo usuario.

1.4.4 SINCRONIZACIÓN EN NOMA-OFDM

El sistema de comunicación emplea la técnica de acceso NOMA basada en OFDM, para la transmisión de datos que permite enviar la información por subportadoras ortogonales, robusto frente a la distorsión causadas en el canal inalámbrico. Sin embargo, este sistema es sensible a los errores de sincronización debido a la presencia del STO (*Symbol Time Offset*) y CFO (*Carrier Frequency Offset*) [1]. Este componente analiza el rendimiento de la señal que es afectada por STO cuyos efectos serán explicados a continuación.

1.4.4.1 Efectos del STO

En los sistemas NOMA basado en OFDM es necesario la modulación y demodulación con las funciones fundamentales IFFT y FFT, en el transmisor y receptor respectivamente. En el receptor es necesario tener las muestras exactas de la señal transmitida para la duración del símbolo. En otros términos, es necesario realizar la sincronización de tiempo del símbolo para hallar el comienzo del símbolo recibido sin tener presente el prefijo cíclico.

En la Tabla 1.1 se puede observar que el STO de δ muestras afectan a los símbolos recibidos en el dominio del tiempo y frecuencia. STO de δ muestras en el dominio del tiempo provoca un error o desplazamiento de fase en el dominio de la frecuencia. En donde N es el número total de subportadoras y k es el índice de la subportadora [16].

Tabla 1.1. Efectos de STO [16].

	Señal Recibida	STO(δ)
Dominio del tiempo	$y[n]$	$x[n + \delta]$
Dominio de la frecuencia	$Y[k]$	$e^{j2\pi k\delta/N} X[k]$

El efecto del STO puede ser diferente, dependiendo del punto de inicio estimado del símbolo. Se muestra en la Figura 1.5, cuatro casos distintos de desplazamiento de tiempo, dado que el punto de partida estimado es exacto, un poco antes, demasiado pronto y un poco tarde al inicio exacto del símbolo [1].

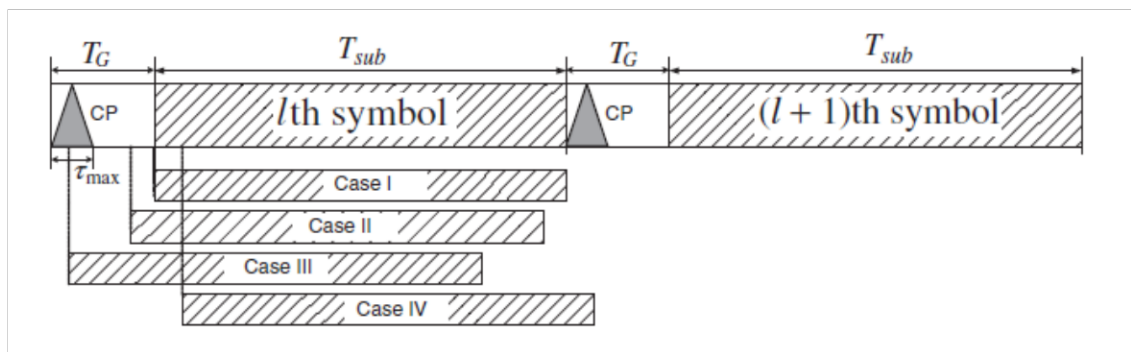


Figura 1.5. Diferentes casos de estimación del inicio del símbolo en presencia de STO [16].

1.4.4.2 Técnicas de estimación de STO

Para contrarrestar los efectos del STO sobre la señal se debe determinar el inicio del símbolo OFDM correctamente, mediante las técnicas de sincronización en el lado del receptor. Por consiguiente, se detallan los tipos de técnicas para la estimación de STO en el dominio del tiempo con la utilización del prefijo cíclico y con símbolos de entrenamiento o preámbulo [16].

Estimación del STO utilizando el prefijo cíclico.

El prefijo cíclico es la réplica de la parte final de los símbolos colocadas al inicio de cada símbolo para eliminar el ISI. Esta se puede utilizar de forma eficiente, no solo como tiempo de guarda, sino también aprovechar su similitud que tiene con los datos efectivos del símbolo, para así, encontrar el inicio del símbolo con precisión. En la Figura 1.6 se observa dos ventanas que tienen el mismo tamaño N_G muestras del símbolo espaciado y N_u de muestra de datos del símbolo. La similitud en las ventanas se verifica continuamente deslizándose entre ellas y al momento que la ventana cubra todo el CP del símbolo, la similitud se maximiza [17].

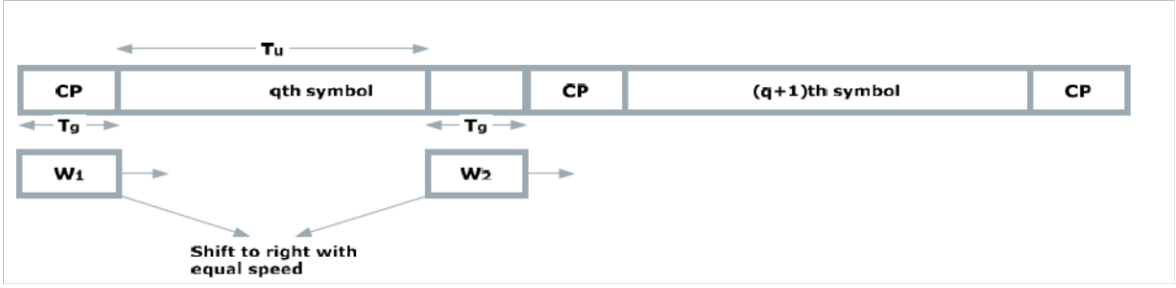


Figura 1.6. Estimación CP mediante el uso de ventanas que indican el inicio del símbolo [17].

El inicio del símbolo se da maximizando o minimizando la correlación. A continuación, se muestra en la ecuación 1.3 y 1.4, respectivamente.

$$\hat{\delta} = \arg_{\delta} \max \left\{ \sum_{j=\delta}^{Ng-1+\delta} |y[n+j] * y^*[n+N+j]| \right\} \quad (1.3)$$

$$\hat{\delta} = \arg_{\delta} \min \left\{ \sum_{j=\delta}^{Ng-1+\delta} |y[n+j] - y[n+N+j]| \right\} \quad (1.4)$$

Estimación del STO utilizando símbolo de entrenamiento

Al usar este método de sincronización, el transmisor incorpora símbolos de entrenamiento a la señal para que sean utilizados al llegar al receptor. Se puede emplear uno o dos símbolos de entrenamiento con una forma repetitiva de periodos $T_{sub}/2$ y $T_{sub}/4$ [16]. En la Figura 1.7 se observa un ejemplo de símbolo de entrenamiento con una forma repetitiva de $T_{sub}/2$. Esta estimación es similar a la anterior con la diferencia que las ventanas ahora maximizan la correlación entre los símbolos de entrenamiento [17].

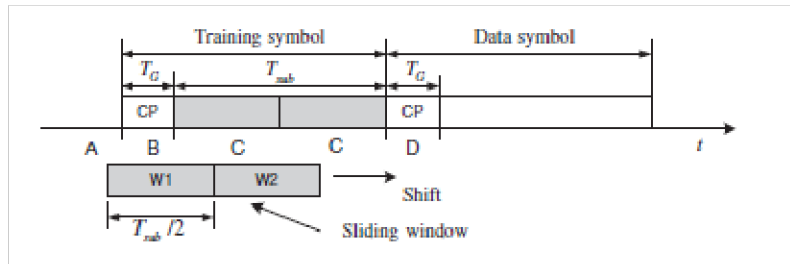


Figura 1.7. Estimación de STO usando el símbolo de entrenamiento repetitivo con periodo $T_{sub}/2$ [16].

De igual forma que la estimación por prefijo cíclico, el STO se puede estimar utilizando la ecuación 1.5, que encuentra la diferencia mínima entre los bloques del preámbulo.

$$\hat{\delta} = \arg_{\delta} \min \left\{ \sum_{j=\delta}^{\frac{N}{4}-1+\delta} \left| y[n+j] - y\left[n + \frac{N}{4} + j\right] \right| \right\} \quad (1.5)$$

2 METODOLOGÍA

En este capítulo se detalla el código realizado para la simulación del sistema NOMA basado en OFDM mediante la utilización del software Matlab. El enfoque utilizado es cuantitativo, ya que, está basado en la medición numérica, es decir que, se puede medir el rendimiento de la técnica por la cantidad de errores del sistema, además, es un planteamiento concreto, ya que, este componente está centrado en la sincronización del STO [18].

Sigue un orden secuencial, es decir, no es posible saltarse algún paso para su desarrollo, este capítulo se realiza luego de la investigación del tema, para así, obtener la simulación del sistema [18]. El sistema de comunicación empleado NOMA-OFDM consta de tres partes principales que son: transmisor, canal de Rician con ruido AWGN y finalmente el receptor con las distintas técnicas de sincronización. Además, se generan las gráficas necesarias para el análisis de los resultados, adicionalmente, se explicarán los comandos utilizados, las variables y funciones necesarias para el desarrollo del programa.

El alcance de la investigación es de tipo mixto, ya que, permite describir un fenómeno como es el STO en la señal y como estimar mediante el uso de las distintas técnicas para su sincronización en el receptor y así medir el BER vs Eb/No. También es explicativo porque se menciona la causa de los fenómenos y explica los resultados obtenidos los cuales se analizan en el siguiente capítulo. Además, es correlacional porque evalúa el grado de relación de dos o más variables en este caso las configuraciones de los distintos parámetros de modulación, tasa de codificación y factor de inyección para el resultado final.

El código desarrollado está formado de tres partes principales que son: el sistema de transmisión NOMA-OFDM, el canal compuesto de desvanecimiento tipo Rician con ruido AWGN y un sistema de recepción compuesto de dos receptores, uno para cada usuario. Esto se representa en el siguiente diagrama de bloques el cual se explica en las siguientes secciones. Es importante mencionar que dentro del receptor existen funciones para añadir, estimar y compensar el STO con la finalidad de simular la sincronización.

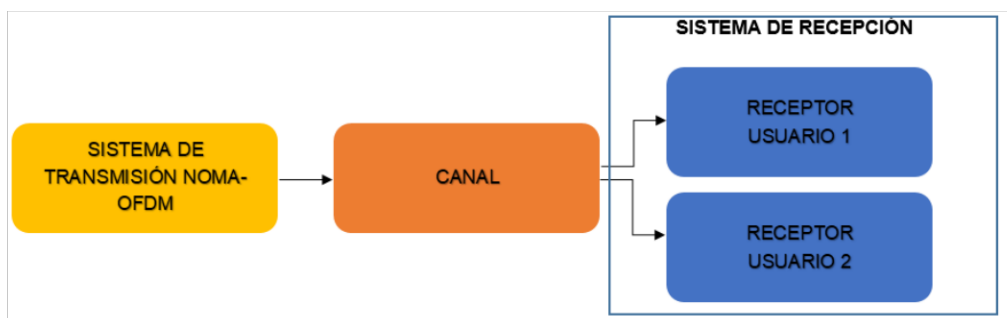


Figura 2.1. Esquema general del código.

2.1 SISTEMA DE TRANSMISIÓN NOMA OFDM

El sistema de transmisión está formado por distintos bloques con funciones específicas que se encargan de procesar la señal con la finalidad de que puedan viajar por el medio de transmisión. Esta parte es la encargada de generar datos aleatorios para cada usuario, codificar y modular de forma independiente, de acuerdo a los requerimientos de cada capa. En la Figura 2.2, se muestra el diagrama de bloques del transmisor NOMA-OFDM.

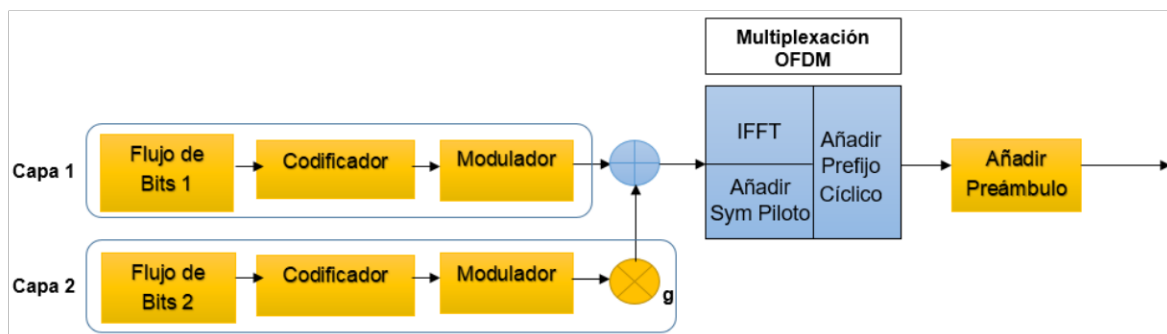


Figura 2.2. Transmisor NOMA-OFDM.

Se puede observar que la diferencia entre las capas radica en, añadir a la señal resultante de la segunda capa un factor de inyección, para luego ser sumada con la señal resultante de la primera capa y después realizar la multiplexación OFDM.

Para la multiplexación OFDM se utilizan parámetros dados por el estándar 802.11p, los mismos que se presentan en la tabla 2.1. Los parámetros para este sistema se configuran de forma independiente en cada capa, es decir, utilizan diferentes tasas de codificación y modulación con distintos factores de inyección.

Tabla 2.1. Parámetros del transmisor.

PARÁMETRO	IEEE 802.11p
Modulación	QPSK, 16 QAM, 64 QAM
Tasa de codificación	1/2, 2/3 Y 3/4
Número de subportadoras	52(48 de datos y 4 pilotos)
Número de subportadoras total	64

A continuación, se muestran los parámetros utilizados dentro de la simulación del transmisor, todos estos parámetros están definidos antes de la simulación Monte-Carlo que es el método para estimar el BER.

Los parámetros de entrada son los datos que se pueden modificar para cada capa dependiendo de las configuraciones que se necesite y los parámetros OFDM son fijos para la transmisión de la información.

```

2   %Parámetros de entrada-----
3 -   iter=40;                %Número de iteraciones simulación
4 -   M1 = 4;                %64 para 64QAM, 16 para 16QAM, 4 para QPSK
5 -   M2 = 16;              %64 para 64QAM, 16 para 16QAM, 4 para QPSK
6 -   k1 = 2/3;             %Tasa de codificación 1/2, 2/3, 3/4
7 -   k2 = 1/2;            %Tasa de codificación 1/2, 2/3, 3/4
8 -   fac_dB = -8;         %Factor de inyección en dB
9 -   EbNo_dB=0:2:40;     %Intervalo para simulación

14  %Parámetros OFDM-----
15 -   n_sym=8;              %Número de símbolos OFDM por paquete
16 -   Nfft=64;             %Tamaño fft
17 -   data_sub=48;         %Número subportadoras de datos
18 -   pilot_sub=4;         %Número subportadoras piloto
19 -   pilotos=repmat([1;-1;1;-1],1,n_sym);

```

Segmento de código 2.1. Parámetros iniciales para el transmisor

Para la transmisión se creó la función *TX_NOMA_OFDM* sus variables de entrada y salida se detallan en la Tabla 2.2, esta función se utiliza para la codificación, modulación, la suma de las señales y la multiplexación OFDM. Es necesario definir las características del codificador y decodificador para cada capa y también generar los bits aleatorios de cada usuario antes de usar la función *TX_NOMA_OFDM* como se observa en el Segmento de código 2.2. En esta sección se describe el proceso para la transmisión de datos.

Tabla 2.2. Variables de entrada y salida para la Transmisión.

Datos	Tipo de Dato	Detalle
convEncoder1	Entrada	Características del codificador conv. para usuario 1
convEncoder2	Entrada	Características del codificador conv. para usuario 2
M1	Entrada	Modulación para usuario 1
M2	Entrada	Modulación para usuario 2
n_sym	Entrada	Número de símbolos OFDM
factor	Entrada	Factor de inyección
pilotos	Entrada	Subportadoras Piloto
data_sub	Entrada	Número de subportadoras de datos
OFDM_mod	Entrada	Modulador OFDM
data_bit1	Entrada	Bits aleatorios para el usuario 1
data_bit2	Entrada	Bits aleatorios para el usuario 2
txSig	Salida	Señal a transmitir

```

txSig = TX_NOMA_OFDM(convEncoder1,convEncoder2,M1,M2,n_sym,factor,pilotos,data_sub,...
    OFDM_mod,data_bit1,data_bit2);%Codificador/Modulador/Suma de usuarios/OFDM

```

Segmento de código 2.2. Procesar la señal con la función *TX_NOMA_OFDM*.

2.1.1 GENERACIÓN DE BITS ALEATORIOS PARA CADA USUARIO

La primera parte del sistema de transmisión es la generación de bits para el usuario, en este caso son dos usuarios. Para generar valores aleatorios se utiliza el comando *randi* el

cual se define con valores entre 0 y 1, y se especifica el tamaño de la matriz de bits que dependen de la tasa de codificación y modulación de cada capa. El cálculo del tamaño que debe tener la matriz de bits aleatorios se muestra en el Segmento de código 2.3. Estos bits aleatorios serán almacenados en las variables *data_bit1* y *data_bit2* como se muestra en el Segmento de código 2.4.

```
42 - num_bits_c1=48*k1*m1*n_sym; %Número de BITS usuario 1
43 - num_bits_c2=48*k2*m2*n_sym; %Número de BITS usuario 2
```

Segmento de código 2.3. Número de bits para cada usuario.

```
50 - data_bit1 = randi([0 1],num_bits_c1,1); %generacion bits usuario 1
51 - data_bit2 = randi([0 1],num_bits_c2,1); %generacion bits usuario 2
```

Segmento de código 2.4. Generación de bits para cada usuario.

Los bits aleatorios generados para cada usuario son enviados a la función *TX_NOMA_OFDM* para realizar el procesamiento de la señal.

2.1.2 CODIFICACIÓN

Para realizar la codificación es necesario definir las características del codificador de cada capa, para esto se utiliza las funciones *conf_cod1* y *conf_cod2* que también define las características del decodificador como se muestra en el Segmento de código 2.5, que se encuentra dentro de la simulación Monte-Carlo. Estas funciones utilizan las tasas de codificación, el codificador y decodificador convolucional de Matlab, los patrones de *puncturing* y los polinomios generadores.

```
48 - [convEncoder1,vitDecoder1]=conf_cod1(k1,traceBack); %Definir las características
49 - [convEncoder2,vitDecoder2]=conf_cod2(k2,traceBack); %Definir las características
```

Segmento de código 2.5. Define las características del codificador y decodificador.

A continuación, se codifica los bits de cada capa, dentro de la función *TX_NOMA_OFDM* que se encarga de realizar el procesamiento de la señal para luego enviarla por el medio de transmisión, la codificación de bits se observa en el Segmento de código 2.6.

```
2 - dataEnc1 = convEncoder1(data_bit1); %codificacion FEC
3 - dataEnc2 = convEncoder2(data_bit2); %codificacion FEC
```

Segmento de código 2.6. Codificación.

2.1.3 MODULACIÓN

De igual manera la modulación se realiza dentro de la función *TX_NOMA_OFDM*, luego de la codificación se procede a modular los bits de cada capa, esto se realiza siguiendo el orden de la Figura 2.2. Como se indica en el Segmento de código 2.7 se modula los bits con las características seleccionadas para cada capa, estas pueden ser QPSK, 16 QAM o 64 QAM, además, se normaliza los símbolos para así obtener una potencia promedio unitaria.

Este proceso se guarda en las variables *data_sym1* que corresponde al usuario 1 y en *data_sym2* corresponde al usuario 2, que son vectores para luego convertirlos a matriz con el comando *reshape* y usarlos en las siguientes etapas.

```
4 - data_sym1 = qammod(dataEnc1,M1,'InputType','bit','UnitAveragePower',true);
5 - data_sym2 = qammod(dataEnc2,M2,'InputType','bit','UnitAveragePower',true);
6 - data_sym_mat1=reshape(data_sym1,data_sub,n_sym); %convierte en matriz
7 - data_sym_mat2=reshape(data_sym2,data_sub,n_sym); %convierte en matriz
```

Segmento de código 2.7. Modulación.

2.1.4 SUMA DE SEÑALES Y USO DEL FACTOR DE INYECCIÓN

La suma de las señales se realiza una vez que los datos de cada capa estén codificados y modulados, adicionalmente la señal de la capa dos se multiplica por un factor de inyección, el cual está definido como *factor*. Esto se puede observar en el Segmento de código 2.8 en donde se detalla la suma y multiplicación por el factor.

```
8 - data_sym_tot=data_sym_mat1+data_sym_mat2*factor; %suma 2 usuarios
```

Segmento de código 2.8. Utilización del factor de inyección y suma de señales.

2.1.5 MULTIPLEXACIÓN OFDM

Para realizar esta etapa, es necesario definir el modulador OFDM de Matlab con ayuda del *Communications Toolbox*, en donde se puede especificar los parámetros como se muestra en el Segmento de código 2.9, esto se define en el código principal antes de realizar la simulación Monte-Carlo. El modulador OFDM de Matlab se encarga de realizar la IFFT, añadir el prefijo cíclico, añadir las subportadoras piloto e insertar las subportadoras null.

```

24 %Modulador OFDM-----
25 - OFDM_mod = comm.OFDMModulator('FFTLength',64, ...
26                               'PilotInputPort',true, ...
27                               'PilotCarrierIndices',[12; 26; 40; 54], ...
28                               'InsertDCNull',true, ...
29                               'CyclicPrefixLength',16, ...
30                               'NumSymbols',n_sym);

```

Segmento de código 2.9. Modulador OFDM.

Luego de sumar las señales, se aplica la multiplexación OFDM con el comando *step* este comando le indica que el modulador OFDM de Matlab se debe aplicar a la señal resultante *data_sym_tot* y se envía los pilotos como se indica en el Segmento de código 2.10, todo esto se realiza dentro de la función *TX_NOMA_OFDM*.

```

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

```

Segmento de código 2.10. Uso del modulador OFDM de Matlab.

2.1.6 PREÁMBULO

La utilización del preámbulo es necesario para el desarrollo de este trabajo, ya que, se utiliza para la sincronización, al ser el mismo tanto para el transmisor como en el receptor, se hace uso de la función *get_pream* que tiene como entrada el número de subportadoras totales, para la obtención del preámbulo y luego unirlo a la señal que se va a transmitir, dada por la función *TX_NOMA_OFDM* que está almacenada en la variable *txSig*. Como se puede observar en el Segmento del código 2.11 el preámbulo se coloca al inicio de la señal.

```

52 - txSig = TX_NOMA_OFDM(convEncoder1,convEncoder2,M1,M2,n_sym,factor,pilotos,data_sub,OFDM_mod,data_bit1,data_bit2);
53 - [pream]=get_pream(Nfft); %Creación del preámbulo de la forma A A-A-A
54 - pream_TX = pream;
55 - length_pream=length(pream_TX); %Tamaño del prambulo
56 - txSig=[pream_TX ;txSig]; %Unir preámbulo y datos

```

Segmento de código 2.11. Creación y unión del preámbulo con la señal.

En la función *get_pream* el preámbulo se forma por la secuencia de ruido PN (*pseudo noise sequence*) que se transmite en las frecuencias pares, mientras que, en las frecuencias impares se envían ceros, con la finalidad de formar un símbolo con sus dos mitades iguales conocidas como A. En la variable *PN_seq* se utiliza una misma secuencia de ruido PN para realizar las pruebas con un mismo preámbulo, también se multiplica por un valor de $2\sqrt{2}$ con el objetivo de mantener la energía de la señal lo más constante posible.

Adicionalmente, se realiza un reordenamiento y luego se transforma al dominio del tiempo utilizando la IFFT. Al tener una estructura repetitiva se añade un tiempo de guarda compuesta de ceros con el fin de evitar un rango de valores donde la estimación del STO

pueda ser incierta. Para mejorar la técnica de estimación se duplica el símbolo con el signo negativo y así obtener una estructura como se observa en la Figura 2.3.

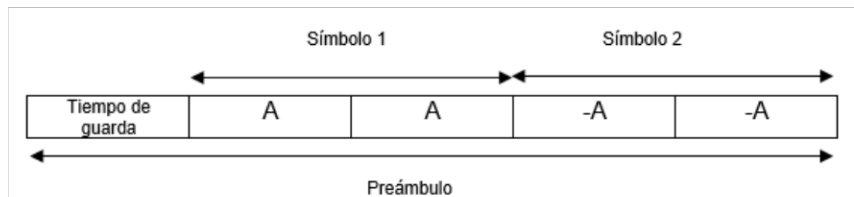


Figura 2.3. Forma final del preámbulo.

La creación del preámbulo se puede observar en el Segmento de código 2.12.

```
function [pream]=get_pream(NsubT)
%Secuencia de Pseudo Ruido
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 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];

PN_seq=PN_seq*2*sqrt(2); %Mantener Energía
PN_seq(2:2:end)=0; %Envío de ceros en frecuencias pares
%Mapeo de subportadoras
PN_seq_ord=zeros(NsubT,1); %Inicializar variable
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, NsubT); %Dominio del tiempo
T_guarda=0*prel(33:64); %Tiempo de guarda (TG)
[pream]=[T_guarda;prel;-prel];%Preámbulo de la forma [TG A A -A -A]
```

Segmento de código 2.12. Función para obtener el preámbulo.

2.2 CANAL

Este trabajo utiliza un canal tipo Rician con ruido AWGN, que simulan el medio de transmisión por donde se envía la información, además, se considera que la señal que envía el transmisor sufre un desvanecimiento diferente al llegar al receptor de cada usuario, como se muestra en la Figura 2.4.

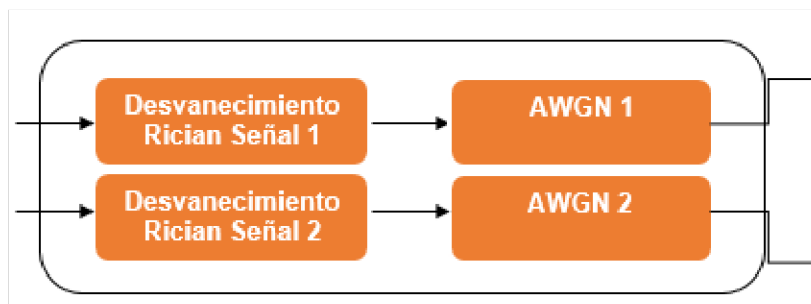


Figura 2.4. Canal.

Para la simulación del canal tipo Rician, se define un canal con la función *Channel()*, en el código principal antes de la simulación Monte-Carlo, como se observa en el Segmento de código 2.13, esta función permite definir ciertas variables para el canal conocido como PDP (*Power Delay Profile*), con la ayuda de Matlab usando *comm.RicianChannel* se crea el canal y luego se calcula el *sampIdx*.

```

33 %Generar canal-----
34 - [canal_fading1,sampIdx1]=Channel(); %Genera canal 1 con función
35 - [canal_fading2,sampIdx2]=Channel(); %Genera canal 2 con función

```

Segmento de código 2.13. Generar el canal.

La función del canal se puede observar en el Segmento de código 2.14, en donde se definen las variables para un canal Rural LOS; este es un escenario sin construcciones en un espacio abierto, existe línea de vista entre el transmisor y receptor. En el canal Rician se tiene un valor *k* que define la potencia de la señal con línea de vista, se considera que no existe frecuencia *doppler* por esta razón *fd* es cero y se define la frecuencia de muestreo que es 10 MHz por el estándar 802.11p.

En el código principal, después de unir la señal con el preámbulo en la variable *txSig*, se resetea el objeto del canal en este caso *canal_fading1* con el comando *reset*, con la finalidad que los coeficientes del canal no se queden fijos, para que sean aleatorio y vaya cambiando el estado de canal como se puede observar en el Segmento de código 2.15.

```

function [ canal,sampIdx ] = Channel()
%Establecer parámetros
%Potencia en dB de cada rayo.
%Delay Retardo de cada rayo.
Pot = [0 -14 -17];
Delay = [0 100 200]*1e-9;
K=20;
% 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 2.14. Función *Channel()*.

En la siguiente línea del Segmento de código 2.15 se aplica el canal con el comando *step* enviando el canal que se define en *canal_fading1* y la señal que se va transmitir *txSig*. El comando devuelve *y1* que es la señal resultante al aplicar el canal y el *pathgains1* que tiene las ganancias que se aplicó al canal. Luego se añade ruido AWGN con el comando *awgn* que tiene como entrada la señal después de atravesar el canal de Rician, y un valor de SNR para cada iteración y con el parámetro *measured* se mide la potencia de la señal antes de sumar el ruido. Este proceso se repite para la misma señal *txSig* por segunda vez para así obtener un diferente desvanecimiento en el receptor del usuario 2.

```

57 %*****CANAL*****
58 - reset (canal_fading1); %Resetear el objeto canal 1
59 - [y1,pathgains1] = step (canal_fading1,txSig); %Filtrar la señal con canal 1
60 - rxSig1_con_desv=awgn (y1,SNR_dB (i), 'measured'); %Aplicar ruido a la señal 1
61 - reset (canal_fading2); %Resetear el objeto canal 2
62 - [y2,pathgains2] = step (canal_fading2,txSig); %Filtrar la señal con canal 2
63 - rxSig2_con_desv=awgn (y2,SNR_dB (i), 'measured'); %Aplicar ruido a la señal 2

```

Segmento de código 2.15. Canal Rician y ruido AWGN.

2.3 SISTEMA DE RECEPCIÓN NOMA OFDM

El receptor una vez que recibe la señal realiza el proceso inverso del transmisor con la diferencia que se añaden los bloques de sincronización y ecualización necesarios para recuperar la señal, como se observa en el siguiente diagrama de bloques en la Figura 2.5.

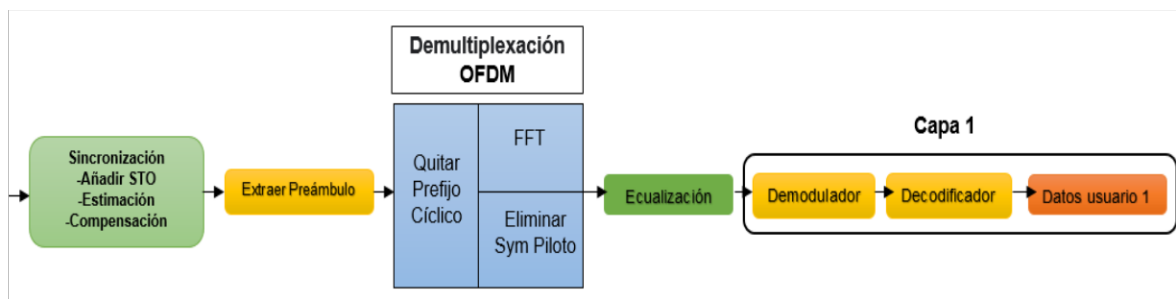


Figura 2.5. Receptor para el Usuario 1.

En la Figura 2.6 se observa el receptor para el usuario 2, este sistema de recepción realiza la recuperación de la señal adicionando un proceso en la capa dos para eliminar la información del usuario 1, cuenta con el bloque de sincronización y ecualización al igual que el receptor del usuario 1.

Este componente se centra en el desarrollo principalmente de la sincronización, por esta razón, se definen variables y valores de STO que se agrega a la señal. También se declara la variable *tec_STO* con el objetivo de escoger la técnica de sincronización que desee usar.

Como se muestra en el Segmento de código 2.16 estas variables son declaradas en el código principal antes de la simulación Monte-Carlo.

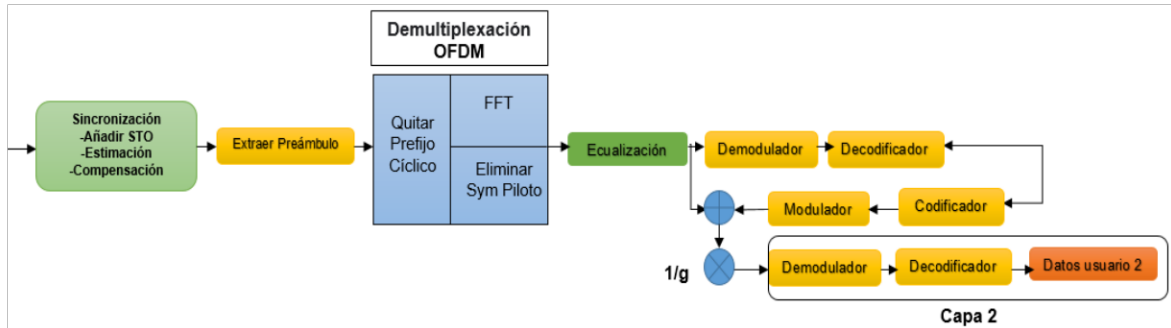


Figura 2.6. Receptor para el Usuario 2.

```

20 %Datos para el STO-----
21 - nSTO = 5; %Desplazamiento STO
22 - tec_STO = 1; %Técnica usando PC (1) Técnica usando preámbulo (2)
23 - con_sin_sync=0; %Con Sincronización (0) sin sincronización (1)

```

Segmento de código 2.16. Datos de la sincronización STO.

Para realizar este proceso de sincronización, se creó una función con el nombre de *synch*, con la finalidad de obtener la señal sincronizada en el receptor del usuario 1 almacenada en la variable *rxSig1_add_est_com* y del usuario 2 almacenada en la variable *rxSig2_add_est_com*, esta etapa se encuentra dentro de una condición *if* con la finalidad de usar o no las técnicas de sincronización. En la Tabla 2.3 se indican las variables de entrada y salida de la función para el receptor del primer usuario, esto se repite para el segundo usuario, por ello, únicamente se modifica el nombre de las variables que tienen 1 por el número 2.

Tabla 2.3. Variables de entrada y salida para la sincronización.

Dato	Tipo de Dato	Detalle
rxSig1_con_desv	Entrada	Señal a través del canal
nSTO	Entrada	Número de muestras STO
tec_STO	Entrada	Técnicas de estimación STO
Nfft	Entrada	Número total de subportadoras
n_sym	Entrada	Número de símbolos OFDM
length_pream	Entrada	Longitud del preámbulo
rxSig1_add_est_com	Salida	Señal sincronizada

A continuación, en el Segmento de código 2.17, se observa el funcionamiento del condicional *if*, si la variable *con_sin_sync* es igual cero, sincroniza la señal de cada receptor

y en caso que el valor de la variable *con_sin_sync* sea diferente de cero, asume sincronización perfecta, es decir, no es necesario estimar ni compensar la señal, ya que, no se le añade el STO a la señal, esto se realiza con la finalidad de comparar los resultados.

```

65 | %-----SINCRONIZACION-----
66 - | if con_sin_sync ==0
67 |     %CON SINCRONIZACION
68 - |     rxSig1_add_est_com=synch(rxSig1_con_desv,nSTO,tec_STO,Nfft,n_sym,length_pream); %Insertar/Estimar/Com
69 - |     rxSig2_add_est_com=synch(rxSig2_con_desv,nSTO,tec_STO,Nfft,n_sym,length_pream); %Insertar/Estimar/Com
70 - |     estado_sig1= rxSig1_add_est_com;
71 - |     estado_sig2= rxSig2_add_est_com;
72 - | else
73 |     %SIN SINCRONIZACION
74 - |     estado_sig1= rxSig1_con_desv;
75 - |     estado_sig2= rxSig2_con_desv;
76 - | end

```

Segmento de código 2.17. Sincronización STO.

2.3.1 SINCRONIZACIÓN NOMA-OFDM

Al llegar la señal al receptor se realiza la sincronización, la cual está compuesta de tres etapas que son: la inserción, estimación y compensación del STO que se encuentran dentro de la función *synch* como se puede observar en el Segmento de código 2.18.

```

function rxSig1_add_est_com=synch(rxSig_con_desv,nSTO,tec_STO,Nfft,n_sym,length_pream)
%-----Añadir STO a la señal con desvanecimiento-----
rxSig_STO=add_STO(rxSig_con_desv,nSTO); %Insercion de STO a la señal
%-----Técnica de sincronización-----
STO_est=tec_syn(tec_STO,rxSig_STO,Nfft,n_sym,length_pream);%Estimacion STO
%-----Compensación-----
Sym_STO_comp= STO_comp(rxSig_STO,STO_est,Nfft,n_sym,length_pream);
rxSig1_add_est_com=reshape(Sym_STO_comp,[length(Sym_STO_comp),1]); %convertimos a vector

```

Segmento de código 2.18. Función *synch*.

2.3.1.1 Inserción, Estimación y Compensación del STO

Una vez que la señal llega al receptor de cada usuario se le añade el efecto STO para simular un entorno real, para esto se necesita insertar a la señal un desplazamiento *nSTO* que consiste en añadir ceros al inicio o al final, dependiendo del valor de *nSTO* si es mayor que cero implica que se atrasa con respecto al punto de inicio del símbolo y si es menor que cero indica que está adelantado y en el caso que *nSTO* sea igual a cero indica que coincide con el inicio del símbolo.

La inserción, estimación y compensación se realiza dentro de la función *synch*, en donde se utiliza la función *add_STO* para la inserción del STO, que tiene como variables de entrada la señal después de atravesar el canal y el desplazamiento STO como se observa en el Segmento de código 2.19. La inserción del STO se puede ver en el Segmento de código 2.20.

```

2 | %-----Añadir STO a la señal con desvanecimiento-----
3 | rxSig_STO=add_STO(rxSig_con_desv,nSTO); %Insercion de STO a la señal

```

Segmento de código 2.19. Inserción del STO.

```

function rxSig_STO=add_STO(rxSig, nSTO)
% add STO (symbol time offset)
% rxSig : Received signal
% nSTO : Number of samples corresponding to STO
rxSig=reshape(rxSig,[1,length(rxSig)]);
if nSTO>=0
    rxSig_STO=[rxSig(nSTO+1:end) zeros(1,nSTO)]; % advance
else
    rxSig_STO=[zeros(1,-nSTO) rxSig(1:end+nSTO)]; % delay
end

```

Segmento de código 2.20. Función de Inserción del STO.

En la variable *rxSig_STO* se tiene la señal con STO, la cual se usa para realizar la estimación STO con las técnicas estudiadas en el capítulo anterior de las cuales se presentan en la Tabla 2.4 las técnicas implementadas en este trabajo.

Tabla 2.4. Técnicas de estimación STO

Nº	Tipo de Técnica
1	Estimación del STO con la diferencia mínima del Prefijo Cíclico
2	Estimación del STO con la diferencia mínima del preámbulo

Para implementar la estimación del STO se creó la función *Tec_syn* la cual contiene las dos técnicas de estimación como se muestra en el Segmento de código 2.21, las variables de entrada y salida de la función se detallan en la Tabla 2.5 y el contenido de la función *Tec_syn* se muestra en el Segmento de código 2.22.

```

4 | %-----Técnica de sincronización-----
5 | STO_est=Tec_syn(tec_STO,rxSig_STO,Nfft,n_sym,length_pream);%Estimacion STO

```

Segmento de código 2.21. Técnica de sincronización.

Tabla 2.5. Variables de entrada y salida para la sincronización.

Dato	Tipo de Dato	Detalle
Tec_STO	Entrada	Técnica de estimación STO
rxSig_STO	Entrada	Señal con STO
Nfft	Entrada	Número total de subportadoras
n_sym	Entrada	Número de símbolos
length_pream	Entrada	Longitud de preámbulo.
STO_est	Salida	Valor estimado


```

function STO_est = Tec_syn(tec_STO,rxSig_STO,Nfft,n_sym,length_pream)
rxSig_STO=reshape(rxSig_STO,[1,length(rxSig_STO)]);
switch tec_STO
case 1
    %Técnica de estimación el Prefijo Cíclico
    STO_est = STO_diferencia_PC(rxSig_STO,Nfft,n_sym);
case 2
    %Técnica de estimación el preámbulo
    STO_est = STO_diferencia_pream(rxSig_STO,length_pream);
end
end

```

Segmento de código 2.22. Función *Tec_syn* con las técnicas de sincronización.

Se implementaron dos técnicas de estimación mediante el uso de prefijo cíclico y del preámbulo, estas realizan una estimación en el dominio del tiempo. La primera técnica está basada en el prefijo cíclico la cual utiliza la ecuación 1.4 del Capítulo 1. Esta ecuación estima el STO mediante la mínima diferencia entre dos ventanas, por esta razón, se crea la función *STO_diferencia_PC* la cual se encarga de encontrar la mínima diferencia entre el prefijo cíclico y su símbolo.

En la función *STO_diferencia_PC*, se define distintas variables las cuales empiezan con el tamaño de la ventana W el cual tiene la misma longitud que el prefijo cíclico y se desplazará todo el tamaño del símbolo N_{ofdm} , es importante considerar que existe el preámbulo al inicio de la señal, así que, la ventana debe iniciar casi al final del preámbulo esto se indica en la variable *com_delay*. Para mejorar los resultados de la técnica se encuentra la máxima similitud entre varios prefijos cíclicos realizando la comparación de todos los símbolos y no solo al inicio, por esta razón se crea la variable *Num_CP* que se obtiene del número de símbolos menos uno, para que, al momento de comparar, la longitud que recorre la ventana no sea mayor que el número de símbolos. En la Figura 2.7 se puede observar de manera gráfica lo que realiza el Segmento de código 2.23.

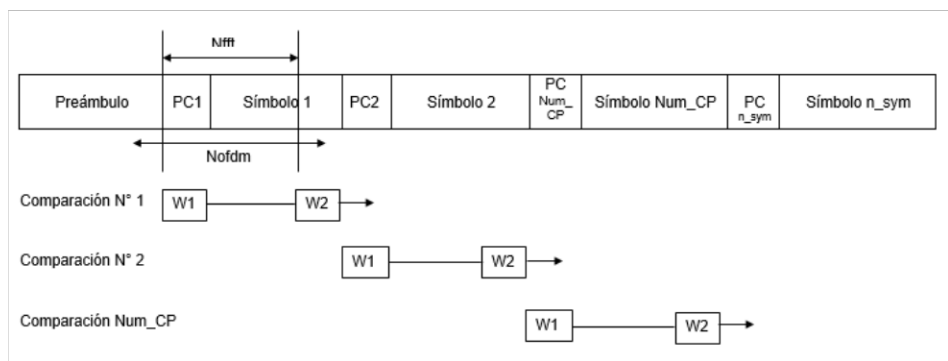


Figura 2.7. Estimación STO usando el prefijo cíclico.

```

function STO_est= STO_diferencia_PC(rxSig_STO,Nfft,n_sym)
%Técnica de estimación mínima diferencia utilizando Prefijo Cíclico
W=Nfft/4;           %Tamaño de ventana igual al PC
Nofdm=Nfft+W;      %Longitud que corre la ventana
com_delay=140;     %Punto de partida de la ventana para evitar el preámbulo
Num_CP=n_sym-1;   %Número de comparaciones
%Inicializacion
Sum=zeros(Num_CP,Nofdm);
Sum_t=zeros(1,Nofdm);
m = 0;
while m<Num_CP-1 %Repite para comparar el prefijo cíclico de cada simbolo
    %Desplazamiento de ventana
    for i=1:Nofdm
        nn=i+com_delay+(0:W-1);
        tmp=rxSig_STO(nn+m*Nofdm)-rxSig_STO(nn+Nfft+m*Nofdm);
        Sum(m+1,i) = sum(abs(tmp)); %ecuación 1.4 Diferencia mínima
    end
    Sum_t=Sum(m+1,:)+Sum_t;       %Máxima similitud
    m=m+1;
end
[Mag, Pos]=min(Sum_t);          %Valor y posición del mínimo valor
STO_est = -Pos+21;              %Estimación STO
end

```

Segmento de código 2.23. *Función STO_diferencia_PC.*

La segunda técnica implementada en este componente es la estimación basada en el preámbulo que utiliza la ecuación 1.5 del capítulo anterior. Esta técnica realiza la estimación STO mediante la mínima diferencia entre símbolos iguales dentro del preámbulo. Por esta razón, se crea la función *STO_diferencia_pream* con las siguientes variables, W el tamaño de la ventana, en donde, se divide la longitud del preámbulo entre cinco, ya que, el preámbulo está formado de cuatro símbolos repetitivos y un tiempo de guarda, para el desplazamiento se utiliza $Nofdm$ que es el tamaño de la ventana más la mitad de la misma ventana.

El inicio de la ventana tiene un desplazamiento el cual inicia dentro del tiempo de guarda, para esto, se usa la variable *com_delay*. Al tener cuatro símbolos repetitivos únicamente se necesitan dos comparaciones como indica la Figura 2.8 en el cual se puede observar de forma gráfica lo que realiza la función de estimación usando el preámbulo en el Segmento de código 2.24.

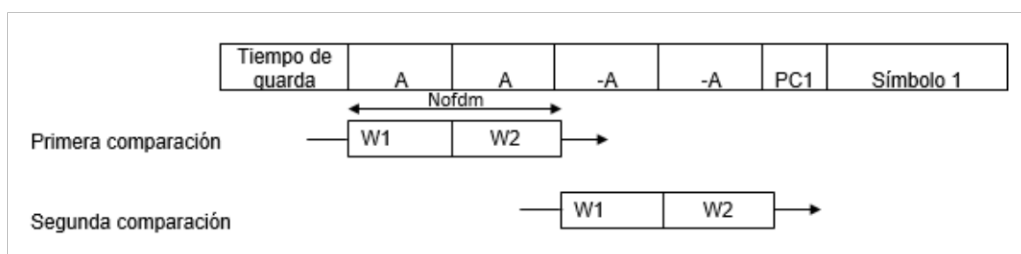


Figura 2.8. Estimación STO usando el preámbulo.

```

function STO_est=STO_diferencia_pream(rxSig_STO,length_pream)
%Técnica de estimación mínima diferencia utilizando Preámbulo
W=length_pream/5;      %tamaño de la ventana 32
Nofdm=W+W/2;          %Logitud que recorre 48
com_delay=Nofdm/4;     %Punto de partida de la ventana

%Inicializacion
Sum =zeros(2,Nofdm);
Sum_t= zeros(1,Nofdm);
STO_est=0;

for m=0:1      %Comparación de los simbolos repetitivos del preámbulo A&-A
    %Desplazamiento de ventanas
    for i=1:Nofdm
        nn=i+com_delay+(0:W-1);
        tmp =rxSig_STO(nn+2*W*m)-rxSig_STO(nn+W+2*W*m);
        Sum(m+1,i)=sum(abs(tmp)); %ecuacion 1.5 Diferencia mínima
    end
    Sum_t = Sum(m+1,:)+Sum_t;      %Máxima similitud
end
[Mag,Pos]=min(Sum_t);             %Valor y posición del mínimo valor
STO_est = -Pos+21;                %Estimación STO
end

```

Segmento de código 2.24. Función *STO_diferencia_pream*.

Luego de estimar el STO, se realiza la compensación del desplazamiento, que consiste en la operación contraria de la inserción del STO para esto se utiliza la función *STO_comp* como se indica en el segmento de código 2.25. Adicional, cuando STO es negativo se copia el prefijo cíclico del último símbolo con el objetivo de mejorar el valor del BER y cuando es mayor o igual a cero, no es necesario ya que la compensación se realiza al preámbulo el cual no influye en los resultados porque se extrae luego de realizar la sincronización esto se puede observar en el Segmento de código 2.26.

```

6  %-----Compensación-----
7 - Sym_STO_comp= STO_comp(rxSig_STO,STO_est,Nfft,n_sym,length_pream);
8 - rxSig1_add_est_com=reshape(Sym_STO_comp,[length(Sym_STO_comp),1]); %convertimos a vector

```

Segmento de código 2.25. Compensación STO.

```

function Sym_STO_comp= STO_comp(rxSig_STO,STO_est,Nfft,n_sym,length_pream)
if STO_est >= 0
    Sym_STO_comp=[zeros(1,STO_est) rxSig_STO(1:end-STO_est)];
else
    Sym_STO_comp=[rxSig_STO(-(STO_est-1):end) zeros(1,-STO_est)];
end
if STO_est <0
    W=Nfft/4;          %Longitud del prefijo cíclico
    Nofdm=Nfft+W;     %Longitud del simbolo
    Pc_Nsym=length_pream+Nofdm*(n_sym-1);%Posición del último PC
    %Símbolo OFDM transmitido
    nn=Pc_Nsym+(W+STO_est+1:W);
    Sym_STO_comp(nn+Nfft)=Sym_STO_comp(nn);
end

```

Segmento de código 2.26. Función *STO_comp*.

2.3.2 EXTRAER PREÁMBULO

Una vez que se obtiene la señal sincronizada en cada receptor se realiza el proceso inverso del transmisor, para ello, se creó la función *RX1_NOMA_OFDM* que se encarga de realizar la extracción del preámbulo, el proceso inverso OFDM y la ecualización. Esta función se encuentra dentro del código principal como se muestra en el segmento de código 2.27 para el primer usuario y en el segmento de código 2.28 para el segundo usuario.

```
77 %USUARIO 1
78 - rx_data = RX1_NOMA_OFDM(OFDM_demod,estado_sig1,n_sym,pathgains1,sampIdx1,data_sub,length_pream);
```

Segmento de código 2.27. Función *RX1_NOMA_OFDM* para el usuario 1.

```
84 %USUARIO 2
85 - rx_data2 = RX1_NOMA_OFDM(OFDM_demod,estado_sig2,n_sym,pathgains2,sampIdx2,data_sub,length_pream);
```

Segmento de código 2.28. Función *RX1_NOMA_OFDM* para el usuario 2.

En esta etapa se elimina el preámbulo de la señal que se encuentra en la variable *estado_sig1* y *estado_sig2*. En el Segmento de código 2.29 se puede ver como se retira el preámbulo, esto se realiza dentro de la función *RX1_NOMA_OFDM* sus variables de entrada y salida se detallan en la Tabla 2.6 para el usuario 1 y este mismo proceso se realiza para el otro usuario, se reemplaza las variables que tienen número 1 por el 2.

Tabla 2.6. Variables de entrada y salida para la función *RX1_NOMA_OFDM*.

Datos	Tipo de Dato	Detalle
OFDM_demod	Entrada	Demodulador OFDM
estado_sig1	Entrada	Señal sincronizada
n_sym	Entrada	Número de símbolos
pathgains1	Entrada	Ganancias que se aplican al canal
sampIdx1	Entrada	Posición de los rayos
Data_sub	Entrada	Subportadora de datos
length_pream	Entrada	Tamaño del preámbulo
rx_data	Salida	Señal ecualizada

```
2 - rxSig=rxSig((length_pream +1):end); %extracción del preámbulo
```

Segmento de código 2.29. Retirar el preámbulo de la señal.

2.3.3 DEMULTIPLEXACIÓN OFDM

Para realizar este paso es necesario definir el demodulador OFDM de Matlab en el código principal como se muestra en el Segmento de código 2.30.

```
31 %Demodulador OFDM-----
32 - OFDM_demod = comm.OFDMDemodulator(OFDM_mod);
```

Segmento de código 2.30. Definir demodulador OFDM.

Dentro de la función *RX1_NOMA_OFDM* con el comando *step* se envía la señal libre del preámbulo a realizar la demodulación OFDM como se muestra en el Segmento de código 2.31, en donde se obtiene la información demultiplexada con los parámetros del estándar 802.11p.

```
rx_data_mat= step(OFDM_demod,rxSig); %proceso inverso OFDM
```

Segmento de código 2.31. Proceso inverso OFDM.

2.3.4 ECUALIZACIÓN

La última etapa dentro de la función *RX1_NOMA_OFDM* es la ecualización que es necesaria, ya que, la señal al atravesar el canal de Rician se desvanece, y es necesario ecualizar para ello se utiliza la función *one_tap* como se muestra en el Segmento de código 2.32.

```
4 - rx_data_eq=one_tap(rx_data_mat,n_sym,pathgains,sampIdx); %ecualización
5 - rx_data=reshape(rx_data_eq,n_sym*data_sub,1); % se convierte a vector
```

Segmento de código 2.32. Ecualización.

El ecualizador de canal *one_tap* es el más sencillo en OFDM, para utilizar esta función se necesita el estado del canal, el *pathgains* representa la respuesta del canal en el dominio del tiempo y cada valor del *pathgains* representa la ganancia de cada rayo. Para realizar la ecualización es necesario tener la información en el dominio de la frecuencia por esa razón se utiliza un *for* para recorrer todos los símbolos esto se usa en un canal variante en el tiempo, pero al ser un canal fijo solo es necesario hacerlo una vez para convertirlo al dominio de la frecuencia con la FFT. Como se muestra en el Segmento de código 2.33 se realiza un reordenamiento y saca las subportadoras de datos. Finalmente se toma la matriz de símbolos de datos recibidos y se divide por el estado del canal que se encuentra en la variable *H_eq*.

```
function data_eq=one_tap(rx_data_mat,n_sym,pathgains,sampIdx)
H=zeros(64,n_sym);
for i=1:n_sym
temp=pathgains((i-1)*80+17,:);
temp2=zeros(1,64);
temp2(sampIdx)=temp;
H(:,i)=fft(temp2); %se gener la respuesta de frecuencia de cada simbolo OFDM transmitido
end
H_eq=[H(39:43,:);H(45:57,:);H(59:64,:);H(2:7,:);H(9:21,:);H(23:27,:)]; %reorganizar orden por como
%se forma simbolo OFDM en transmisor
rx_data_mat=rx_data_mat(:,1:end);
rx_data_eq=rx_data_mat./H_eq; %ecualizacion one-tap
data_eq=reshape(rx_data_eq,48*n_sym,1);
```

Segmento de código 2.33. Función *one_tap*.

2.3.5 CAPA UNO

En la primera capa la recuperación de la señal del usuario se realiza mediante un proceso común el cual consiste en demodular y decodificar la señal para obtener la información, esto gracias a que la señal de la capa dos es considerada como ruido.

Una vez que se tiene la señal ecualizada en la variable *rx_data* se realiza la demodulación dependiendo de la modulación usada en el transmisor con el comando *qamdemod* para luego emplear la decodificación como se observa en el Segmento de código 2.34.

```
79 - rx_bit1 = qamdemod(rx_data,M1,'OutputType','bit','UnitAveragePower',true);
80 - rx_bit_dec1 = vitDecoder1(rx_bit1); %decodificación FEC
81 - [number1,ratio1] = biterr(data_bit1(1:end-traceBack),rx_bit_dec1((traceBack+1:end))); %calculo BER
82 - BER1(i)=BER1(i)+ratio1; %BER acumulativo usuario 1
```

Segmento de código 2.34. Recuperación de información para la capa uno

2.3.6 CAPA DOS

En la segunda capa el proceso para la recuperación de la señal es más complejo, la señal luego de la ecualización está guardada en la variable *rx_data2* en donde se realiza un proceso de demodulación y decodificación para lo cual se define nuevamente las características del codificador y decodificador para usarlo posteriormente.

Una vez que se demodula y decodifica la señal se la codifica y modula nuevamente para obtener los datos del usuario 1 almacenados en la variable *sig_tx_1* esta señal se cancela con la señal *rx_data2*, luego se quita el factor de inyección que se agregó en la capa dos, de tal forma se obtiene la señal del usuario 2 almacenada en la variable *sig_tx_2*, como se indica en el Segmento de código 2.35. Finalmente, se realiza la demodulación y decodificación de la señal.

```
84 - %USUARIO 2
85 - rx_data2 = RX1_NOMA_OFDM(OFDM_demod,estado_sig2,n_sym,pathgains2,sampIdx2,data_sub,length_pream); %
86 - [convEncoder1,vitDecoder1]=conf_cod1(k1,traceBack);
87 - rx_bit1 = qamdemod(rx_data2,M1,'OutputType','bit','UnitAveragePower',true);
88 - rx_bit_dec1 = vitDecoder1(rx_bit1); %decodificación FEC
89 - [convEncoder1,vitDecoder1]=conf_cod1(k1,traceBack);
90 - rx_bit_cod1 = convEncoder1([rx_bit_dec1(traceBack+1:end);zeros(traceBack,1)]); %codificación FEC
91 - sig_tx_1 = qammod(rx_bit_cod1,M1,'InputType','bit','UnitAveragePower',true); %Modulación
92 - sig_tx_2=(rx_data2-sig_tx_1)./factor; %resta de señales
93 - rx_bit2 = qamdemod(sig_tx_2,M2,'OutputType','bit','UnitAveragePower',true);
94 - rx_bit_dec2 = vitDecoder2(rx_bit2); %decodificación FEC
95 - [number2,ratio2] = biterr(data_bit2(1:end-2*traceBack),rx_bit_dec2((traceBack+1:end-traceBack)));
96 - BER2(i)=BER2(i)+ratio2; %BER acumulativo usuario 2
```

Segmento de código 2.35. Recuperación de información para la capa dos.

2.4 CÁLCULO DEL BER Y GENERACIÓN DE GRÁFICAS

Para el cálculo del BER se realiza una comparación, entre la información enviada del transmisor y la señal recibida en la capa uno y dos, esta comparación indica la cantidad de errores obtenidos al realizar la transmisión. Para esto se utiliza el comando *biterr* que permite contar los errores, esta función devuelve la variable *Ratio*, que es el BER que se va acumulando en cada iteración en la variable *BER1* para el usuario uno y para el usuario dos en la variable *BER2*, luego se saca el BER promedio para cada usuario dividiendo el BER obtenido para el número de iteraciones realizadas como se indica en el Segmento de código 2.36.

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

Segmento de código 2.36. BER promedio.

Para graficar el resultado se utiliza la función *semilogy* que tiene como entrada el BER del usuario uno y dos, se coloca una etiqueta para identificar la señal con el uso de *legend*. Esto se observa en el Segmento de código 2.37.

```
104 - semilogy(EbNo_dB, BER1, EbNo_dB, BER2)
105 - legend('Usuario 1', 'Usuario 2')
106 - ylim([0.0001 0.5])
107 - xlim([0 EbNo_dB(end)])
108 - grid on
109 - xlabel('E_b/N_0 [dB]');
110 - ylabel('BER')
```

Segmento de código 2.37. Gráfica del BER de cada usuario.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En el presente capítulo se presentan los resultados obtenidos del sistema NOMA basado en OFDM. Los resultados se muestran en gráficas BER vs Eb/No, en donde, el BER está en el eje Y y el Eb/No en el eje X. Para este capítulo, se realiza un análisis de las gráficas obtenidas con las distintas técnicas de sincronización que se encuentran en el receptor, con distinto valor de STO positivo y negativo. Además, se realiza una comparación del rendimiento del sistema al utilizar diferentes configuraciones en cada capa con distintos factores de inyección. Finalmente, se incluyen las conclusiones y recomendaciones del trabajo.

3.1 ESCENARIO DE SIMULACIÓN

La simulación se basa en la transmisión y recepción de una señal que atraviesa un canal tipo Rician con ruido AWGN. Por esta razón, se generaron datos aleatorios para cada una de las capas, estas utilizan tasas de codificación 1/2, 2/3, 3/4 y pueden ser modulados con QPSK, 16 QAM o 64 QAM; de estas modulaciones se utilizan las dos primeras, ya que, se pueden obtener mejores resultados. En la capa uno se modifican los parámetros de modulación y tasa de codificación para observar su comportamiento, mientras que, en la capa dos se obtienen mejores resultados con la tasa de codificación 1/2 y modulación QPSK. En la capa dos se aplica el factor de inyección g el cual permite obtener un nivel de señal menor con respecto a la señal de la capa uno. A continuación, estos datos son multiplexados en OFDM para ser enviados a través del canal tipo Rician con ruido AWGN.

Este trabajo está enfocado en la sincronización en tiempo, por esta razón, se encarga de estimar el STO que fue añadido, con alguna de las técnicas de estimación STO presentadas en este trabajo. Luego de la estimación, la señal es compensada para realizar el proceso inverso en el receptor de cada usuario y recuperar la información de cada capa. Es importante mencionar que la recuperación de información de cada capa es diferente ya que existe un nivel de complejidad en cada receptor, es decir, la información que tiene cada capa se la recupera aplicando diferentes procesos.

El análisis se da en función de que Eb/No necesita para un valor específico de BER. Es decir, mientras menor sea el Eb/No que necesite para llegar a un BER dado, mayor es el rendimiento. Dentro del análisis es necesario mencionar que se utiliza siempre el mismo PDP para el canal Rician.

En la Tabla 3.1 se indican los parámetros que serán fijos durante todas las simulaciones del sistema empleado. También son fijos los parámetros PDP del canal tipo Rician y estos se muestran en la Tabla 3.2.

Tabla 3.1. Parámetros fijos de las simulaciones

Datos	Valor
Número de subportadoras	52(48 de datos y 4 pilotos)
Número de subportadoras total	64
Número de símbolos	8
Número de iteraciones	4000

Tabla 3.2. PDP Rural LOS

Datos	Valor
"Pot" Potencia en dB de cada rayo	Pot = [0 -14 -17]
"Delay" Retardo de cada rayo	Delay = [0 100 200]*1e-9
Factor "k"	K=20

3.2 PRUEBAS DEL SISTEMA NOMA

Para las pruebas de funcionamiento se realiza una comparación de la simulación del sistema NOMA OFDM empleado con un sistema NOMA que usa codificación LDPC (*Low Density Parity Check*), es decir, el sistema que vamos a comparar tiene distintas tasas de codificación a las empleadas en este trabajo.

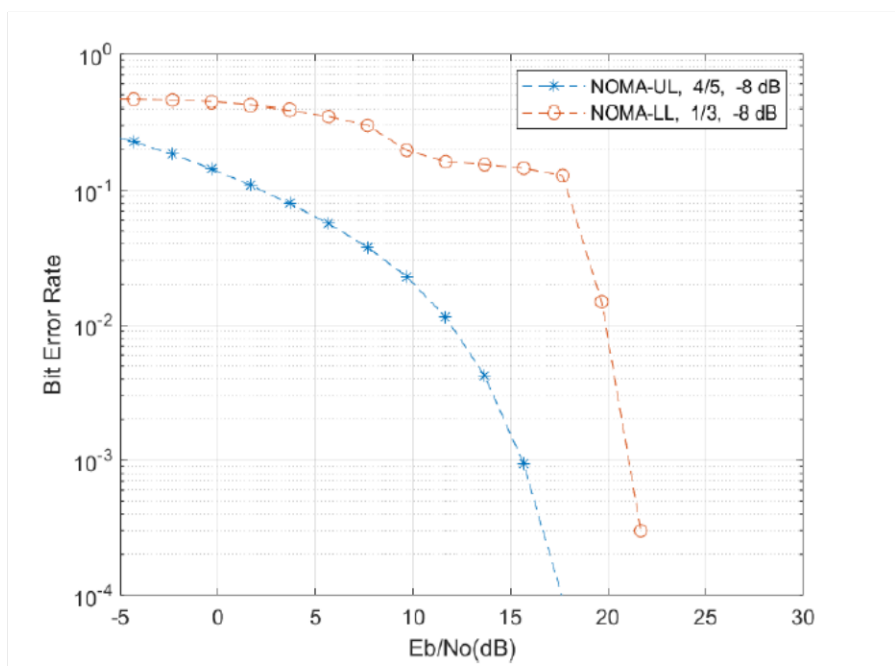


Figura 3.1. BER vs Eb/No del sistema NOMA con factor de inyección de -8 dB [2].

En la Figura 3.1, se puede observar la simulación del sistema NOMA con codificación LDPC, en donde, la tasa de codificación aplicada es UL: 4/5 y LL: 1/3, UL es la capa uno y LL es la capa dos del sistema. Para comparar con el sistema empleado se colocaron tasas de datos similares, en la capa uno se aplica 3/4 y en la capa dos 1/2, con el mismo factor de inyección de -8 dB. Se realizan 1000 iteraciones para comprobar el funcionamiento del sistema, además, es importante considerar que se está comparado únicamente con un canal AWGN. En la Figura 3.2, se puede observar el funcionamiento del sistema que tiene un comportamiento similar al de la Figura 3.1.

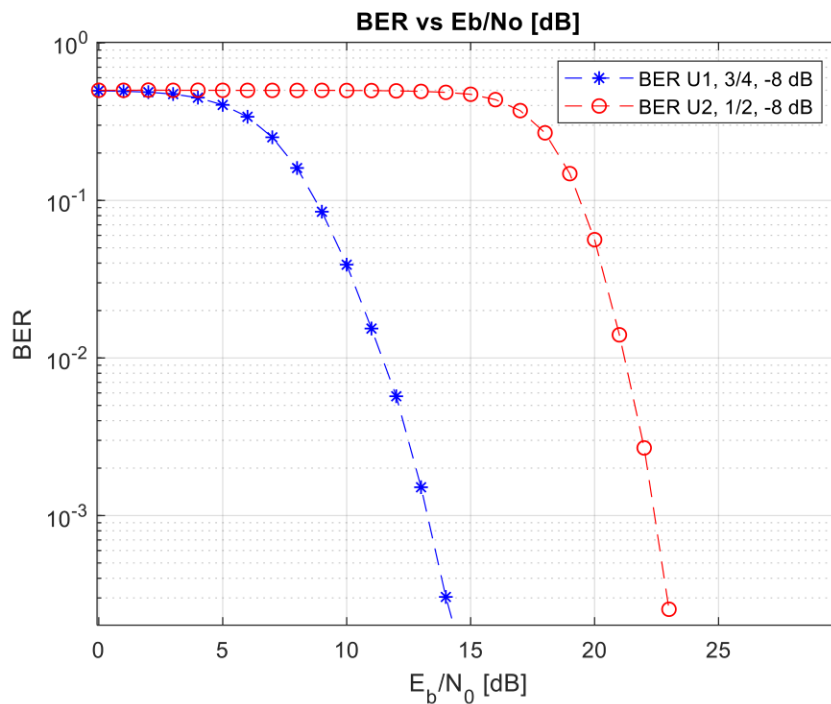


Figura 3.2. BER vs Eb/No del sistema empleado con factor de inyección de -8 dB.

A continuación, en la Figura 3.3 se muestra el funcionamiento del sistema NOMA basado en OFDM con un canal tipo Rician con ruido AWGN y se lo compara con un sistema que tiene únicamente un canal AWGN, con la finalidad de, comprobar que la simulación del canal tipo Rician funciona de forma correcta.

Como se puede observar en la Figura 3.3, para ambas capas las curvas del BER al usar un canal de tipo Rician se encuentran desplazadas hacia la derecha, comparando con las curvas de un canal únicamente con AWGN. Este comportamiento es el esperado, ya que, el canal con desvanecimiento Rician degrada el BER comparado con un canal puramente AWGN.

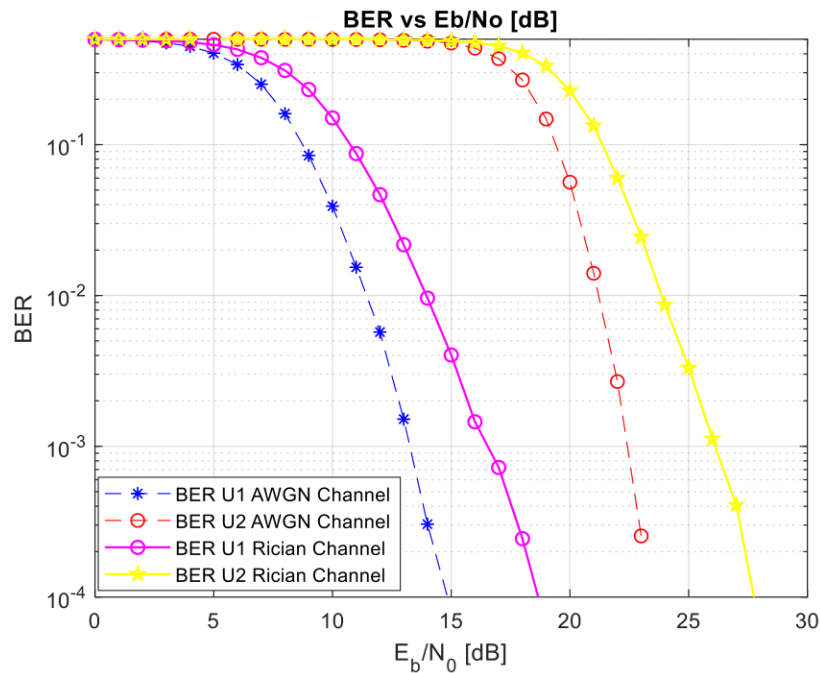


Figura 3.3. Comparación del BER vs Eb/No de un canal Rician vs un canal solo AWGN.

3.3 RESULTADOS

En esta sección se presentan los resultados de la simulación de las dos técnicas de estimación del STO mediante el uso del prefijo cíclico y el uso del preámbulo. La simulación se realizó con los siguientes parámetros, modulación QPSK para la capa uno y capa dos, las cuales son las más robustas, pero se envía menor cantidad de información, con una tasa de codificación de 3/4 para la capa uno y 1/2 para la capa dos, con la finalidad realizar una mejor corrección de errores en la capa del usuario dos. Se utiliza un factor de inyección de -4 dB, con un valor de 5 muestras positivas de STO.

A continuación, en la Figura 3.4 se presenta las curvas del sistema con las dos técnicas de estimación, estas se comparan con las curvas de sincronización perfecta que están con líneas entrecortas de color azul y rojo para el usuario uno y dos, respectivamente. Los valores que se obtienen aplicando la sincronización son muy cercanos a los valores del BER con sincronización perfecta, esto se observa en la técnica de estimación del STO con la mínima diferencia del preámbulo. Por otro lado, se observa que en la técnica de estimación del STO con diferencia mínima del prefijo cíclico tiene un BER similar hasta los 14 dB en Eb/No, para el usuario uno y para el usuario dos, luego de esto se obtiene un piso de ruido que indica que la técnica dejó de funcionar.

En la Tabla 3.3 se muestran los valores que se obtuvieron en la gráfica, para un BER dado, el valor de Eb/No que necesita cada técnica de estimación, siendo la técnica 1 la que utiliza

el prefijo cíclico y la técnica 2 usando el preámbulo la cual presenta mejor rendimiento que la técnica 1 en este escenario de simulación.

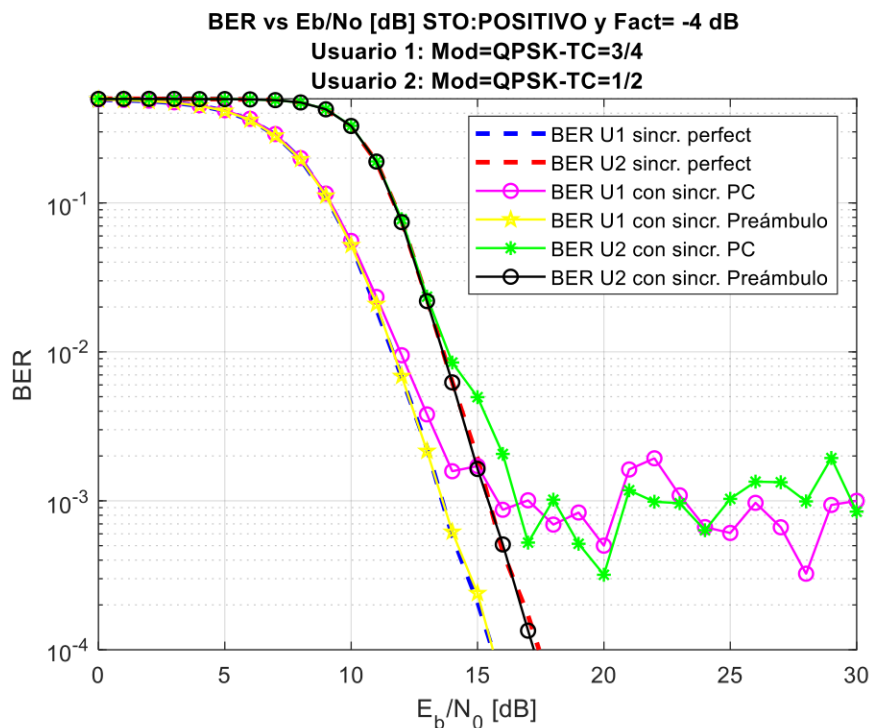


Figura 3.4. BER vs Eb/No utilizando las distintas técnicas de estimación con ST0=5.

Tabla 3.3. Valor de Eb/No para cada valor específico de BER de las dos técnicas implementadas.

Modulación U1=QPSK U2=QPSK Muestras de ST0: 5	Eb/No (dB)					
	USUARIO 1			USUARIO 2		
	Sinc. Perfect	Técnica 1	Técnica 2	Sinc. Perfect	Técnica 1	Técnica 2
BER 10 ⁻¹	9.14	9.19	9.14	11.68	11.71	11.68
BER 10 ⁻²	11.63	11.94	11.66	13.63	13.84	13.62
BER 10 ⁻³	13.60	-	13.60	15.45	-	15.42
BER 10 ⁻⁴	15.61	-	15.61	17.43	-	17.22

Para demostrar que las técnicas de estimación funcionan para diferentes valores de ST0 se realizan simulaciones con los mismos parámetros, únicamente variando el ST0 con un valor de 10 y -15 como se muestran en las Figuras 3.5 y 3.6.

Se puede observar, en la Figura 3.5 que al cambiar el ST0 a un valor positivo distinto al simulado anteriormente no influyó en el resultado, la técnica 2 sigue funcionando correctamente, mientras que, la técnica 1 sigue sin mostrar ninguna mejora. Al variar el

valor de STO a un valor negativo como se muestra en la Figura 3.6, se observa que la técnica 2 sigue funcionando y la técnica 1 continúa teniendo errores de estimación, siendo su mejor BER igual a 10^{-2} para ambos usuarios.

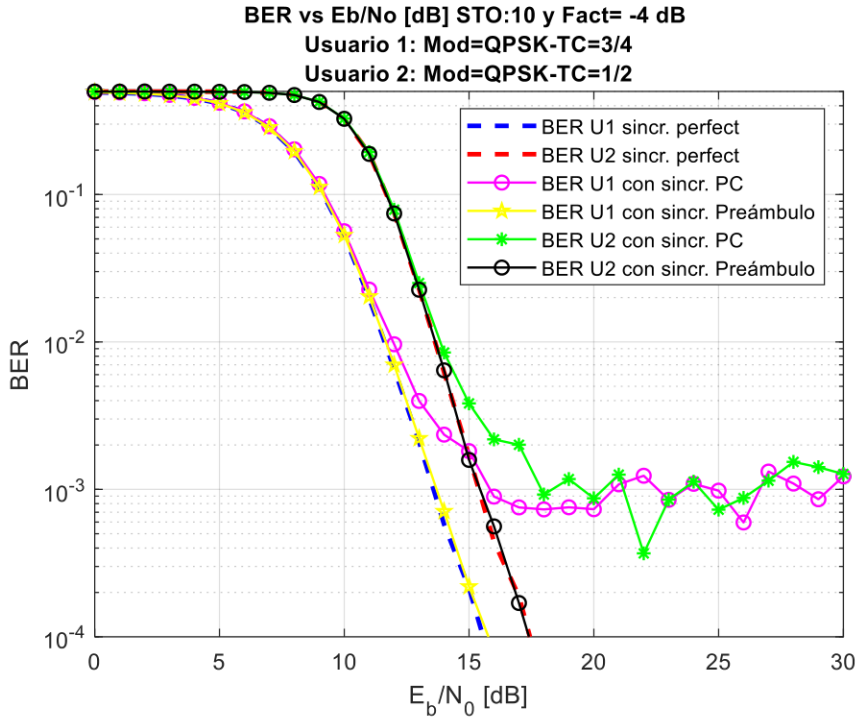


Figura 3.5. BER vs Eb/No para STO=10.

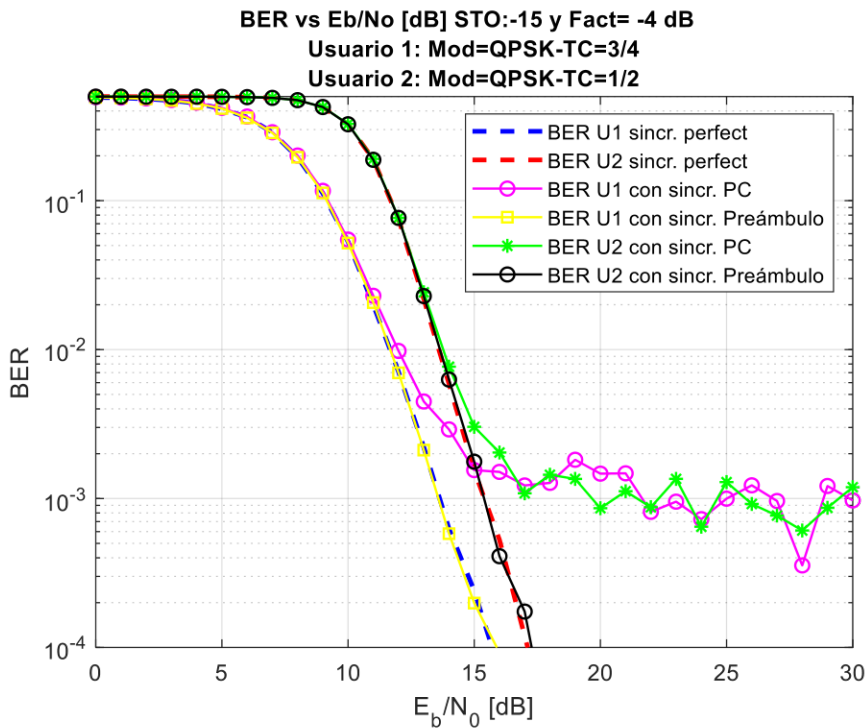


Figura 3.6. BER vs Eb/No STO=-15.

Como se puede observar, el funcionamiento de las técnicas de estimación STO no afecta su rendimiento si se modifica el valor del STO, este puede ser positivo o negativo, por esta razón, las simulaciones que se presentarán de aquí en adelante solo se realizarán con un valor de 5 muestras de STO positivo.

3.3.1 RESULTADOS DE SIMULACIONES CON MODULACIÓN 16 QAM PARA CAPA UNO

A continuación, se muestran las gráficas obtenidas de la simulación con los siguientes parámetros, modulación 16 QAM para la capa uno, con una tasa de codificación de 2/3, mientras que, para la capa dos se conservan la modulación QPSK y tasa de codificación de 1/2 y se emplean diferentes factores de inyección.

Se puede observar, que para esta configuración de capas se obtiene un mejor resultado para ambos usuarios usando el factor de inyección de -6 dB. Además, las técnicas de estimación siguen funcionando correctamente independientemente del valor del factor de inyección tomando en cuenta que la técnica del prefijo cíclico es la que tiene un menor rendimiento.

En la Figura 3.7, se puede observar que cuando el factor de inyección es de -4 dB, las dos capas tienen un BER similar, en la Figura 3.8 cuando el factor de inyección es -6 dB, mejora sustancialmente las dos capas y existe una mejora de alrededor de 13 dB en E_b/N_0 . Finalmente, en la Figura 3.9 con un factor de inyección de -8 dB en la capa uno se puede observar que mejora el BER mientras que la capa dos se degrada.

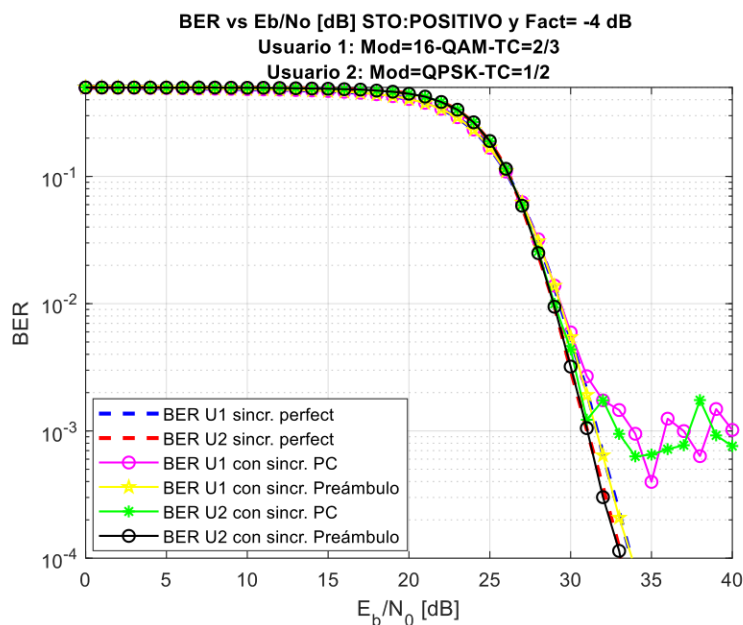


Figura 3.7. BER vs E_b/N_0 con factor de inyección de -4 dB.

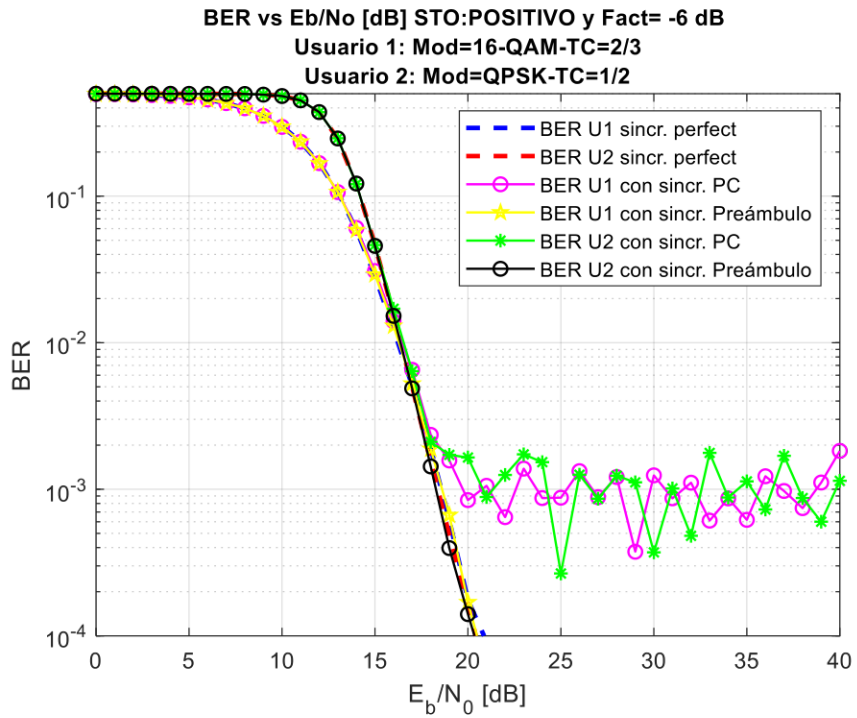


Figura 3.8. BER vs Eb/No con factor de inyección de -6 dB.

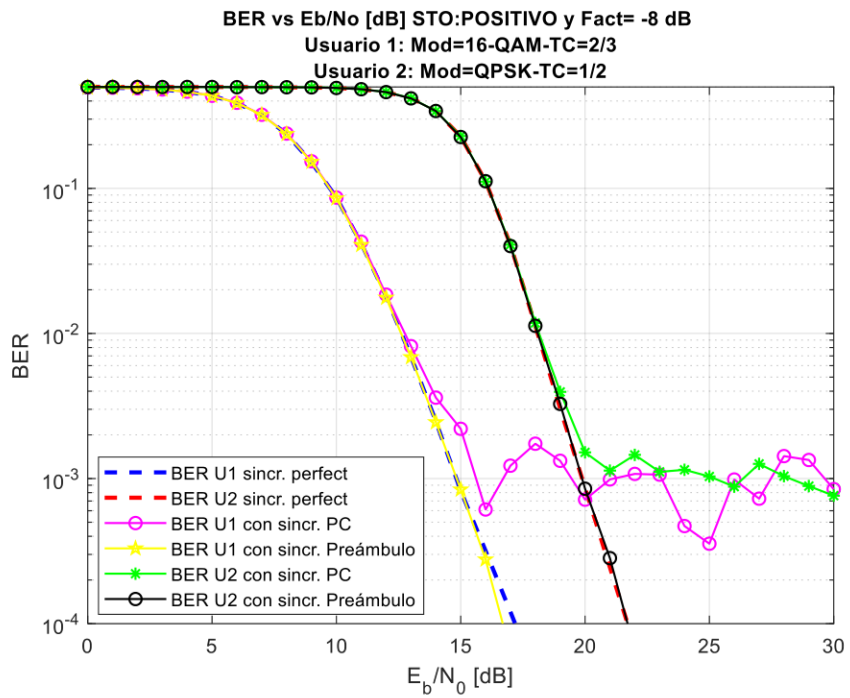


Figura 3.9. BER vs Eb/No con factor de inyección de -8 dB.

Se puede observar claramente que el mejor factor de inyección para esta configuración es -6 dB, por esta razón, en la Tabla 3.4 se muestran los valores que se obtuvieron en la gráfica, para un BER dado, el valor de Eb/No que necesita cada técnica de estimación.

Tabla 3.4. Valor de E_b/N_0 para cada valor específico de BER de las dos técnicas implementadas para un factor de inyección de -6 dB.

Modulación U1=16 QAM U2=QPSK Muestras de STO: 5	Eb/No (dB)					
	USUARIO 1			USUARIO 2		
	Sinc. Perfect	Técnica 1	Técnica 2	Sinc. Perfect	Técnica 1	Técnica 2
BER 10^{-1}	13.13	13.13	13.13	14.20	14.20	14.20
BER 10^{-2}	16.24	16.43	16.29	16.37	16.55	16.37
BER 10^{-3}	18.60	-	18.61	18.36	-	18.27
BER 10^{-4}	20.87	-	20.50	20.41	-	21.35

Para aumentar el rendimiento de la técnica de estimación usando el prefijo cíclico se aumenta el número de símbolos, ya que, el código implementado realiza una comparación del prefijo cíclico de cada símbolo, mejorando así la técnica. En la Figura 3.10 se puede observar el rendimiento de la técnica al utilizar 20 símbolos, con la misma configuración del mejor caso anterior, únicamente cambiando la tasa de codificación a 3/4 para la capa uno. Se puede notar que al usar la estimación por preámbulo no depende del número de símbolos a transmitir.

Esto se realiza con la finalidad de comprobar si mejora el rendimiento de la técnica de estimación basado en el uso del prefijo cíclico al aumentar el tamaño del paquete. Además, se puede observar que el piso de ruido se encuentra cerca de un valor de BER de 10^{-4} .

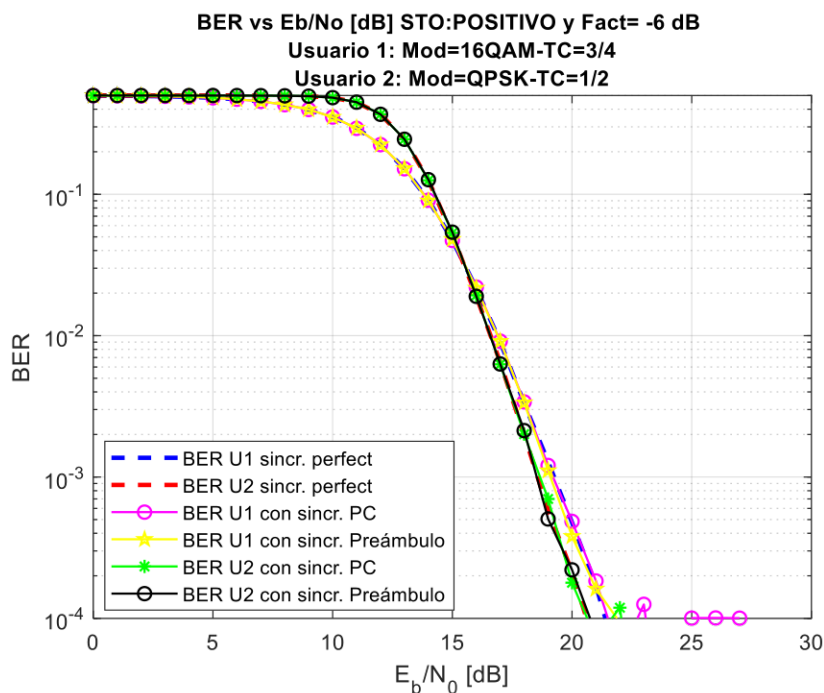


Figura 3.10. BER vs E_b/N_0 con 20 símbolos.

3.4 CONCLUSIONES

El sistema simulado NOMA basado en OFDM, permite a dos usuarios compartir un mismo recurso en el dominio de la potencia, pero al sumar la información de las capas provoca la degradación del BER para ambos usuarios y al añadir STO en la señal es necesario la utilización de técnicas de sincronización para la recuperación de la información.

La presencia de STO en la señal es muy crítica, ya que, provoca que la señal recuperada en el receptor no sea igual a la enviada por el transmisor, por lo que aumenta significativamente su BER, causando que se pierda la información de los usuarios, por esta razón son importantes las técnicas de estimación STO en el lado del receptor.

Se observa que la técnica de estimación utilizando el preámbulo presenta un mejor rendimiento que al usar el prefijo cíclico, esto se obtiene gracias a la estructura del preámbulo. A pesar de esto, se puede obtener un buen rendimiento en la técnica de estimación por prefijo cíclico si se aumenta el número de símbolos por lo menos a 20, ya que, su rendimiento depende del número de símbolos que se envía, a diferencia de la técnica de estimación del preámbulo que funciona correctamente a pesar de enviar pocos símbolos.

Al tener un valor de STO positivo, la compensación no afecta la información transmitida, ya que, se aumentan ceros en el preámbulo, por otro lado, si el valor de STO es negativo se pierden datos y la compensación se realiza al final de la información con ceros, pero estos datos pueden ser recuperados con ayuda del prefijo cíclico, lo que permite tener una buena estimación independientemente del signo del valor del STO.

Se observó un buen rendimiento del sistema al emplear en ambas capas una modulación QPSK con tasa de codificación de $3/4$ en la capa uno y tasa de codificación de $1/2$ en la capa dos con un factor de inyección de -4 dB, en donde se ve una separación entre el BER del usuario uno y el BER del usuario dos de 2 dB aproximadamente en E_b/N_0 , esto se debe a que esta modulación envía menos bits por símbolo y es menos probable que existan errores.

De acuerdo a las simulaciones dependiendo de la configuración de las capas, el factor de inyección influye en los resultados del rendimiento del sistema, en donde, -6 dB es un valor correcto para el equilibrio entre las capas al usar modulación 16 QAM con tasa de codificación $2/3$ para la capa uno y QPSK con tasa de codificación $1/2$ para la capa dos, ya que, al usar -4 dB se degrada el BER de ambos usuarios y con valores menores del factor de inyección se ve un mejor BER para el usuario de la capa uno, mientras que, para el

usuario dos se degrada, independientemente de estas configuraciones las técnicas de sincronización de STO siguen funcionando.

Se observa que las técnicas de sincronización desarrolladas para OFDM, se pueden aplicar satisfactoriamente para el sistema NOMA-OFDM, esto es posible, ya que, únicamente se suma la información de ambos usuarios, además, para la técnica usando el preámbulo este no es afectado por el factor de inyección.

Con un canal tipo Rician las técnicas funcionaron, sin embargo, con un PDP con mayores obstáculos para la señal el rendimiento de las técnicas se ve afectada, ya que, la información de las ventanas de comprobación que son utilizadas para la estimación STO son perjudicadas por el PDP.

3.5 RECOMENDACIONES

Se recomienda usar un gran número de iteraciones para obtener curvas bien definidas, con valores lo más cercanos a la realidad.

Se recomienda seleccionar distintos PDP para el canal Rician, a fin de analizar los efectos que tiene sobre las técnicas de estimación STO y cómo afecta esto en la sincronización.

Se recomienda aumentar el número de símbolos al utilizar la técnica de estimación por prefijo cíclico para obtener mejores resultados en su rendimiento y obtener una buena sincronización.

Como trabajo futuro, se puede combinar en el sistema distintas técnicas de sincronización, para reducir la incertidumbre y así mejorar la estimación para la compensación de STO, haciendo un sistema más robusto.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] K. B. Toromoreno Paz y Miño, "Simulación de técnicas de sincronización en sistemas OFDM," Escuela Politécnica Nacional, Quito, 2020.
- [2] C. M. Arrieta Tapia, "Simulación de un sistema de comunicación inalámbrica usando la técnica de acceso NOMA (Non - Orthogonal Multiple Access)," Escuela Politécnica Nacional, Quito, 2020.
- [3] L. Contreras, "Historia de la Informática," 2 Diciembre 2010. [En línea]. Available: <https://histinf.blogs.upv.es/2010/12/02/historia-de-las-redes-inalambricas/>. [Último acceso: 21 Noviembre 2021].
- [4] M. A. Del Valle Diego, R. Enríquez Caldera, J. M. Ramírez Cortés y J. Martínez Carballido, "OFDM wireless channel modeling, towards to 4G communications," *e-Gnosis*, pp. 1-7, 2009.
- [5] W. C. Y. Lee, "Future vision for wireless communications," *Proceedings of PIMRC '96 - 7th International Symposium on Personal, Indoor, and Mobile Communications*, vol. I, pp. 1-4, 1996.
- [6] A. L. Intini, "Orthogonal Frequency Division Multiplexing for Wireless Networks," University of California, 2000.
- [7] V. P. Gil Jiménez, "Algoritmos en Transmisión y Recepción para OFDM en entornos Multi-usuario," Universidad Carlos III de Madrid, Madrid, 2005.
- [8] C. V. Vargas, W. E. Lopez y C. F. da Rocha, "Sistemas de Comunicación Inalámbrica MIMO-OFDM," *GPqCom - UFSC*, vol. III, nº 4, pp. 761-777, 2007.
- [9] E. X. Bayas Moposita, "Análisis de la eficiencia de U-OFDM y UF-OFDM," Pontificia Universidad Católica del Ecuador, Quito, 2020.
- [10] L. Jiménez, J. Parrado, C. Quiza y C. Suárez, "Modulación Multiportadora OFDM," *Academia y desarrollo*, p. 5, 2001.
- [11] K. Higuchi y A. Benjebbour, "Non-orthogonal Multiple Access (NOMA) with Successive," *IEICE Transactions on Communications*, vol. 98, nº 3, pp. 403-414, 2015.

- [12] L. A. Vasquez Toledo, "Análisis de teletráfico de sistemas celulares con usuarios VoIP persistentes y calendarizador de máximo SIR basado en estadística de orden," Instituto Politécnico Nacional, 2018.
- [13] J. d. J. Sebastián Villa, "Evaluación del desempeño de las comunicaciones D2D en redes inalámbricas de quinta generación," Instituto Politécnico Nacional, 2018.
- [14] S. R. Islam, N. Avazov, O. A. Dobre y K.-S. Kwak, "Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 19, n° 2, pp. 721-742, 2017.
- [15] A. Khan y S. Y. Shin, "Linear Precoding Techniques for OFDM-Based NOMA over Frequency-Selective Fading Channels," *IETE Journal of Research*, vol. 63, n° 4, pp. 536-551, 2017.
- [16] Y. Soo Cho y W. Young Yang, *MIMO-OFDM Wireless Communications with MATLAB*, Asia: John Wiley & Sons, 2010.
- [17] W. Aziz, G. Abbas, E. Ahmed, S. Saleem y Q. Islam, "Time Offset Estimation for OFDM Using MATLAB," *World Science Publisher*, vol. 1, n° 2, pp. 56-61, 2012.
- [18] R. Hernández Sampieri, C. Fernández Collado y P. Baptista Lucio, *Metodología de la Investigación*, México: McGRAW-Hill, 2014.

5 ANEXOS

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