

# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

### **SIMULACIÓN DE ESTIMACIÓN DE CANAL Y SINCRONIZACIÓN EN LA TÉCNICA DE COMUNICACIÓN INALÁMBRICA SEFDM**

#### **SIMULACIÓN DE ESTIMACIÓN DE CANAL EN SEFDM**

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

**CRISTIAN ANDRÉS GALLO RIVERA**

**cristian.gallo@epn.edu.ec**

**DIRECTOR: DIEGO JAVIER REINOSO CHISAGUANO**

**diego.reinoso@epn.edu.ec**

**DMQ, octubre 2022**

## **CERTIFICACIONES**

Yo, CRISTIAN ANDRÉS GALLO RIVERA 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.



---

**CRISTIAN ANDRÉS GALLO RIVERA**

Certifico que el presente trabajo de integración curricular fue desarrollado por CRISTIAN ANDRÉS GALLO RIVERA, 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.

CRISTIAN ANDRÉS GALLO RIVERA

DIEGO JAVIER REINOSO CHISAGUANO

## **DEDICATORIA**

A mis padres Luis y Carmen por su amor y apoyo incondicional durante toda mi vida. A mi hermano David y a mis abuelitas Emma y Graciela por su cariño incansable hacia mi persona.

## **AGRADECIMIENTO**

Agradezco el amor de toda mi familia, especialmente de mis padres, hermano y abuelitas. Gracias por el apoyo, confianza y motivación que me han permitido convertirme en la persona que soy.

Agradezco a mis amigos Caro, Mila y Steven por su sincera amistad, por los gratos momentos que hemos compartido y por acompañarme en los momentos más difíciles de mi vida universitaria.

Agradezco a todos los ingenieros que me permitieron adquirir nuevos conocimientos, especialmente al Dr. Diego Reinoso por el apoyo que supe recibir para poder realizar el presente trabajo.

# 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
1.3.1 Fase de diseño .....	2
1.3.2 Fase de implementación .....	3
1.3.3 Fase de evaluación .....	3
1.4 Marco teórico .....	3
1.4.1 Sistemas SEFDM .....	3
1.4.1.1 Transmisor y receptor SEFDM .....	5
1.4.1.2 Canal .....	9
1.4.2 Estimación de canal .....	10
1.4.2.1 Símbolos piloto y preámbulo .....	11
2 METODOLOGÍA.....	12
2.1 Programa principal.....	12
2.1.1 Ingreso de variables .....	12
2.2 Función SEFDM_Rician.....	13
2.2.1 Etapa de transmisión SEFDM .....	14
2.2.1.1 Generación de bits y codificación FEC .....	14
2.2.1.2 Modulación QPSK-QAM .....	15
2.2.1.3 IFFT y prefijo cíclico .....	15
2.2.1.4. Preámbulo .....	16
2.2.2 Canal inalámbrico .....	17

2.2.3	Etapa de recepción SEFDM .....	18
2.2.3.1	Estimación de canal SEFDM .....	19
2.2.3.2	Lazo de recepción de datos SEFDM .....	24
2.2.3.3	Detección MMSE .....	25
2.2.3.4	Demodulación y decodificación FEC .....	26
2.2.3.5	Cálculo de errores .....	28
2.3	Gráfica de resultados .....	28
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	30
3.1	Escenario de simulación .....	30
3.2	Análisis de resultados .....	31
3.2.1	Comparación entre OFDM y SEFDM .....	31
3.2.2	Comparación entre estimación perfecta y estimación LS .....	34
3.2.3	Análisis del factor de compresión .....	37
3.3	Conclusiones.....	39
3.4	Recomendaciones.....	40
4	REFERENCIAS BIBLIOGRÁFICAS .....	41
5	ANEXOS.....	43

## RESUMEN

El presente trabajo de integración curricular presenta una simulación en el software de MATLAB para un sistema SEFDM (*Spectrally Efficient Frequency Division Multiplexing*) con estimación de canal. Para el estado del canal inalámbrico se usa ruido AWGN (*Additive White Gaussian Noise*) y desvanecimiento de tipo Rician. Los resultados de la simulación son presentados mediante graficas de BER (*Bit Error Rate*) vs  $E_b/N_0$  (*Energy per-bit to noise power spectral density ratio*).

El capítulo 1 presenta una introducción bibliográfica y un diagrama de bloques simple, acerca de los sistemas SEFDM. Además, se presenta los conceptos teóricos sobre la estimación de canal.

El capítulo 2 detalla la implementación del sistema SEFDM para MATLAB, se compone de una etapa de transmisión, canal inalámbrico y etapa de recepción. Para la etapa de recepción se desarrolla la técnica de estimación LS con método SEFDM-OFDM.

El capítulo 3 muestra los resultados de las curvas de BER vs  $E_b/N_0$ . Se compara la técnica de estimación perfecta y la técnica de estimación LS desarrollada. Se modifican los parámetros configurables para verificar como afectan al rendimiento del BER. Además, se presentan las conclusiones y recomendaciones del trabajo realizado.

**PALABRAS CLAVE:** SEFDM, canal inalámbrico, estimación de canal, BER.



## **ABSTRACT**

This curricular integration project presents a simulation in MATLAB software for SEFDM (Spectrally Efficient Frequency Division Multiplexing) system with channel estimation. AWGN noise (Additive White Gaussian Noise) and Rician fading are used for the state of the wireless channel. The results of the simulation are presented by plotting BER (Bit Error Rate) vs.  $E_b/N_0$  (Energy per-bit to noise power spectral density ratio).

Chapter 1 presents a bibliographical introduction and a simple block diagram about SEFDM systems. In addition, the theoretical concepts of channel estimation are presented.

Chapter 2 details the implementation of the SEFDM system for MATLAB, it is composed of a transmission stage, a wireless channel and a reception stage. For the reception stage, the LS estimation technique is developed with the SEFDM-OFDM method.

Chapter 3 shows the results of the BER vs  $E_b/N_0$  curves. The perfect estimation technique and the developed LS estimation technique are compared. The configurable parameters are modified to verify how they affect the performance of the BER. In addition, the conclusions and recommendations of the work carried out are presented.

**KEYWORDS:** SEFDM, wireless channel, channel estimation, BER.

# 1 INTRODUCCIÓN

En el campo de las telecomunicaciones existe una constante evolución por encontrar sistemas con mejores tasas de transmisión, por tanto, el aumento de la eficiencia espectral nace de la necesidad de mejorar las tecnologías inalámbricas, basadas en sistemas multiportadoras [1]. Cada día existen más dispositivos electrónicos que se conectan de manera inalámbrica, además, existen requerimientos de transmitir más datos con menor ancho de banda, prueba de ello es la tecnología 5G de sistemas celulares, donde se busca una alta eficiencia espectral, mayor número de usuarios, reducción en la latencia, etc.

En el año 2002 M. Rodríguez y I. Darwazeh proponen un nuevo esquema OFDM (*Orthogonal Frequency Division Multiplexing*), llamado Fast-OFDM, este nuevo esquema tiene el doble de eficiencia de ancho de banda [2]. Un año después los mismos autores proponen un sistema multiportadora denominado SEFDM (*Spectrally Efficient Frequency Division Multiplexing*), este sistema manejaba separaciones más pequeñas entre las portadoras que los sistemas OFDM [3].

Los sistemas de alta eficiencia fueron desarrollados en las comunicaciones para mejorar futuros estándares inalámbricos. SEFDM al ser una técnica de acceso multiportadora basada en OFDM, obtiene una mejor eficiencia espectral. La eficiencia espectral obtenida en sistemas SEFDM se fundamenta en el ahorro de ancho de banda, esto se logra al comprimir las subportadoras mediante un factor de compresión, las subportadoras al comprimirse pierden ortogonalidad lo cual produce una superposición que causa una interferencia entre las mismas, conocida como ICI (*Inter-Carrier Interference*) [1] [4].

En la práctica los sistemas multiportadoras como OFDM y SEFDM al usar un medio inalámbrico, aparte de lidiar con el ruido, están expuestos a sufrir desvanecimiento, este se caracteriza por la variación en la amplitud de la señal tanto en el dominio del tiempo, como de la frecuencia, por tanto, los procesos de estimación de canal y de ecualización en el receptor son indispensables para recuperar de manera fiable la información transmitida. La estimación de canal es necesaria en cualquier sistema de comunicaciones inalámbricas, porque en la práctica se desconoce el estado de canal. En el receptor se requiere la estimación de canal para realizar la ecualización y compensación del estado del canal. Dicho estado es desconocido y aleatorio, al realizar la estimación se logra conocer el comportamiento del canal para compensar los cambios que se den en la señal y recuperarla en la recepción [5].

En el presente trabajo se implementará una simulación en MATLAB basado en un sistema SEFDM con estimación de canal. Para las simulaciones se utilizarán diferentes tipos de

modulaciones, un factor de compresión variable, y un canal con ruido, de esta manera, se logrará determinar el rendimiento de la estimación y ver cómo afecta a la tasa de errores.

## **1.1 OBJETIVO GENERAL**

Realizar la simulación de un sistema SEFDM con estimación de canal en MATLAB.

## **1.2 OBJETIVOS ESPECÍFICOS**

1. Diseñar el diagrama de bloques para la simulación de un sistema SEFDM con estimación de canal.
2. Revisar la bibliografía relacionada con las técnicas de estimación de canal para un sistema SEFDM.
3. Implementar una simulación en MATLAB con una técnica de estimación de canal para SEFDM.
4. Analizar los resultados obtenidos en la simulación usando diferentes parámetros.

## **1.3 ALCANCE**

Se implementará una simulación de un sistema SEFDM con estimación de canal usando MATLAB, el sistema SEFDM utilizará un codificador convolucional con una tasa de 1/2. Los sistemas SEFDM utilizan un factor de compresión para reducir el ancho de banda, el factor de compresión será un parámetro configurable en la simulación. Además, se tendrá diferentes tipos de modulación como QPSK, 16QAM y 64QAM. Para la parte de canal se utilizará un canal con ruido AWGN (*Additive White Gaussian Noise*) con desvanecimiento. Para realizar la estimación de canal en el receptor se usará un preámbulo, es decir, un símbolo conocido transmitido para conocer el estado del canal y facilitar la sincronización, en este caso el sistema SEFDM tendrá una sincronización perfecta. Para las simulaciones se variarán los parámetros de modulación y factor de compresión. Además, de analizar cómo afecta a un sistema SEFDM utilizar o no estimación de canal. Se obtendrá como resultado una gráfica con una curva de BER (*Bit Error Rate*) vs  $E_b/N_0$ , comparando la gráfica cuando se tiene una estimación perfecta. El trabajo se desarrollará en 3 fases:

### **1.3.1 FASE DE DISEÑO**

Se realizará una revisión bibliográfica acerca de sistemas SEFDM, selección de la técnica de estimación de canal a utilizarse en la simulación, se diseñará un diagrama de bloques para la simulación de un sistema SEFDM con estimación de canal.

### **1.3.2 FASE DE IMPLEMENTACIÓN**

Se implementará una simulación en MATLAB para un sistema SEFDM con estimación de canal, se desarrollará la simulación para que los parámetros como modulación y factor de compresión puedan ser configurables.

### **1.3.3 FASE DE EVALUACIÓN**

Se realizará una prueba de funcionamiento para verificar que los resultados obtenidos correspondan con los resultados de trabajos similares, se realizará un análisis de resultados obtenidos y finalmente se desarrollarán las conclusiones y recomendaciones.

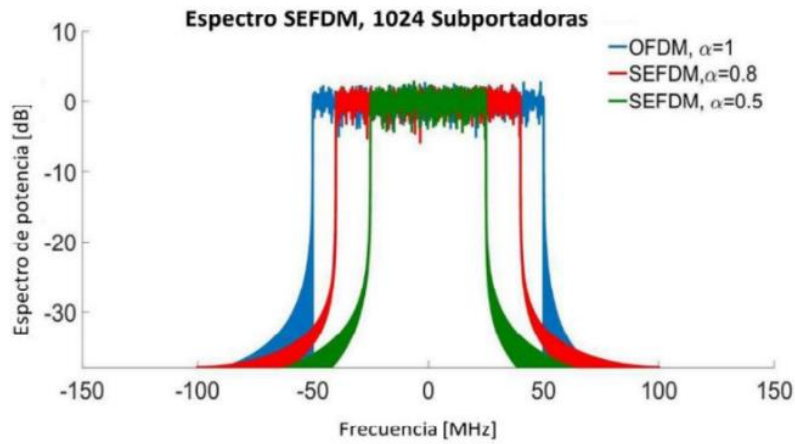
## **1.4 MARCO TEÓRICO**

En esta sección se describe acerca de los sistemas SEFDM, se detallarán los bloques de transmisión y recepción de estos sistemas. Además, se explicará acerca de la estimación de canal y como se realiza la estimación de canal en la recepción.

### **1.4.1 SISTEMAS SEFDM**

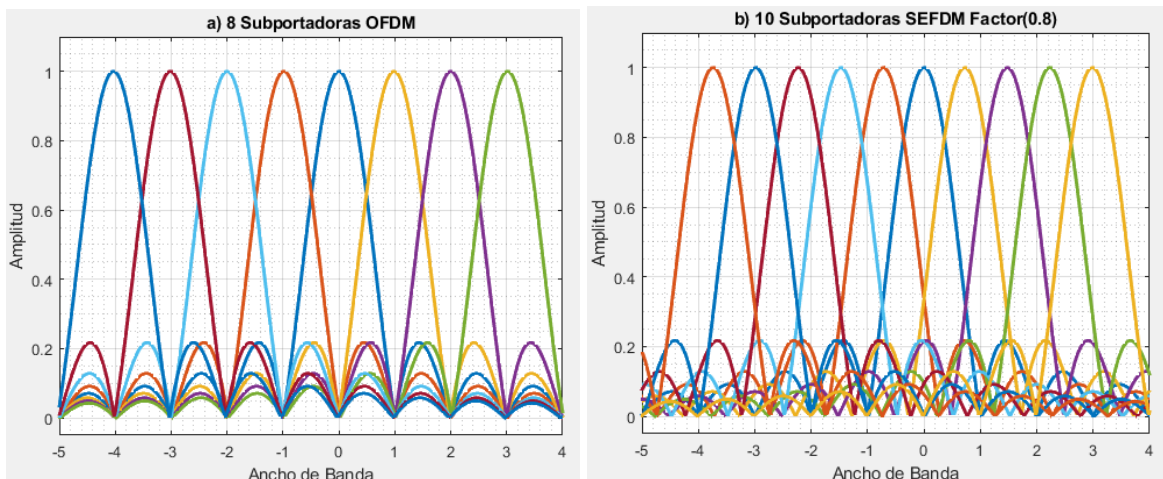
SEFDM es una técnica de acceso multiportadora, se fundamenta en el ahorro de ancho de banda que se logra al comprimir las subportadoras, de esta forma, se consigue una mejor eficiencia espectral con respecto a sistemas OFDM. La eficiencia de SEFDM se da a costa de la degradación del BER que se produce al perder la ortogonalidad de las subportadoras, al perder dicha ortogonalidad en las subportadoras se produce una interferencia denominada como ICI. Para lograr corregir este fenómeno conocido como ICI se introduce un bloque detector en la parte de recepción de un sistema SEFDM [4].

Para lograr comprimir las subportadoras en los sistemas SEFDM se utiliza un factor de compresión denotado por  $\alpha$ , este factor está definido en un dominio de  $]0,1[$ . Por tanto, la separación entre las subportadoras consecutivas es  $\Delta f = \frac{\alpha}{T}$ , donde  $T$  es el tiempo del símbolo SEFDM, en otras palabras, el espacio que existía entre subportadoras se comprimirá dependiendo del valor del factor de compresión. En SEFDM el ancho de banda que se ahorra es  $(1-\alpha) * 100\%$  con respecto a OFDM [6] [7]. En la figura 1.1 se puede observar diferentes espectros de potencia, uno de ellos es de un sistema OFDM, el resto son de sistemas SEFDM con diferentes factores de compresión. Al visualizar la imagen se aprecia como dependiendo del valor del factor de compresión se reduce el ancho de banda, para un mismo número de subportadoras.



**Figura 1.1.** Espectros de Potencia [8].

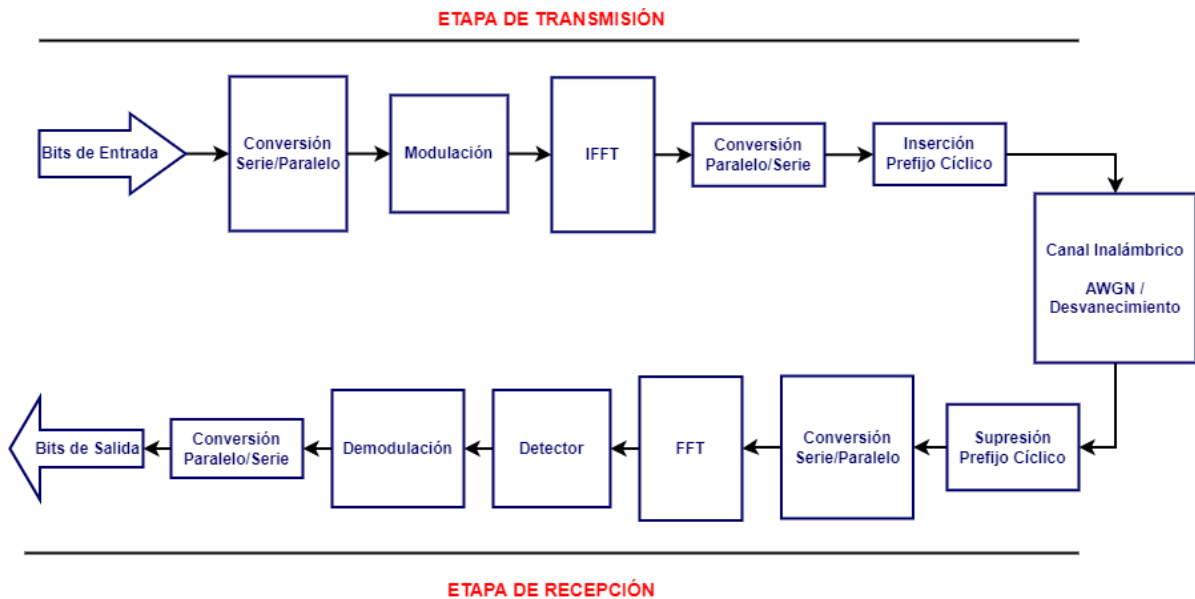
SEFDM es una técnica basada en OFDM, por tanto, cuando ambas técnicas tienen el mismo ancho de banda, SEFDM logra transmitir a mayores velocidades, debido a que mayor cantidad de símbolos son enviados por segundo. Para determinar el factor de velocidad de datos que aumenta para SEFDM es  $\frac{1-\alpha}{\alpha} * 100\%$ . Dado un mismo ancho de banda, se define  $S_{SEFDM} = \frac{S_{OFDM}}{\alpha}$ , donde  $S_{SEFDM}$  es el número de subportadoras de un sistema SEFDM, y  $S_{OFDM}$  es el número de subportadoras de un sistema OFDM [9] [10]. En la figura 1.2 se puede observar la relación que existe entre SEFDM y OFDM, para el número de subportadoras que tendrá SEFDM en un mismo ancho de banda con respecto a OFDM, utilizando un factor de compresión de 0.8 y para 8 subportadoras OFDM, el número de subportadoras de SEFDM será 10.



**Figura 1.2.** Subportadoras a) OFDM  $\alpha=1$  b) SEFDM  $\alpha=0.8$ . Basado en [8].

### 1.4.1.1 Transmisor y receptor SEFDM

En la figura 1.3 se detalla el esquema de transmisión y recepción de un sistema SEFDM.



**Figura 1.3.** Esquema de Transmisión y Recepción SEFDM. Basado en [8].

A continuación, se detallan todos los bloques utilizados para un sistema SEFDM:

#### Bits de Entrada

Los bits de entrada son bits que se desean transmitir, estos bits pueden ser 0 o 1. Estos contienen la información de forma digital.

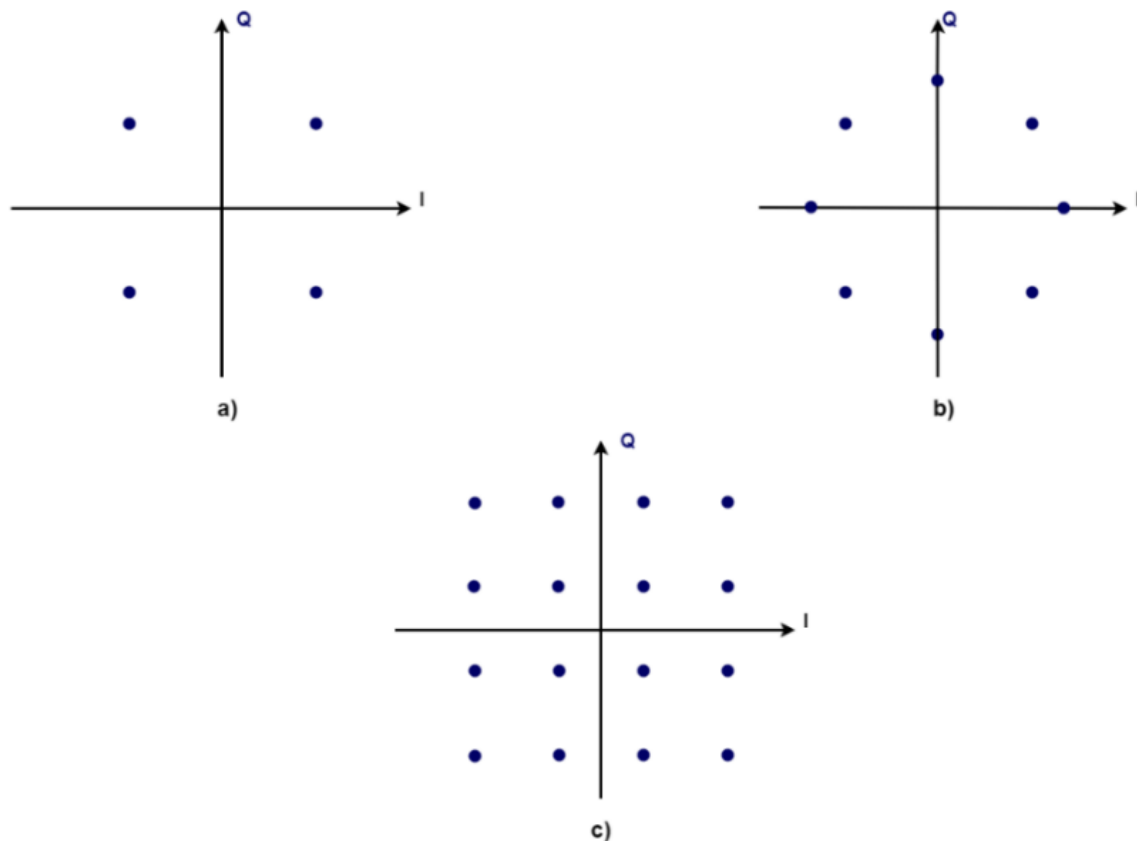
#### Conversión Serie/Paralelo o Paralelo/Serie

En los sistemas SEFDM los datos deben ser convertidos de serie a paralelo para poder ingresar al bloque IFFT (*Inverse Fast Fourier Transform*) en la transmisión y al bloque FFT (*Fast Fourier Transform*) en la recepción. Se usa conversión paralelo a serie para poder enviar los datos por el canal luego del proceso de IFFT en transmisión y para recuperar los datos en la recepción luego del bloque FFT.

#### Modulación y Demodulación de Datos

El proceso de modulación consiste en agrupar datos de bits, para ser enviados por el canal, la modulación se puede presentar en diagramas de constelaciones, para SEFDM se utilizan modulaciones de tipo PSK (*Phase Shift Keying*) y QAM (*Quadrature Amplitude Modulation*) principalmente. Cuando se realizan modulaciones con un mayor nivel de modulación, se brinda una mejor eficiencia espectral; sin embargo, el BER puede verse empeorado debido a que los puntos en el diagrama de constelación se encontraran más cercanos, por tanto,

están más expuestos a interferencias y ruidos [8]. En la figura 1.4 se muestra el diagrama de constelación de diferentes tipos de modulaciones.



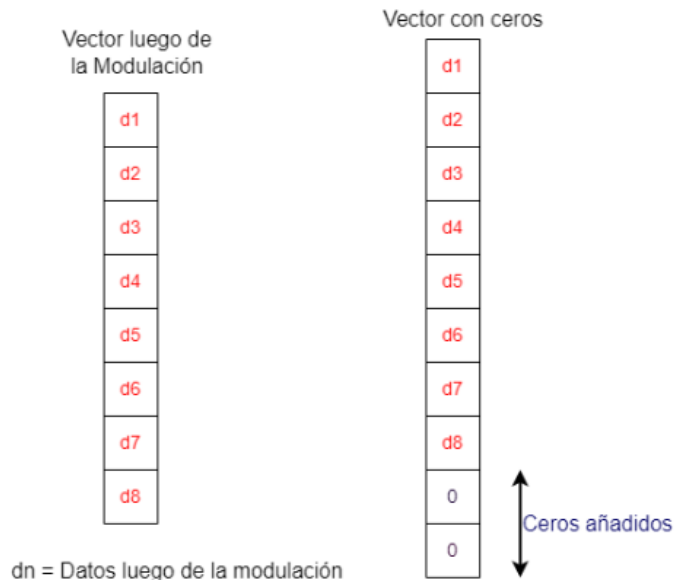
**Figura 1.4.** Diagramas de Constelación a) QPSK, b) 8-PSK, c) 16 QAM.

La demodulación busca recuperar la señal de datos antes de ser modulada en el transmisor, de esta manera, se recupera la información transmitida en el receptor. Dependiendo del nivel de modulación que se empleó, la recuperación de la señal será más fiable a los datos originales.

### IFFT y FFT

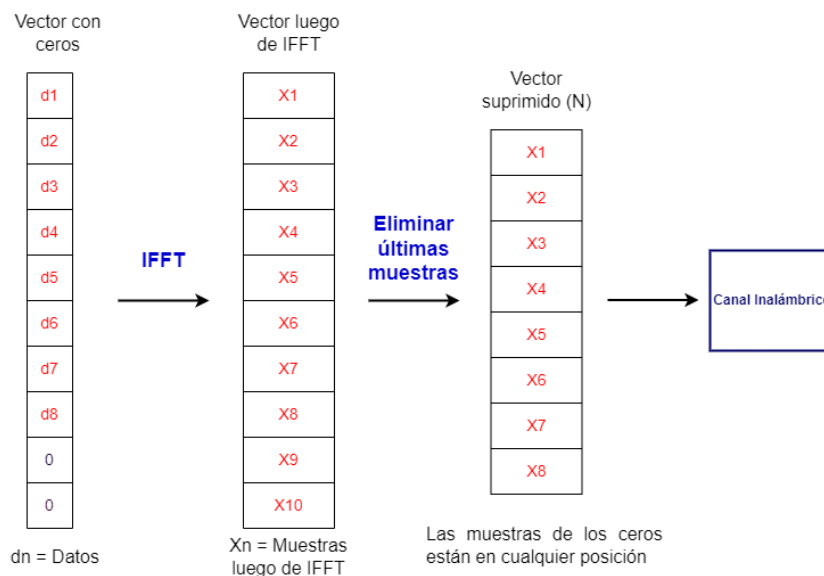
Una vez que los datos han sido modulados ingresan a un bloque de IFFT el cual utiliza un algoritmo rápido de transformada de Fourier, mediante la IDTF (*Inverse Discrete Fourier Transform*) es posible convertir a la señal del dominio de la frecuencia al dominio del tiempo, esta señal será transmitida por el canal [8]. Dado un esquema de  $N$  subportadoras y el bloque IFFT se genera una señal en el dominio del tiempo. Siendo  $N = 8$  y al usar un factor de compresión de  $\alpha = 0.8$ , a la salida del bloque de modulación se tiene un vector de  $8 \times 1$  y a este vector se le deben añadir un número específico de ceros, dados por la

función  $ceros = \frac{1-\alpha}{\alpha} * N = \frac{1-0.8}{0.8} * 8 = 2$  creando un nuevo vector con muestras añadidas, como se muestra en la figura 1.5.



**Figura 1.5.** Vector con muestras de ceros añadidos. Basado en [8].

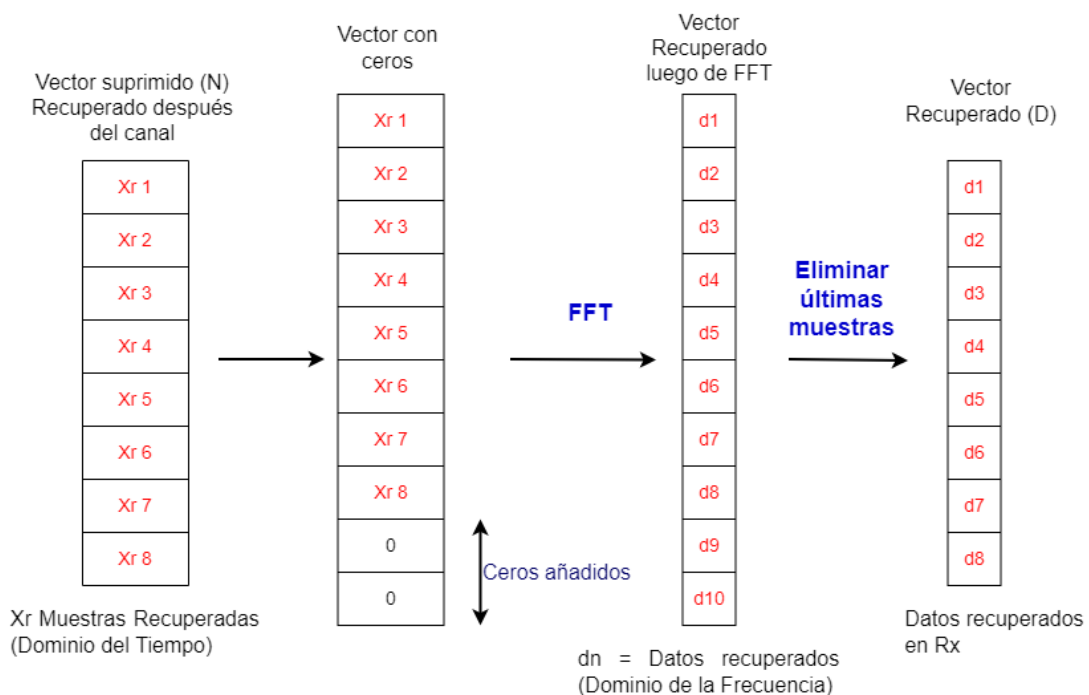
Luego de que al vector se añadieron los ceros, este pasa al bloque IFFT. Por lo tanto, las muestras insertadas como ceros estarán ubicadas en cualquier posición. El tamaño de este vector está definido por  $\frac{N}{\alpha} = \frac{8}{0.8} = 10$ . El nuevo vector tiene dimensiones  $10 \times 1$ . El objetivo para el vector luego de IFFT es que se eliminen las últimas muestras, para formar un nuevo vector dado por  $\frac{N}{\alpha} - \left(\frac{N}{\alpha} - N\right) = N = 8$  [8]. De esta manera, se recuperará el tamaño de  $N$  para que el vector se transmita por el canal, como se muestra en la figura 1.6.



**Figura 1.6.** Proceso de IFFT. Basado en [8].



En la recepción se realiza el proceso contrario que la transmisión. En la figura 1.7 se tiene un vector  $N$  con 8 muestras que llegan al receptor, luego del canal inalámbrico; en este vector se deben añadir los ceros correspondientes de la definición de SEFDM para un  $\alpha=0.8$ . Por lo tanto, el tamaño del nuevo vector está definido por  $\frac{N}{\alpha} = \frac{8}{0.8} = 10$ . El vector con los ceros añadidos ingresa al bloque FFT. Por lo cual, la señal pasa del dominio del tiempo al dominio de la frecuencia. Finalmente, son eliminadas las últimas muestras para formar un nuevo vector de datos recuperados dado por  $\frac{D}{\alpha} - \left(\frac{D}{\alpha} - N\right) = N = 8$ . De esta manera, se recupera el tamaño vector de datos en la recepción, el cual será demodulado posteriormente [8].



**Figura 1.7.** Proceso de FFT. Basado en [8].

### Prefijo Cíclico

La banda de guarda que se introduce en los símbolos es conocida como prefijo cíclico (CP). El prefijo cíclico consiste en colocar las últimas muestras del símbolo al inicio de este. La inserción de prefijo cíclico es para evitar la interferencia de ISI (*Inter Symbol Interference*) y mitigar los efectos de ICI. El prefijo cíclico se incluye después del bloque de IFFT ya que pertenece al dominio del tiempo. Además, debe ser mayor o igual al máximo retardo del canal. La supresión del prefijo cíclico es realizada antes de ingresar al bloque FFT en la etapa de la recepción [8] [11].

## Detector

La recuperación de la señal es un proceso indispensable en los sistemas de comunicación, en sistemas SEFDM al perder la ortogonalidad de las subportadoras, la interferencia crece, degradando el BER, por ello, es necesario incluir un bloque detector para mitigar el ICI, la complejidad de este bloque dependerá del nivel de modulación que se usó y del tamaño del sistema. El detector a partir de datos recibidos puede estimar la señal transmitida [8].

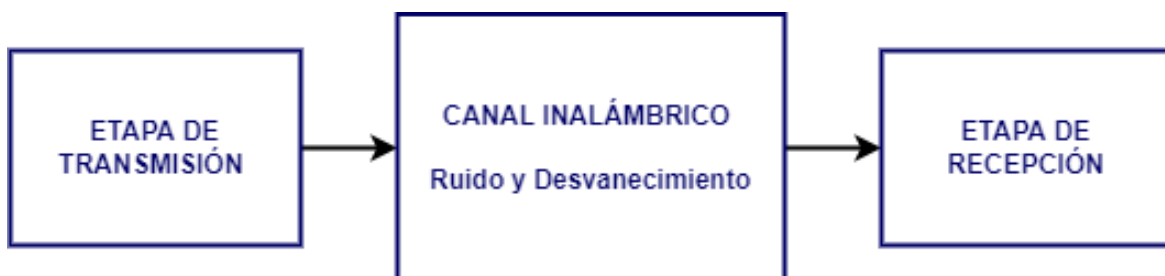
## Bits de Salida

Los bits de salida son bits recuperados en la recepción, estos buscan ser lo más parecidos a los bits de entrada, por lo cual, se busca un BER que tienda a ser 0, estos bits toman valores de 0 o 1.

### 1.4.1.2 Canal

El canal es el medio por donde viajan las señales de información de un transmisor a un receptor, los sistemas SEFDM son utilizados para sistemas de comunicación inalámbricos, por tanto, su medio de transmisión es el aire. Existen diversos modelos de representación para canales inalámbricos, el más simple es un canal AWGN, donde al canal se le añade ruido AWGN, por tanto, en la parte de recepción se recibe la señal transmitida más el ruido del canal [8].

Otra característica de un canal inalámbrico es el fenómeno denominado como desvanecimiento, se conoce como desvanecimiento a la variación de la amplitud de la señal en el dominio del tiempo y de la frecuencia, el desvanecimiento corresponde a una degradación de la señal. Este se debe a la propagación de múltiples trayectos denominado como desvanecimiento por multitrayectoria (*multipath fading*) o cuando existen obstáculos en la propagación de la señal, desvanecimiento por sombra (*shadowing*) [11]. En la figura 1.8 se observa un diagrama de transmisión-canal-recepción para SEFDM.



**Figura 1.8.** Esquema de Transmisión-Canal-Recepción (SEFDM).

## 1.4.2 ESTIMACIÓN DE CANAL

La estimación de canal es un proceso por el cual se determinan y corrigen los cambios que se producen en las señales de información cuando son transmitidos por un canal con ruido y desvanecimiento. Este proceso es importante para todos los sistemas de comunicación inalámbricos, al desconocer el estado del canal, el receptor debe realizar un proceso de detección y compensación en la señal para recuperar los datos que fueron transmitidos correctamente [11].

Debido a la naturaleza de los sistemas SEFDM es indispensable realizar un proceso de estimación de canal, debido a la pérdida de ortogonalidad, las interferencias que se producen en un canal inalámbrico son mucho mayores. Al usar estimación de canal es posible evitar el impacto del mal acondicionamiento del sistema lo cual beneficia al rendimiento de la tasa de errores [5].

Existen diversas técnicas de estimación de canal para SEFDM como se menciona en [12]. Los esquemas de estimación en el dominio del tiempo se basan en la transmisión de símbolos piloto que invocan la estructura de ortogonalidad completa FCE (*Full Channel Estimation*) o estructura de ortogonalidad parcial PCE (*Partial Channel Estimation*), este último logra un mejor rendimiento que las técnicas de estimación que se basan en la estimación completa. Aunque estos esquemas tienen un desempeño aceptable, en la práctica se ven limitados al tener una mayor complejidad, debido a que requieren inversiones de matrices [5] [12].

Los esquemas en el dominio de la frecuencia resaltan sobre los esquemas en el dominio del tiempo, por su simplicidad, pocos requerimientos computacionales, alto rendimiento y precisión, independientemente del factor de compresión o número de subportadoras. En los esquemas de estimación para el dominio de la frecuencia se divide las regiones de multiplexación en ortogonales y no ortogonales. La región ortogonal es usada para la estimación mediante el método LS (*Least Squares*), mientras que la región no ortogonal es interpolada a partir de la región ortogonal lo cual aumenta la complejidad de la estimación y reduce la precisión, debido, a que las frecuencias de las subportadoras de la región OFDM no son las mismas que para la región SEFDM [12].

La mayoría de las técnicas de estimación para sistemas SEFDM, utilizan una parte de ortogonalidad para determinar cual es el comportamiento del canal. Al encontrar un esquema que se beneficie del concepto de subportadoras comprimidas en los sistemas SEFDM tendrá ganancias en rendimiento y complejidad computacional sobre otros esquemas existentes.

### **1.4.2.1 Símbolos piloto y preámbulo**

Un canal puede ser estimado utilizando símbolos pilotos, estos símbolos pilotos deben ser conocidos tanto en el transmisor como en el receptor, los mismos son transmitidos mediante subportadoras piloto. Los símbolos transmitidos sirven como entrenamiento en la recepción para determinar como se comporta el canal, posteriormente mediante un proceso ecualización se pueden corregir los errores detectados [11].

El preámbulo son un grupo símbolos conocidos tanto por el transmisor como por el receptor, son enviados al principio de la comunicación, estos símbolos poseen una buena correlación entre sí, de esta manera, se logra conseguir estimación de canal al comparar los símbolos con ruido que llegan del canal, con los símbolos que conoce el receptor. Dependiendo de la técnica de estimación se puede mejorar la tasa de errores.

## 2 METODOLOGÍA

En el presente capítulo se muestra el desarrollo para la simulación de un sistema SEFDM con estimación de canal en el software de MATLAB. El sistema implementado consta de: un transmisor SEFDM, un canal de tipo Rician con ruido AWGN y un receptor SEFDM, este último tendrá un mayor enfoque, ya que, en la etapa de recepción se implementa la técnica de estimación de canal para sistemas SEFDM. Además, se explicará a detalle las funciones y bloques, tanto, del transmisor, canal y receptor.

Para el desarrollo del código de simulación en la etapa de transmisión se toma en cuenta: un codificador convolucional con una tasa de 1/2 en conjunto con un codificador FEC (*Forward Error Correction*), un modulador con diferentes tipos de modulación como: QPSK, 16QAM y 64QAM, y un preámbulo utilizado para la estimación de canal. Para el canal inalámbrico se añade ruido AWGN y desvanecimiento, siendo un canal de tipo Rician. En el receptor mediante el preámbulo que se recupera, se realiza una técnica de estimación de canal, el resultado del proceso de estimación se dirige a un bloque detector, para la compensación del canal inalámbrico utilizado. Finalmente se obtiene una gráfica con los valores de BER vs  $E_b/N_0$  para un sistema SEFDM con estimación de canal.

Para el presente trabajo se define un enfoque de tipo cuantitativo, esto debido, a que los resultados pueden ser cuantificados mediante los parámetros de BER y  $E_b/N_0$ . De la misma manera se define un alcance descriptivo en su mayoría, debido, a que presenta una revisión de los sistemas y de técnicas de estimación de canal para SEFDM. Además, presenta un enfoque correlacional, al tener parámetros que pueden ser configurados en la transmisión como son: el tipo de modulación o el factor de compresión, el cambio de estas variables de configuración tendrá una afectación sobre los valores del BER.

### 2.1 PROGRAMA PRINCIPAL

El desarrollo del programa principal fue diseñado para la simulación de un sistema SEFDM con estimación de canal mediante el software de MATLAB. El programa principal de nombre *SEFDM\_VARIOS\_ALFA\_RICIAN*, está compuesto por: ingreso de variables, función *SEFDM\_Rician* y gráfica de resultados.

#### 2.1.1 INGRESO DE VARIABLES

Es necesario definir parámetros que permiten realizar la simulación de un sistema de comunicación SEFDM y entre ellos están: el número de subportadoras, el factor de compresión, el tipo de modulación, el número de repeticiones de la simulación, el valor del

$E_b/N_0$  o el número de símbolos que contiene cada paquete. Las variables de ingreso para el sistema se visualizan en la tabla 2.1.

**Tabla 2.1.** Variables de ingreso del sistema SEFDM.

Variable	Definición
N	Número de subportadoras
m	Tipo de modulación: QPSK, 16-QAM y 64-QAM
a	Ceros añadidos (Concepto SEFDM)
alfa	Factor de Compresión
Ntrials	Número de iteraciones
$E_b/N_0$	Valor de $E_b/N_0$
N_sym	Número de símbolos por paquete

## 2.2 FUNCIÓN SEFDM\_RICIAN

La función *SEFDM\_Rician* tiene partes de código para la etapa de transmisión, el canal inalámbrico y la etapa de recepción. Por lo cual, *SEFDM\_Rician* tiene un rol fundamental para la simulación del sistema SEFDM. Dicha función se define en el segmento de código 2.1. Los parámetros que necesita la función están definidos en la sección de ingreso de variables.

```

% INGRESO DE VARIABLES
%Datos del transmisor
N=16;           %Número de Subportadoras
m=4;           %Número de bits agrupados según la modulación
               % 2-->QPSK | 4--->16QAM | 6-->64QAM
a=[0,2]; %Número de ceros añadidos (Definición SEFDM)
alfa=(N./(N+a)); %Definición del Factor de Compresión de AB

%Datos para la obtención de gráficas
Ntrials=20;    %Número de iteraciones para cada valor de SNR
EbNo=0:2:30;  %Eb/No
N_sym=5;      %numero de símbolos por paquete

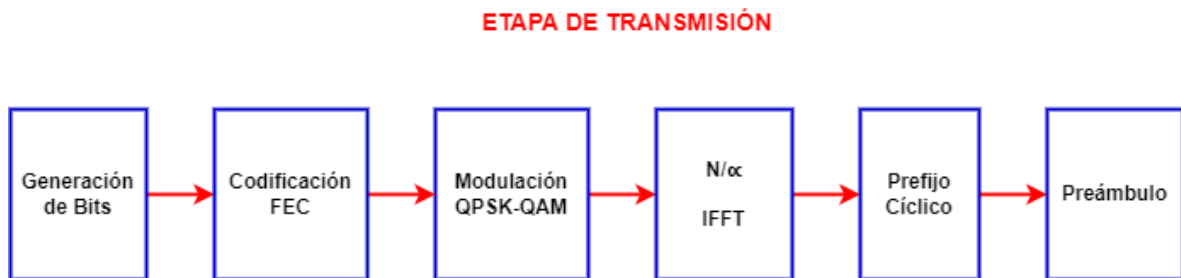
% OBTENCIÓN DE LOS VECTORES BER PARA DISTINTOS VALORES DE ALFA
%Uso de la función SEFDM
% Para el último valor de la función SEFDM define:
% 1--> Estimación Perfecta | 2--> Tecnica de estimación OFDM-SEFDM
BER(1,:)=SEFDM_Rician(N,m,alfa(1),Ntrials,EbNo,N_sym,1); %Alfa=N/N+0
BER(2,:)=SEFDM_Rician(N,m,alfa(2),Ntrials,EbNo,N_sym,1); %Alfa=N/N+2
BER(3,:)=SEFDM_Rician(N,m,alfa(1),Ntrials,EbNo,N_sym,2); %Alfa=N/N+0
BER(4,:)=SEFDM_Rician(N,m,alfa(2),Ntrials,EbNo,N_sym,2); %Alfa=N/N+2

```

**Segmento de código 2.1.** Ingreso de variables y función SEFDM\_Rician.

## 2.2.1 ETAPA DE TRANSMISIÓN SEFDM

La etapa de transmisión de un sistema de comunicaciones SEFDM se presenta en la figura 2.1 donde constan los bloques de: generación de bits, codificación FEC, modulación QPSK o QAM, IFFT, prefijo cíclico y preámbulo. A continuación, se detalla cada bloque con el respectivo código utilizado.



**Figura 2.1.** Esquema de la etapa de Transmisión.

### 2.2.1.1 Generación de bits y codificación FEC

El primer bloque de la transmisión es la generación de bits y se define por: el número de subportadoras, la tasa de codificación, el tipo de modulación y el número símbolos por paquete. Para el código implementado se tiene 16 subportadoras, una tasa de codificación de 1/2, el tipo de modulación es un parámetro configurable por el usuario puede ser 2, 4 o 6, y el número de símbolos por paquete es 5. Se utiliza la función *randi* para generación aleatoria de unos o ceros que son guardados por la variable *bit\_tx\_s*, como se visualiza en el segmento de código 2.2.

```
%TRANSMISOR SEFDM
bit_tx_s=randi([0,1],N*m*tc*N_sym,1);      %Generación de bits
data_cod = convEncoder(bit_tx_s);          %codificación FEC
```

**Segmento de código 2.2.** Bloque de generación de bits y Bloque de codificación FEC.

Mediante la variable *data\_cod* se obtienen los valores, luego de la codificación FEC, la cual se define en el segmento de código 2.3 por la función *convEncoder*. Con el proceso de codificación FEC se añaden bits de redundancia, teniendo en cuenta que se utiliza el objeto *comm.ConvolutionalEncoder*, el cual es un codificador convolucional, que utiliza una estructura de *trellis*. Además, se utiliza *traceBack* que es un parámetro del codificador FEC de valor 32. Los valores corresponden a un codificador con tasa de 1/2 [13].

```

%codificación
tc=1/2; %tasa codificación
traceBack = 32; %Parámetro de codificador FEC
trellis = poly2trellis(7,[171 133]);
%codificador convolucional
convEncoder = comm.ConvolutionalEncoder('TrellisStructure',trellis);

```

**Segmento de código 2.3.** Codificación FEC.

### 2.2.1.2 Modulación QPSK-QAM

Para el nivel de modulación se tiene la variable  $M = 2^m$  donde el valor de  $m$  es seleccionado por el usuario, el cual puede ser 2, 4 o 6. Por tanto, para una modulación QPSK,  $m$  debe ser igual a 2, para una modulación 16 QAM,  $m$  debe ser igual a 4 y para una modulación 64 QAM,  $m$  debe ser igual a 6 [14].

Para la simulación en el software de MATLAB existe la función *qammod*, la misma realiza la modulación de un vector de datos, la señal de entrada *data\_cod*, con el nivel de modulación definido por  $M$ . Además, el parámetro *InputType* selecciona el tipo de dato que es ingresado, en este caso, binario y *UnitAveragePower* es para configurar la señal de salida con una potencia promedio de 1 W. La función de modulación se observa en el segmento de código 2.4.

```

%Modulación digital
tx_mod=qammod(data_cod,M,'InputType','bit',...
              'UnitAveragePower',true);

```

**Segmento de código 2.4.** Bloque de Modulación.

### 2.2.1.3 IFFT y prefijo cíclico

Partiendo de la definición de SEFDM, se tiene que el valor de la variable *IFFT\_FFT\_size* es  $\frac{N}{\alpha}$  siendo  $N$  las 16 subportadoras, y  $\alpha$  el factor de compresión definido por el usuario. Se crea un vector *ceros*, para añadirlos en el bloque IFFT como se muestra en el segmento de código 2.5.

El lazo *for* se repite por la variable *N\_sym* es decir por valor del número de símbolos por paquete. Se define una variable *temp* la cual toma 16 valores en orden del vector de datos modulados *tx\_mod* para cada interacción del lazo *for*. A los 16 valores se le añade el vector de *ceros* dando como resultado el vector *tx\_sefdm*, este vector pasa al bloque IFFT, por tanto, el vector resultante *iff\_t\_x* tiene una dimensión de tamaño *IFFT\_FFT\_size*. Se



desconocen los últimos valores de los ceros añadidos, dejando un vector  $tx\_out$  de tamaño  $1 \times N$  es decir  $1 \times 16$ .

Para la inserción de prefijo cíclico se toma un valor de  $\frac{N}{4}$  es decir 4. Los últimos 4 valores del vector  $tx\_out$  son copiados al principio del vector  $tx\_out\_cp$  y posteriormente se añade el vector  $tx\_out$ , por lo que se añade el PC para cada símbolo en cada interacción del lazo *for*. Finalmente, en el vector  $tx\_out\_s$  se guardan todos los símbolos de datos generados [13].

```

ceros=zeros(1,IFFT_FFT_size-N);
tx_mod_s=tx_mod.';

for jj=1:N_sym
    temp=tx_mod_s((jj-1)*N+1:jj*N);
    tx_sefdm=[temp ceros];           % SEFDM
    ifft_tx=ifft(tx_sefdm,IFFT_FFT_size); %Bloque IFFT
    tx_out=ifft_tx(1:N);           %Desconocer los últimos valores
    tx_out_cp=[tx_out(N-(N/4)+1:end) tx_out]; %CP
    tx_out_s((jj-1)*(N+N/4)+1:jj*(N+N/4))=tx_out_cp;
end

```

**Segmento de código 2.5.** Lazo de IFFT para transmisión.

#### 2.2.1.4 Preámbulo

En el segmento de código 2.6 se genera un preámbulo en el dominio de la frecuencia dado por el vector  $pream\_f$ , el vector tiene una dimensión de  $1 \times 16$  y los valores que contiene son:  $[1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1]$ . Posteriormente, el vector pasa a un bloque de IFFT, con un tamaño  $N$  es decir 16. Finalmente, en la variable  $pream\_i\_cp$  se añade el PC, pero para los valores del preámbulo, el cual sigue el mismo concepto de los símbolos de los datos explicados anteriormente.

```

%Preámbulo DT
pream_f = [1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1]; %Preámbulo DF
pream_i=ifft(pream_f,N); %Bloque IFFT
pream_i_cp=[pream_i(N-(N/4)+1:end) pream_i]; %CP

```

**Segmento de código 2.6.** Bloque Preámbulo.

El preámbulo es añadido al comienzo del vector  $tx\_out\_s$ , es decir el vector  $tx\_out\_s$  tiene el preámbulo seguido de la señal de datos. El vector  $tx\_out\_s$  se encuentra en el dominio del tiempo para ser transmitido por el canal. Como se observa en el segmento de código 2.7.

```

%Agregación de Preámbulo cuando existe método de estimación SEFDM-OFDM
if opc==2
    tx_out_s=[pream_i_cp tx_out_s]; %Transmisión de datos con Preámbulo
end

```

**Segmento de código 2.7.** Vector de Transmisión.

## 2.2.2 CANAL INALÁMBRICO

El canal inalámbrico está compuesto de un canal con desvanecimiento de Rician y ruido blanco AWGN. Para el desvanecimiento de tipo Rician se utiliza el objeto *comm.RicianChannel*; el cual filtra una señal de entrada a través de un canal con desvanecimiento de múltiples trayectos. Los parámetros utilizados para la creación del canal Rician se observan en la tabla 2.2.

**Tabla 2.2.** Parámetros de canal tipo Rician.

Variable	Definición
<i>SampleRate</i>	Representa la frecuencia de muestreo de la señal de entrada medida en hercios.
<i>PathDelays</i>	Representa el retardo de la ruta discreta. Al usar un vector como entrada, el canal tiene frecuencia plana ( <i>frequency flat</i> ).
<i>AveragePathGains</i>	Es la ganancia promedio de las rutas discretas en decibelios.
<i>MaximumDopplerShift</i>	Es el desplazamiento Doppler máximo para todas las rutas de canal, cuando se establece la propiedad en 0, el canal permanece estático durante toda la entrada.
<i>PathGainsOutputPort</i>	Es la ganancia de la ruta del canal de salida.
<i>KFactor,</i>	Representa el factor K del canal de desvanecimiento de tipo Rician, es el número de retardos en la ruta.

En el segmento de código 2.8 se visualiza la creación de la función *ricianchan*. La función representara el canal tipo Rician con los parámetros descritos en la tabla 2.2 y los valores utilizados con estos parámetros.

```

%DEFINICIÓN DE PARÁMETROS DEL CANAL DE RICIAN
%Desvanecimiento con distribución de Rician
srate=20e6; %Frecuencia de muestreo
delay=[0 2]*50e-9; %Retardos en la ruta[ns]
gains=[0 -9.7]; %Ganancias de trayectoria prom.
ricianchan = comm.RicianChannel('SampleRate',srate,'PathDelays',delay,...
    'AveragePathGains',gains,'MaximumDopplerShift',0,...
    'PathGainsOutputPort',true,'KFactor',10);
ps=round(delay/(1/srate))+1; %Cálculo de las posiciones
    %para las ganancias de trayecto
    %Características del canal de Rician

```

### Segmento de código 2.8. Generación del canal Rician.

Mediante la función *ricianchan*, definida previamente en el segmento de código 2.9 por el objeto *comm.RicianChannel*, se aplica el canal Rician a la señal de datos, *tx\_out\_s* y se genera un nuevo vector de datos *tx\_ray*, el cual ya contiene componentes del desvanecimiento del canal Rician.

```

%Aplicación del Canal de RICIAN
[tx_ray,pgains]=ricianchan(tx_out_s.);

```

### Segmento de código 2.9. Aplicación del canal Rician.

En el segmento de código 2.10 mediante la variable *tx\_ray\_s* se utiliza la función *reshape*, la misma permite rehacer el vector de entrada *tx\_ray*, para ser de una sola fila, es decir, un conversor de paralelo a serie. Posteriormente, para agregar ruido gaussiano blanco a la señal se utiliza la función *awgn*. Mediante la variable *Snr(i)*, se escoge la relación señal-ruido en dB, dado por el parámetro *db*. Con el parámetro *measured*, el nivel de la señal de entrada se calcula para determinar el nivel de ruido adecuado en función del valor de SNR (*Signal to Noise Ratio*).

```

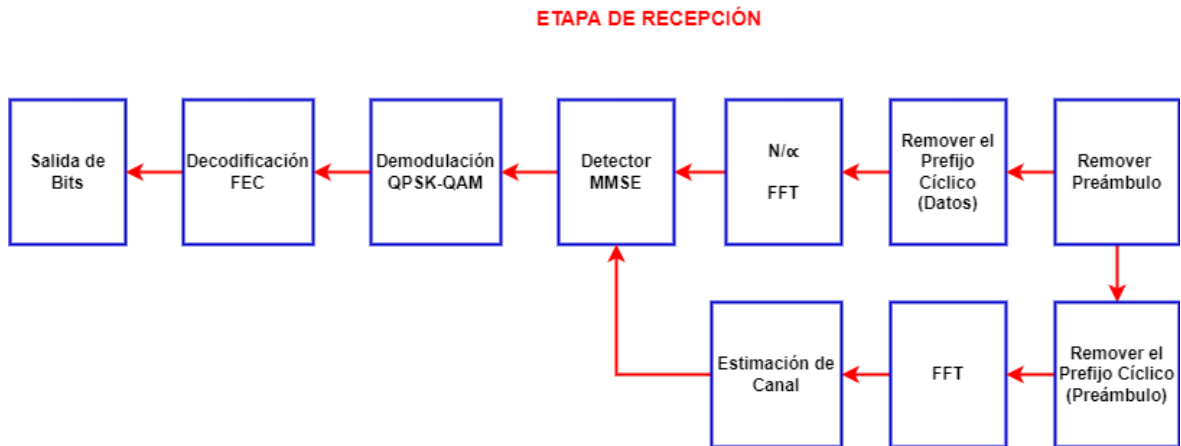
%Canal AWG
tx_ray_s=reshape(tx_ray,1,length(tx_ray));
canal_rx=awgn(tx_ray_s,Snr(i),'measured','db');

```

### Segmento de código 2.10. Aplicación del ruido AWGN.

## 2.2.3 ETAPA DE RECEPCIÓN SEFDM

El diagrama de bloques de la etapa recepción de un sistema de comunicaciones SEFDM se presenta en la figura 2.2; donde constan los bloques de: remover preámbulo, remover prefijo cíclico,  $\frac{N}{\alpha}$  FFT, estimación de canal, detector MMSE, Demodulador QPSK-QAM, decodificador FEC y salida de bits.



**Figura 2.2.** Esquema de la etapa de Recepción.

### 2.2.3.1 Estimación de canal SEFDM

Para el código de simulación se tendrá una estimación de canal perfecta y una estimación de canal utilizando el método SEFDM-OFDM, el cual se presenta en [12]. Para la estimación de canal perfecta se utiliza la siguiente ecuación:

$$H = F' x h x F. \quad (2.1)$$

La matriz de canal  $H$  se define a partir de la matriz de subportadoras  $F$ , la matriz del canal equivalente en el dominio del tiempo  $h$ , y la matriz de subportadoras conjugada  $F'$ . Al ser un sistema SEFDM la matriz de subportadoras se caracteriza por la pérdida de la ortogonalidad. Mientras, la matriz del canal en el dominio del tiempo representa el efecto del canal; para esta simulación un canal con desvanecimiento Rician. La matriz  $h$  se consigue mediante las ganancias de trayecto del canal. Estos valores se ubican en las diagonales definidas por la variable  $ps$ , representada por la ecuación:

$$ps = round \left( \frac{delay}{\frac{1}{srate}} \right) + 1. \quad (2.2)$$

Para los datos en la simulación para el sistema SEFDM con estimación perfecta se tiene un valor de  $ps$ . Un ejemplo se puede observar en la Figura 2.3 donde se realiza la creación de la matriz  $h$  de tamaño  $4 \times 4$ , con valores de  $ps = [1, 3]$  con valores de  $pgains = [pg_1, pg_2]$  [13].

$$h_{4 \times 4} = \begin{bmatrix} pg_1 & 0 & pg_2 & 0 \\ 0 & pg_1 & 0 & pg_2 \\ pg_2 & 0 & pg_1 & 0 \\ 0 & pg_2 & 0 & pg_1 \end{bmatrix}$$

**Figura 2.3.** Matriz de en el dominio del tiempo [13].

Para la creación de la matriz  $h$ , se deben sumar las matrices auxiliares  $D1, D2$  y  $D3$ . Como ejemplo se puede visualizar las matrices auxiliares  $D1, D2$  y  $D3$  en la figura 2.4. El código completo para generar la matriz  $h$  se presenta en el segmento de código 2.11.

$$D1 = \begin{bmatrix} pg_1 & 0 & 0 & 0 \\ 0 & pg_1 & 0 & 0 \\ 0 & 0 & pg_1 & 0 \\ 0 & 0 & 0 & pg_1 \end{bmatrix} \quad D2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ pg_2 & 0 & 0 & 0 \\ 0 & pg_2 & 0 & 0 \end{bmatrix} \quad D3 = \begin{bmatrix} 0 & 0 & pg_2 & 0 \\ 0 & 0 & 0 & pg_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

**Figura 2.4.** Matrices de diagonales auxiliares [13].

```

%Obtención de la matriz del Canal
temp=pgains(1,:);
d1=zeros([1 N]);
d2=zeros([1 N-2]);
d3=[temp(2) temp(2)];
d1(:)=temp(1);
d2(:)=temp(2);
D1=diag(d1);D2=diag(d2,-(ps(2)-1));
D3=diag(d3,N-(ps(2)-1));h=D1+D2+D3; %Idealmente

```

**Segmento de código 2.11** Obtención de la matriz  $h$ .

La matriz de subportadoras  $F$  y la matriz de subportadoras conjugadas  $F'$  se fundamentan en la ecuaciones 2.3 y 2.4, respectivamente. Mediante el segmento de código 2.12 se visualiza como se crea la matriz de subportadoras  $F$ , para la matriz de subportadoras conjugadas en el software de MATLAB se realiza la operación transpuesta ( $'$ ) que genera la matriz transpuesta conjugada.

$$F_{k,n} = \frac{1}{\sqrt{N}} e^{\frac{j2\pi\alpha nk}{N}}. \tag{2.3}$$

$$F'_{k,n} = \frac{1}{\sqrt{N}} e^{-\frac{j2\pi\alpha nk}{N}}. \tag{2.4}$$

```

%MATRIZ DE SUBPORTADORAS F para diferentes alfas
F=zeros(N,N);
for k=0:N-1
    for n=0:N-1
        F(k+1,n+1)=1/sqrt(N)*exp(1i*2*pi*n*alfa*(k/N));
    end
end

```

**Segmento de código 2.12.** Matriz de subportadoras  $F$ .

Para la estimación de canal utilizando el método SEFDM-OFDM, primero se retira el preámbulo que ha pasado por el canal, mediante la variable *prambulorecepcion* se guardan los valores del preámbulo. Posteriormente, en la variable *temp3* se retira el PC del preámbulo. Finalmente se aplica un bloque FFT a los valores del preámbulo, con un tamaño de  $N$  es decir de  $1 \times 16$ . Luego, en la variable  $H$  se realiza la función *estimación\_ls* con las variables de entrada  $Rx\_pre$  que son los datos del preámbulo, pero en el dominio de la frecuencia luego del bloque FFT, *pream\_f* que son los datos del preámbulo de la transmisión,  $F2$  que es una matriz de subportadoras para OFDM, y  $F$  que es la matriz de subportadoras para SEFDM con diferentes alfas.

```

% Matriz del Canal
if opc==1
    H=F'*h*F; % Estimación Perfecta
else
    %Quitar preámbulo
    preambulorecepcion=canal_rx(1:N+(N/4)); %Preámbulo
    temp3=preambulorecepcion((N/4)+1:N+(N/4)); % Quitar PC
    Rx_pre=fft(temp3,N); %Bloque FFT

    %Estimación de Canal, método SEFDM-OFDM
    H=estimacion_ls(Rx_pre,pream_f,F2,F); %Función Estimación

    %Datos del canal sin preámbulo
    canal_rx=canal_rx(N+(N/4)+1:end); % Símbolos
end

```

**Segmento de código 2.13.** Estimación Perfecta o Método de estimación SEFDM-OFDM.

Dentro de la función *estimacion\_ls* se procede a la estimación de canal LS descrito en [12]. Existen dos métodos en el dominio de la frecuencia utilizados: Pilotos OFDM con interpolación y Pilotos OFDM sin interpolación. El método utilizado para la función *estimacion\_ls* es Pilotos OFDM sin interpolación; este método pretende utilizar los pilotos de OFDM, para resolver el problema de la pérdida de ortogonalidad de SEFDM.

Pilotos OFDM con interpolación se basa en ocupar el ancho de banda de SEFDM con un número de subportadoras piloto OFDM ( $N_p$ ) para un determinado ( $\alpha$ ). Dado por la función:

$$Np = [\alpha x N]. \quad (2.5)$$

siendo  $N$  el número de subportadoras del sistema SEFDM. Para  $\alpha = \frac{b}{c}$  donde  $b, c \in \mathbb{Z}$ ,  $b < c$  y  $N > c$ . Existen  $\left\lceil \frac{N}{c} \right\rceil$  subportadoras de  $Np$  con un múltiplo de  $(\alpha \Delta f)$  siendo  $\Delta f$  separación de subportadoras en frecuencia. Por lo tanto, en estas subportadoras se puede hacer directamente estimación LS. A partir de ahí, el resto de las subportadoras  $(N - \left\lceil \frac{N}{c} \right\rceil)$  se obtienen mediante interpolación. Sin embargo, este método tiene limitantes para el factor de compresión  $\alpha$  debido a los valores de  $b$  y  $c$ . Para disminuir la complejidad en el esquema de estimación y eliminar la limitante  $\alpha$  se utiliza el método sin interpolación.

El objetivo del método sin interpolación es tener símbolos pilotos OFDM que ocupen el mismo ancho de banda que los símbolos de información para un sistema SEFDM, con el mismo número de subportadoras  $N$  y separación de frecuencia. Si se cumplen los requisitos anteriores, se puede realizar estimación del canal directamente a través de LS sin interpolación, como se indica a continuación:

$$h_{est} = \hat{P} ./ P. \quad (2.6)$$

donde  $./$  denota una división para todos los elementos de los vectores,  $\hat{P}$  representa un símbolo recuperado en la recepción y  $P$  representa un símbolo transmitido. Por lo tanto, se puede mantener la ortogonalidad cuando se cumple una separación de subportadoras de  $\Delta f = \frac{\alpha}{T}$ . La duración del símbolo del piloto OFDM ( $Tp$ ) aumenta con un factor de  $(\frac{1}{\alpha})$ , por tanto  $Tp = \frac{T}{\alpha}$ . En el segmento de código 2.14 se presenta el código para la estimación de canal LS [12].

```

% Función de Estimación
function H=estimacion_ls(Rx_pre,pream_f,F2,F)

    Form=Rx_pre./pream_f;    % Estimación de canal LS procedimiento
                             % descrito en el paper
    h_int_diag=diag(Form);    % Matriz de canal estimada sin efecto de
                             % interferencia en dominio de la frecuencia.
    h_est=F2*h_int_diag*F2'; % Matriz de canal estimada en el dominio
                             % del tiempo.
    H=F'*h_est*F;           % Generación de matriz del canal dominio
                             % de la frecuencia
end

```

**Segmento de código 2.14.** Función de estimación de canal LS.

En la variable *Form* se guarda el resultado de la división entre el preámbulo recibido y el preámbulo transmitido. Luego, se genera la variable *h\_init\_diag* con la diagonal de la variable *Form*; esta variable *h\_init\_diag* es la matriz de canal estimada sin efecto de interferencia en el dominio de la frecuencia. El canal estimado sin efecto de interferencia en el dominio de la frecuencia puede representarse mediante la fórmula:

$$H_{SINT} = F \times h_{SINT} \times F'. \quad (2.7)$$

siendo  $h_{SINT}$  una variable desconocida se puede despejar de la ecuación 2.7. En el segmento de código 2.14. la variable  $H_{SINT}$  corresponde a *h\_init\_diag*. Además,  $F'$  equivale a  $F^{-1}$ . En las ecuaciones 2.8-2.11 se presenta el despeje de la variable  $h_{SINT}$ .

$$F^{-1} \times H_{SINT} = F^{-1} \times F \times h_{SINT} \times F', \quad (2.8)$$

$$F' \times H_{SINT} \times F = I \times h_{SINT} \times F' \times F, \quad (2.9)$$

$$F' \times H_{SINT} \times F = I \times h_{SINT} \times I, \quad (2.10)$$

$$h_{SINT} = F' \times H_{SINT} \times F. \quad (2.11)$$

En la ecuación 2.11 se encuentra la fórmula de  $h_{SINT}$  que corresponde a una matriz sin efecto de interferencia estimada para el dominio del tiempo. Dado que para el método de estimación de canal utilizado se realiza con pilotos OFDM, las matrices de subportadoras  $F$  y la matriz de subportadoras  $F'$  son para un sistema OFDM, por lo que, el alfa es igual a 1. Denominaremos a la matriz de subportadoras para OFDM  $F_2$ , entonces  $h_{SINT}$  se expresa como:

$$h_{SINT} = F_2' \times H_{SINT} \times F_2. \quad (2.12)$$

Las matrices de subportadoras y subportadoras conjugadas para  $F_2$  son:

$$F_{2\ k,n} = \frac{1}{\sqrt{N}} e^{\frac{j2\pi nk}{N}}. \quad (2.13)$$

$$F_{2'\ k,n} = \frac{1}{\sqrt{N}} e^{-\frac{j2\pi nk}{N}}. \quad (2.14)$$

En el segmento de código 2.15 se observa la generación de la matriz de subportadoras  $F_2$ . Al igual que para la matriz de subportadoras conjugadas  $F'$ , para generar la matriz de subportadoras conjugada  $F_2'$ , en el software de MATLAB se realiza la operación transpuesta ( $'$ ).



```

%MATRIZ DE SUBPORTADORAS F para diferentes alfa=1
F2=zeros(N,N);
for k=0:N-1
    for n=0:N-1
        F2(k+1,n+1)=1/sqrt(N)*exp(1i*2*pi*n*(k/N));
    end
end
end

```

**Segmento de código 2.15.** Matriz de Subportadoras  $F_2$ .

Finalmente, hay que pasar la matriz  $h_{SINT}$  al dominio de la frecuencia. En el código  $h_{SINT}$  es la variable  $h\_est$  y tomando en cuenta la ecuación 2.1, la matriz de canal estimada en el dominio de la frecuencia está dada por:

$$H = F x h_{SINT} x F'. \quad (2.15)$$

### 2.2.3.2 Lazo de recepción de datos SEFDM

Para la recepción de datos SEFDM se puede visualizar el segmento de código 2.16. El lazo *for* repite el valor de los símbolos por paquetes dados en la variable  $Nsym$ . Como primer paso se remueve el prefijo cíclico. En la variable  $temp$  se guardan las muestras de datos, cuando se retiró el PC del vector de datos  $canal\_rx$ .

En la variable  $rx\_sef\_dm\_vec$  se realiza la definición de  $\frac{N}{\alpha}$  FFT para los sistemas SEFDM, dicha variable está compuesta por los valores de la variable  $temp$  y la variable  $ceros$ . De esta manera, se puede realizar la FFT de  $rx\_sef\_dm\_vec$  de longitud  $\frac{N}{\alpha}$  es decir, la longitud de  $IFFT\_FFT\_size$ . En la variable  $fft\_rx$  se guardan los datos luego de la FFT, es decir, cuando los datos han pasado del dominio del tiempo al dominio de la frecuencia.

Mediante la variable  $Rx$ , se desconocen los ceros añadidos. En la variable  $Rx\_p$  se realiza una conversión paralelo a serie de las muestras de  $Rx$ . La dimensión de la variable  $Rx\_p$  debe ser igual a la dimensión de las subportadoras  $N$  ( $1 \times 16$ ).

Se genera una matriz identidad de dimensiones  $N \times N$ . Se utiliza la función  $MMSE$ , la cual, se explica a detalle en la sección de Detección MMSE. Finalmente se genera la variable  $rx\_det\_s$  encargada de guardar los valores de las muestras de cada símbolo recuperado luego del bloque de detección MMSE. La dimensión de la variable  $rx\_det\_s$  depende de la dimensión de la variable  $Rx\_p$  y el número de símbolos por paquetes definidos en  $Nsym$  [13].

```

for jj=1:N_sym
temp=canal_rx((jj-1)*(N+N/4)+1+N/4:jj*(N+N/4));
rx_sefdm_vec=[temp,ceros]; %Definición FFT para SEFDM
fft_rx=fft(rx_sefdm_vec,IFFT_FFT_size); %Bloque FFT
Rx=fft_rx(1:N); %Desconocer los últimos valores
Rx_p=Rx.';
I=eye(N); %Matriz Identidad
rx_det=MMSE(Rx_p,I,Snr(i),H,pt/(N*4)); %Detector MMSE
rx_det_s((jj-1)*N+1:jj*N,1)=rx_det.';
end

```

**Segmento de código 2.16.** lazo de recepción del paquete de datos SEFDM.

### 2.2.3.3 Detección MMSE

El algoritmo MMSE aparece con el fin de reducir el impacto del ruido en la detección, no solo emplea la matriz del canal, sino que además añade un término relacionado con el ruido. El bloque MMSE necesita información estadística de la varianza del ruido  $\sigma^2$  y la potencia de la señal recibida  $\sigma_s^2$ . La relación  $\frac{\sigma^2}{\sigma_s^2}$  representa la inversa de la SNR por la potencia promedio de la señal recibida [13].

El algoritmo MMSE incorpora parámetros de entrada como son: señal de entrada, una matriz identidad, el valor de SNR, la matriz de canal y la potencia promedio de la señal recibida. Los parámetros que utiliza la función de detección MMSE se detallan en la tabla 2.3.

**Tabla 2.3.** Parámetros de entrada para detector MMSE

Variable	Definición
$R$	Señal de entrada
$I$	Matriz identidad
$snrdb$	Valor de SNR en dB
$H$	Matriz del canal
$pt$	Potencia promedio de la señal recibida

Con el argumento de entrada  $snrdb$ , se obtiene el valor de SNR en veces en la variable  $snr$ , luego se aplica el ecualizador MMSE que se observa en la ecuación 2.16. Finalmente, se obtiene la señal detectada  $S$  del producto entre  $W$  y la señal de entrada  $R$ , que se muestra en la ecuación 2.17.

$$W_{MMSE} = \left[ H^H (H H^H + \frac{\sigma^2}{\sigma_s^2} I) \right]^{-1}. \quad (2.16)$$

$$S_{MMSE} = W_{MMSE} \times R. \quad (2.17)$$

En el segmento de código 2.17 se muestra la función de detección MMSE, donde mediante la variable  $W$  se representa la ecuación 2.15, la matriz de canal de la variable  $H$  se obtiene del bloque de estimación perfecta o del método de estimación implementado. Se recupera la variable  $Sv$  la cual contiene los símbolos de los datos recuperados luego de la detección MMSE luego del producto de la variable  $W$  con los símbolos de los datos que salen del bloque FFT [13].

```

%Esta función tiene como propósito la detección de la señal
%enviada con el método MMSE utilizando la matriz del canal H.

function [Sv] = MMSE(R,I,snrdb,H,pt)

snr=10^(snrdb/10);           %Valor de SNR en veces
W=((H'*H)+(pt/snr)*I)\H';   %Aplicación de la definición
                             %del detector MMSE
Sv=W*R;                     %Obtención de símbolos

end

```

**Segmento de código 2.17.** Función MMSE.

### 2.2.3.4 Demodulación y decodificación FEC

Para el bloque de la demodulación se debe tener en cuenta el nivel de la modulación  $M = 2^m$ . Para el código de MATLAB se usa la función `qamdemod`, la misma realiza el proceso de demodulación de un vector de símbolos `rx_det_s`. El vector de entrada proviene del bloque de detección MMSE. Además, el parámetro `OutputType` selecciona el tipo de dato de salida, en este caso, por defecto `integer`. `UnitAveragePower` es para configurar la señal de salida con una potencia promedio de 1 W. `NoiseVariance` se utiliza para especificar el valor de varianza de ruido con el mismo valor para todos los elementos de entrada y `approxllr` representa el valor aproximado de relación logarítmica de similitud para cada bit. Los valores se calculan utilizando el algoritmo de probabilidad logarítmica aproximada. El bloque de demodulación se observa en el segmento de código 2.18.

```

%Demodulación digital
rx_dem=qamdemod(rx_det_s,M,'OutputType','approxllr',...
               'UnitAveragePower',true,'NoiseVariance',noiseVar);

```

**Segmento de código 2.18.** Bloques de demodulación.

Para la decodificación FEC se utiliza el objeto *comm.ViterbiDecoder*, el cual decodifica datos convolucionalmente. Los parámetros utilizados para este objeto se visualizan en la tabla 2.4.

**Tabla 2.4.** Parámetros utilizados en el objeto *comm.ViterbiDecoder*.

Variable	Definición
<i>TrellisStructure</i>	Especifica el uso de la estructura Trellis del código convolucional.
<i>InputFormat</i>	Es el formato de entrada del decodificador.
<i>unquantized</i>	Los datos de entrada deben ser un vector de valor real con precisión simple o doble, que no estén cuantificados. El objeto asigna valores positivos de 1s lógicos y valores negativos de 0s lógicos.
<i>TracebackDepth</i>	Se especificada como un número entero. <i>TracebackDepth</i> influye en la precisión y el retraso de la decodificación.

Para la decodificación FEC se define la función *vitDecoder* del objeto *comm.ViterbiDecoder*. El cual usa un algoritmo de Viterbi. La función *vitDecoder* está definida en el segmento de código 2.19.

```
vitDecoder = comm.ViterbiDecoder('TrellisStructure',trellis, ...
    'InputFormat','unquantized','TracebackDepth',traceBack);
```

**Segmento de código 2.19.** Función *vitDecoder*.

En el segmento de código 2.20 se puede visualizar la variable *rx\_bit\_dec*; la cual representa los símbolos de la recepción luego del bloque de decodificación FEC. La variable *rx\_bit\_dec* permite realizar el proceso para el cálculo de errores.

```
rx_bit_dec = vitDecoder(rx_dem);           %decodificación FEC
```

**Segmento de código 2.20.** Bloque de decodificación.

### 2.2.3.5 Cálculo de errores

Para el cálculo del BER, se utiliza la función de MATLAB *biterr* que permite comparar los símbolos de la transmisión con los recuperados en la recepción, siendo estos una representación binaria; se compara los elementos *bit\_tx\_s* de la transmisión con los de *rx\_bit\_dec* de la recepción. La función devuelve *n\_error* que es el número de bits que difieren en la comparación y *ratio* que es la relación entre los errores encontrados y el número total de bits. El cálculo de la tasa de errores (BER) se visualiza en el segmento de código 2.21.

```
%Número de Errores
[n_error,ratio]=biterr(bit_tx_s(1:end-traceBack),...
    rx_bit_dec(traceBack+1:end));

%Matriz de tasa de errores
BER_T(j,i)=ratio;
```

**Segmento de código 2.21.** Cálculo de tasa de errores (BER).

En el segmento de código 2.22 se hace el promedio de la tasa de errores BER. Mediante la función *mean*, se devuelve la media o el valor promedio de los elementos de *BER\_T* el cual se define por la relación entre el número de errores en los bits de entrada y salida, con respecto a los bits totales, dado por la variable *ratio*.

```
%Promedio de tasa de Errores
BER=mean(BER_T);
```

**Segmento de código 2.22.** Promedio de BER.

## 2.3 GRÁFICA DE RESULTADOS

Dentro del software de MATLAB, existen varias funciones que sirven para cambiar algunas características de la graficación. En la tabla 2.5 se visualizan las funciones utilizadas en la simulación.

**Tabla 2.5.** Funciones para la graficación.

Variable	Definición
<i>hold on</i>	Permite graficar otra curva en la misma figura usada anteriormente por otra curva, sin afectar los valores de los ejes existentes de la primer curva.

<i>grid on</i>	Muestra líneas cuadriculadas en la gráfica.
<i>axis</i>	Permite especificar los límites para los ejes actuales. Se especifica los límites tanto del eje x, como del eje y.

Para realizar el gráfico de BER vs Eb/No se utiliza la función *semilogy* que gráfica con una escala lineal para el eje x y con una escala logarítmica en base 10 para el eje y. Además, se coloca el título respectivo a la gráfica, los nombres de los ejes y los nombres para las legendas de las curvas generadas. En el segmento de código 2.23 se visualiza la gráfica de resultados [13].

```

%Nombres para Legendas
leg1=['Est. Perfecta alfa=' num2str(alfa(1),2)];
leg2=['Est. Perfecta alfa=' num2str(alfa(2),2)];
leg3=['Est. LS alfa=' num2str(alfa(1),2)];
leg4=['Est. LS alfa=' num2str(alfa(2),2)];
%Gráfica BER vs Eb/No en SEFDM
%Para diferentes Factores de Compresión de AB
figure;
semilogy(EbNo,BER(1,:), 'b--o', 'LineWidth',1.2);
hold on; grid on;
semilogy(EbNo,BER(2,:), 'm--o', 'LineWidth',1.2);
semilogy(EbNo,BER(3,:), 'k--s', 'LineWidth',1.2);
semilogy(EbNo,BER(4,:), 'g--s', 'LineWidth',1.2);
axis([0 30 10^-4 1]);
title({'DETECTOR MMSE'; 'BER vs Eb/No en SEFDM para distintos alfa';...
      'Estimacion de Canal'});
legend(leg1,leg2,leg3,leg4)
xlabel('Eb/No [dB]', 'FontSize',11);
ylabel('BER', 'FontSize',11);

```

**Segmento de código 2.23.** Gráfica de resultados.

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan los resultados obtenidos para el sistema SEFDM mediante el software de MATLAB. Se presentan las curvas de BER vs Eb/No para diferentes escenarios de simulación. Se utilizó como parámetros configurables el tipo de modulación y el factor de compresión. Además, se realiza la comparación entre la estimación de canal perfecta y el método de estimación LS (SEFDM-OFDM).

#### 3.1 ESCENARIO DE SIMULACIÓN

Las simulaciones se realizaron en MATLAB en la versión 2022-A. Para la simulación se implementó la etapa de transmisión SEFDM, canal inalámbrico y la etapa de recepción SEFDM. Para el desarrollo de todas las gráficas se definen algunos parámetros fijos que se pueden visualizar en la tabla 3.1.

**Tabla 3.1.** Parámetros fijos de simulación.

Etapa	Parámetro	Valor
Transmisión	Número de subportadoras	16
	Tamaño de prefijo cíclico	4
	Número de símbolos por paquete	5
Codificación	traceback	32
	Tasa de codificación	1/2
Canal Inalámbrico	Parámetro del canal de tipo Rician (K)	10
	Frecuencia de muestreo para el canal Rician	20e6
	Retardo Relativo	50 ns

Además, el canal inalámbrico usa ruido AWGN y desvanecimiento multitrayectoria de tipo Rician. En la tabla 3.2 se pueden visualizar los requerimientos de PDP (*Power Delay Profile*) del canal Rician, como retardos de ruta y ganancias promedio. Estos parámetros son usados en todos los escenarios de simulación.

**Tabla 3.2.** Parámetros PDP para canal Rician.

Retardos de Rutas [ns]	Retardos Normalizados	Ganancias Promedio de ruta [dB]
0	0	0
100	2	-9.7

En el programa principal se definen los valores de los parámetros variables. Para el nivel de modulación se define la variable  $m = [2,4,6]$  siendo 2 para QPSK, 4 para 16QAM y 6 para 64QAM. El factor de compresión se define por un vector  $a$ , dicho vector tiene valores de 0 a 6, este valor numérico representa el número de ceros que se añaden a la fórmula  $alfa = \frac{N}{N+a}$  siendo  $N = 16$ , el  $alfa_{m\acute{a}x} = 1$  y el  $alfa_{min} = 0,73$ .

Para el bloque de estimación de canal se define por un valor de [1,2] siendo 1 para realizar una estimación perfecta y 2 para realizar una estimación LS (método SEFDM-OFDM). Finalmente, para el detector MMSE se define una matriz de identidad de tamaño  $N$  y la matriz canal  $H$ , proveniente del bloque de estimación.

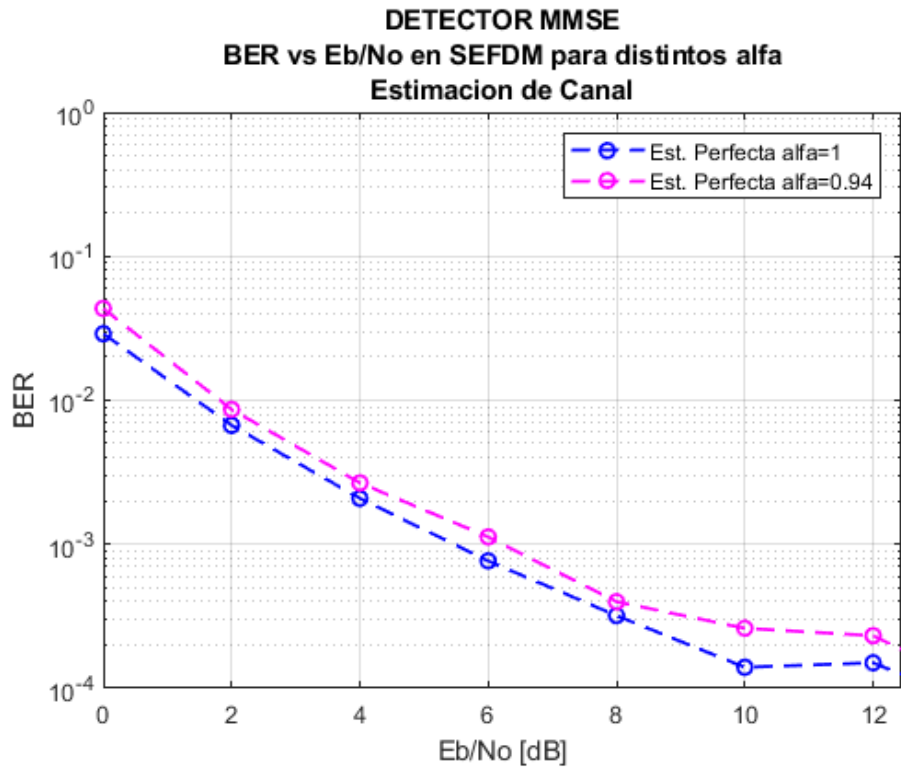
## 3.2 ANÁLISIS DE RESULTADOS

Se verifica que el sistema SEFDM implementado funcione correctamente. Además, se realiza la comparación entre las curvas de BER vs  $E_b/N_0$  tanto para la estimación de canal perfecta, como para la estimación de canal LS. Finalmente, se varían los parámetros configurables para ver cómo afectan a las curvas de BER.

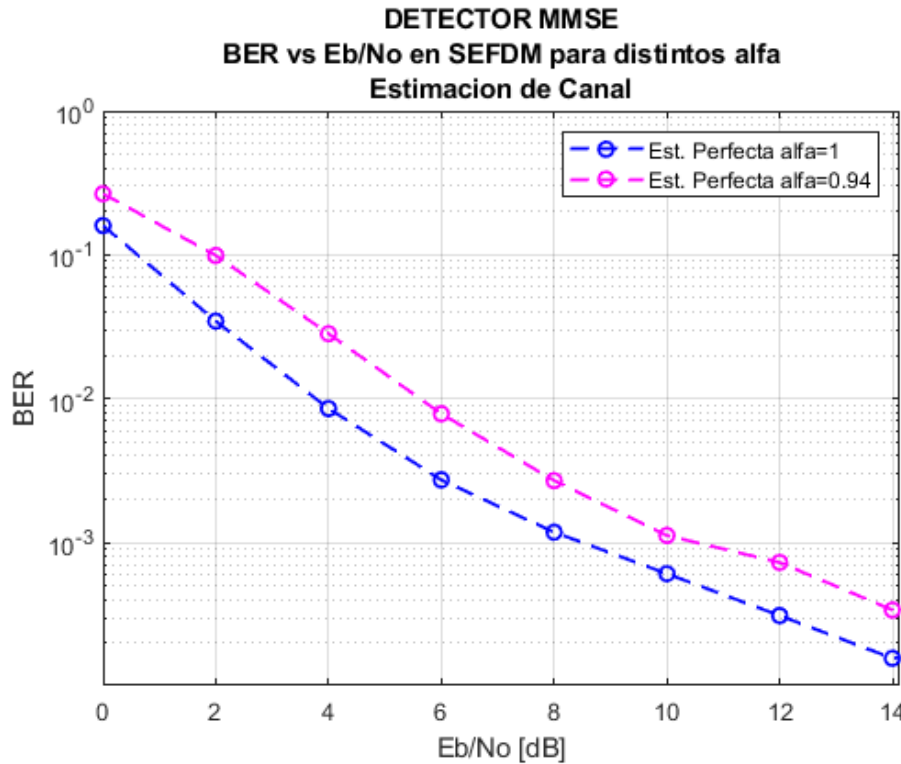
### 3.2.1 COMPARACIÓN ENTRE OFDM Y SEFDM

Para verificar el funcionamiento del sistema SEFDM se lo compara con un sistema OFDM. Al perder la ortogonalidad de las subportadoras, los valores de BER para SEFDM deberán ser peores que para OFDM. Siguiendo la definición de subportadoras para SEFDM de  $\frac{N}{\alpha}$ , si  $\alpha = 1$ , el número de subportadoras de SEFDM es  $N$ . Por tanto, para un  $\alpha = 1$  se tiene un sistema OFDM. En las figuras 3.1 a 3.3 se visualizan las curvas de BER para un sistema OFDM (SEFDM  $\alpha = 1$ ) y SEFDM con un  $\alpha = 0.94$ , con los diferentes esquemas de modulación (QPSK, 16QAM y 64QAM). Para las siguientes gráficas se utilizó una técnica de estimación perfecta y un  $N_{trials} = 10000$ .

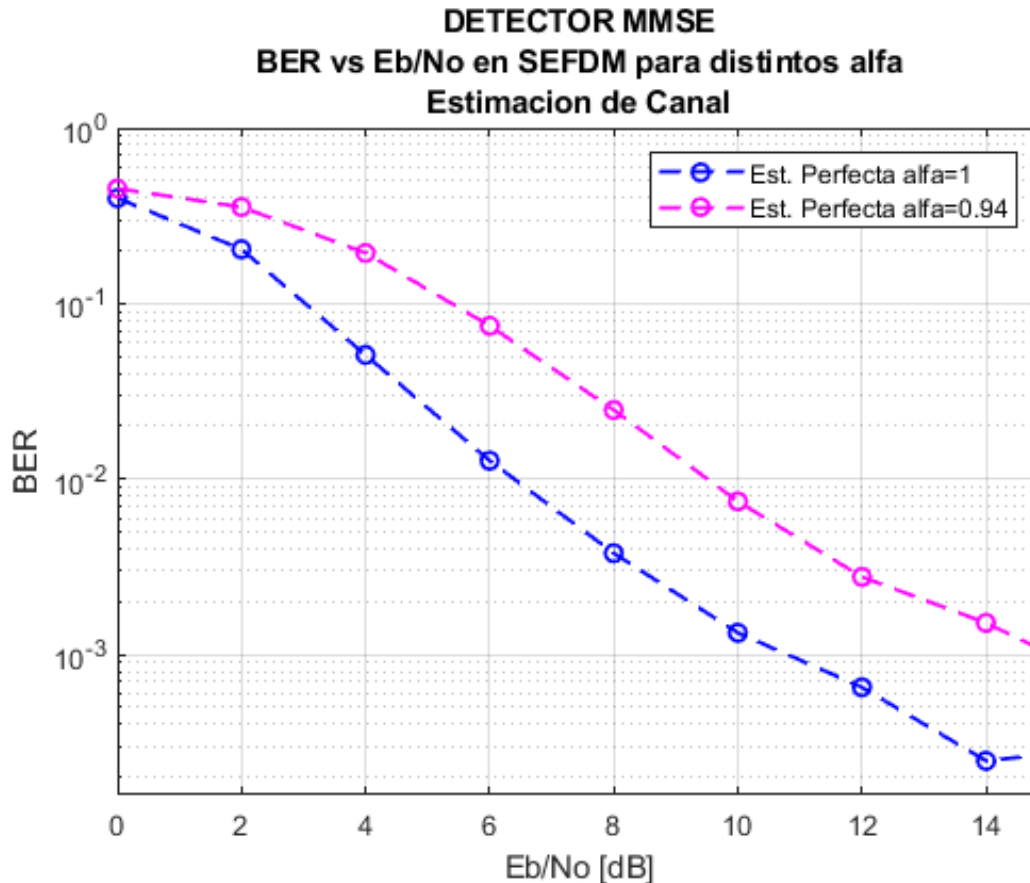




**Figura 3.1.** BER vs Eb/No de sistema OFDM (SEFDM  $\alpha = 1$ ) y sistema SEFDM  $\alpha = 0.94$  para QPSK.



**Figura 3.2.** BER vs Eb/No de sistema OFDM (SEFDM  $\alpha = 1$ ) y sistema SEFDM  $\alpha = 0.94$  para 16QAM.



**Figura 3.3.** BER vs Eb/No de sistema OFDM (SEFDM  $\alpha = 1$ ) y sistema SEFDM  $\alpha = 0.94$  para 64QAM.

Para las curvas de BER para SEFDM con  $\alpha = 0.94$  se puede apreciar cómo los valores son más degradados con respecto a las curvas para OFDM, especialmente cuando incrementa el nivel de modulación. Por lo cual, el sistema SEFDM al perder la ortogonalidad degrada el valor del BER.

Como se puede apreciar en las figuras 3.1, 3.2 y 3.3, para determinados valores de BER el Eb/No es diferente entre OFDM y SEFDM para los tres tipos de modulación usados. Se encuentra una menor diferencia entre los sistemas cuando se usa una modulación de QPSK, por lo que, las curvas de BER son muy parecidas. Caso contrario con la modulación 64 QAM; esto debido a que la distancia entre los puntos del diagrama de constelación es menor, por lo que, al recuperar la señal de datos de SEFDM tendrá más errores.

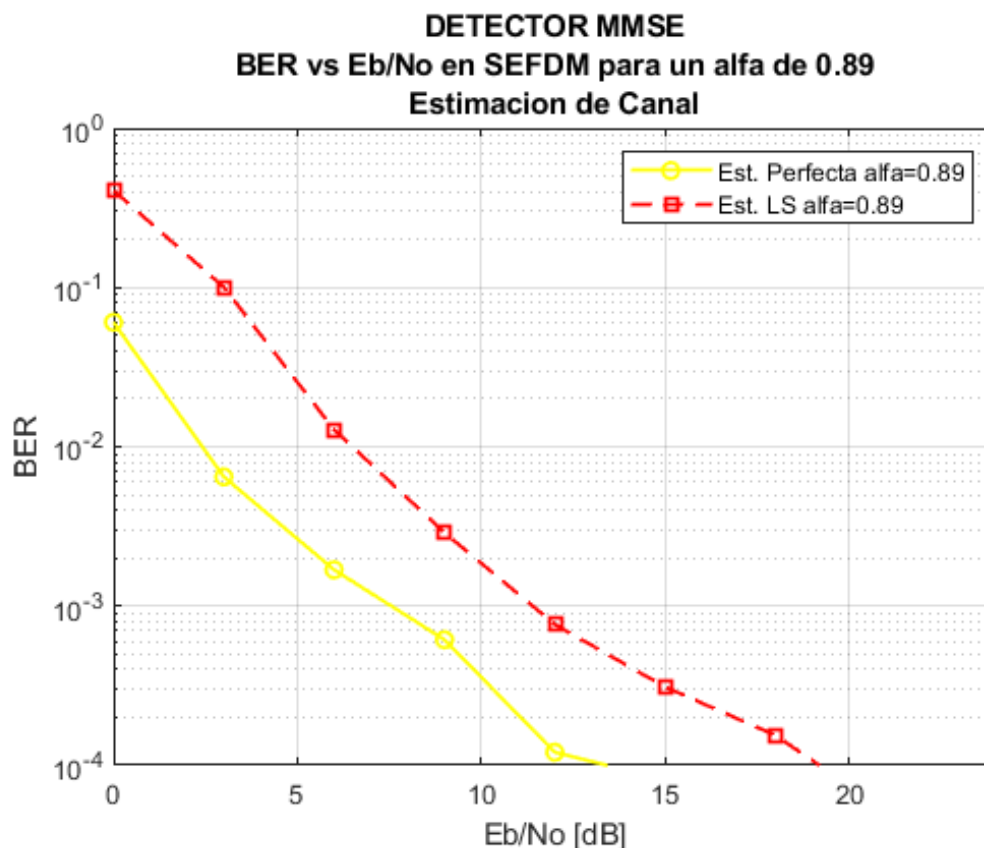
Mediante las figuras 3.1 a 3.3 se aprecian las diferencias en el Eb/No para cierto BER. Por ejemplo, en la modulación QPSK se puede apreciar que para un BER de  $10^{-2}$  el SEFDM requiere un Eb/No adicional de 0.2 dB para alcanzar el mismo BER que OFDM. Mientras

que con 16QAM esa diferencia aumenta a alrededor de 1.8 dB y para 64QAM crece al valor de 3 dB.

A medida que el valor de  $E_b/N_0$  aumenta, las diferencias entre OFDM y SEFDM se vuelven más grandes. Por ejemplo, se puede apreciar que para un BER de  $10^{-3}$  en SEFDM requiere un  $E_b/N_0$  de 5.5 dB y para OFDM de 6 dB. La diferencia aumentó de 0.2 dB para  $10^{-2}$  a 0.5 dB para  $10^{-3}$ . Para 16QAM la diferencia de 1.8 dB pasó a 2 dB y para 64QAM crece de 3 dB a 4.5 dB.

### 3.2.2 COMPARACIÓN ENTRE ESTIMACIÓN PERFECTA Y ESTIMACIÓN LS

Para comprobar el correcto funcionamiento del método de estimación de canal implementado, se realiza la comparación de las curvas de BER vs  $E_b/N_0$ , tanto, de la estimación de canal perfecta, como de la estimación de canal LS. En la figura 3.4 se tiene la comparación entre ambas técnicas de estimación. Dentro de los parámetros de simulación se utilizó un  $m = 2$  es decir QPSK, un  $N_{trials} = 10000$  y un  $\alpha = 0.89$ .



**Figura 3.4.** BER vs  $E_b/N_0$  de estimación perfecta y estimación LS para QPSK.

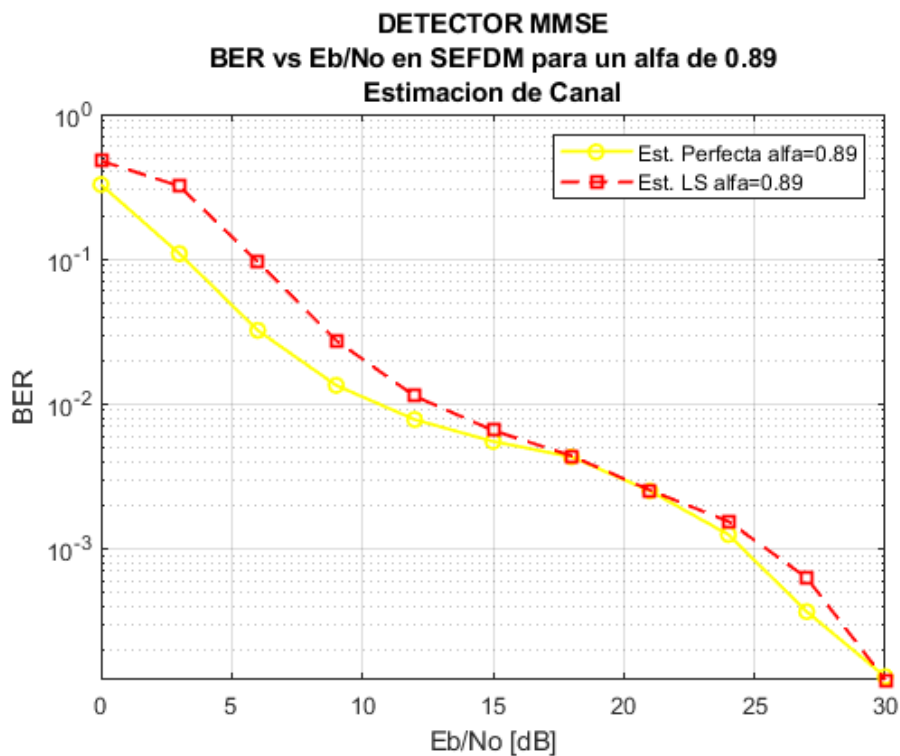
En la figura 3.4 se puede apreciar la comparación de los resultados obtenidos para los valores de BER vs  $E_b/N_0$ . Estos valores permiten medir que tan cercana es la curva del

método de estimación propuesto para la curva de estimación perfecta. Se mide el rendimiento de las técnicas de estimación. Si bien, para la estimación LS (método SEFDM-OFDM) no es exactamente igual a la estimación perfecta, posee un buen rendimiento.

Ambas técnicas son muy similares en cuanto a la caída de sus pendientes. A medida que el valor de BER disminuye, la diferencia entre  $E_b/N_0$  de las técnicas de estimación aumenta. Por lo tanto, para un valor de BER de  $10^{-2}$  el valor de  $E_b/N_0$  para estimación perfecta es de 3 dB y para la estimación LS es de 6.1 dB, por lo que, existe una diferencia de 3.1 dB entre ambas técnicas. Para un valor de BER de  $10^{-3}$  el valor de  $E_b/N_0$  para estimación perfecta es de 7.5 dB y para la estimación LS es de 11.5 dB, la diferencia aumenta a 4 dB.

La diferencia entre las curvas de BER para las técnicas de estimación son más visibles cuando se usa una modulación QPSK. Además, dicha diferencia es más notoria con valores altos de  $E_b/N_0$ . Por tanto, la técnica de estimación LS para QPSK no logra tener el mismo rendimiento de la técnica de estimación perfecta.

De la misma manera se realiza la comparación entre la estimación de canal perfecta y la estimación de canal LS para un  $m = 4$  es decir, para 16QAM. En la figura 3.5 se visualiza la comparación entre las curvas de estimación de canal. Los parámetros se mantienen iguales que para QPSK con  $N_{trials} = 10000$  y un  $\alpha = 0.89$ .

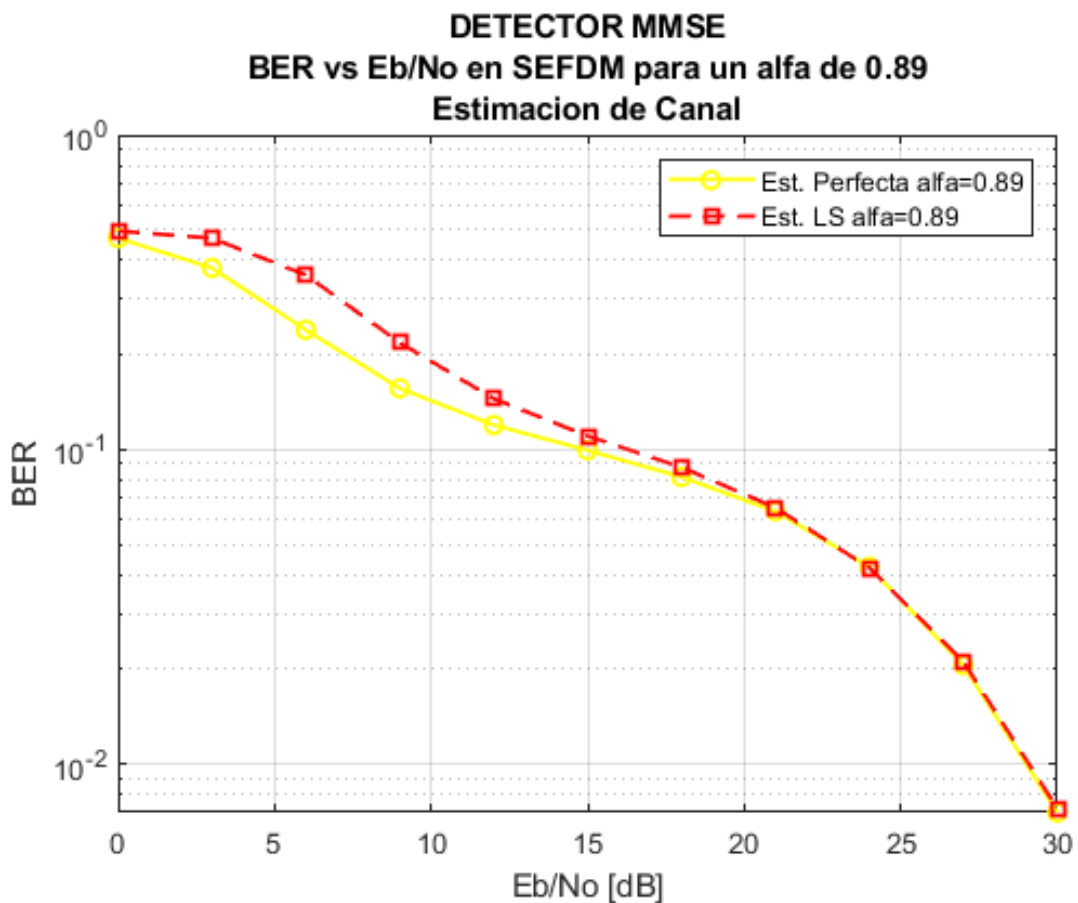


**Figura 3.5.** BER vs  $E_b/N_0$  de estimación perfecta y estimación LS para 16QAM.

En la figura 3.5 las curvas para 16QAM son mucho más similares que para QPSK. En este caso, la diferencia entre las curvas es mucho menor y son prácticamente idénticas para un  $E_b/N_0$  de 18 a 22 dB. Se puede apreciar los resultados obtenidos de BER. Estos valores permiten medir que tan cercana es la curva del método de estimación propuesto con respecto a la curva de estimación perfecta.

El rendimiento de ambas técnicas es muy similar para 16 QAM. Sin embargo, para el valor de BER como  $10^{-1}$  se aprecia una diferencia entre ambas curvas; para la estimación perfecta se tiene un  $E_b/N_0$  de 3 dB y para la estimación LS un  $E_b/N_0$  de 6 dB. Cuando el  $E_b/N_0$  pasa de 18 dB, los valores en ambas gráficas son prácticamente similares, como se puede apreciar cuando se tiene un valor de BER de  $10^{-4}$ .

Finalmente se realiza la comparación de las curvas de estimación de canal perfecta y la estimación de canal LS para un  $m = 6$  es decir, para 64QAM. En la figura 3.6 se puede observar las curvas de BER de las técnicas de estimación con los siguientes parámetros:  $N_{trials} = 10000$  y  $\alpha = 0.89$ .



**Figura 3.6.** BER vs  $E_b/N_0$  de estimación perfecta y estimación LS para 64QAM.

En la figura 3.6 ambas gráficas visualmente son iguales cuando el valor del  $E_b/N_0$  supera los 20 dB. A medida que aumentó el nivel de modulación ambas técnicas de estimación se vuelven muy idénticas en cuanto al rendimiento del sistema SEFDM. Se puede apreciar que los valores de BER para ambas técnicas de estimación son prácticamente los mismos. Por ejemplo, para un valor de BER de  $10^{-1}$  se tiene un  $E_b/N_0$  de 15 dB para OFDM y para SEFDM un valor de  $E_b/N_0$  de 15.5 dB. A partir del valor de  $E_b/N_0$  de 20 dB los valores de BER son los mismos, como se aprecia para un BER de  $10^{-2}$  donde el  $E_b/N_0$  es 29 dB para ambas técnicas de estimación de canal.

### 3.2.3 ANÁLISIS DEL FACTOR DE COMPRESIÓN

Al manejar parámetros configurables dentro de la simulación, las curvas de BER generadas se verán afectadas, por lo cual, se debe realizar un análisis de como el factor de compresión afecta a las curvas del BER. Por tanto, se realiza simulaciones para la técnica de estimación LS con diferentes valores de alfa. Los parámetros de simulación cambian usando un  $N_{trials} = 8000$ . En las figuras 3.7 y 3.8 se tiene la comparación entre curvas de BER para diferentes alfas usando los esquemas de modulación de QPSK y 64QAM.

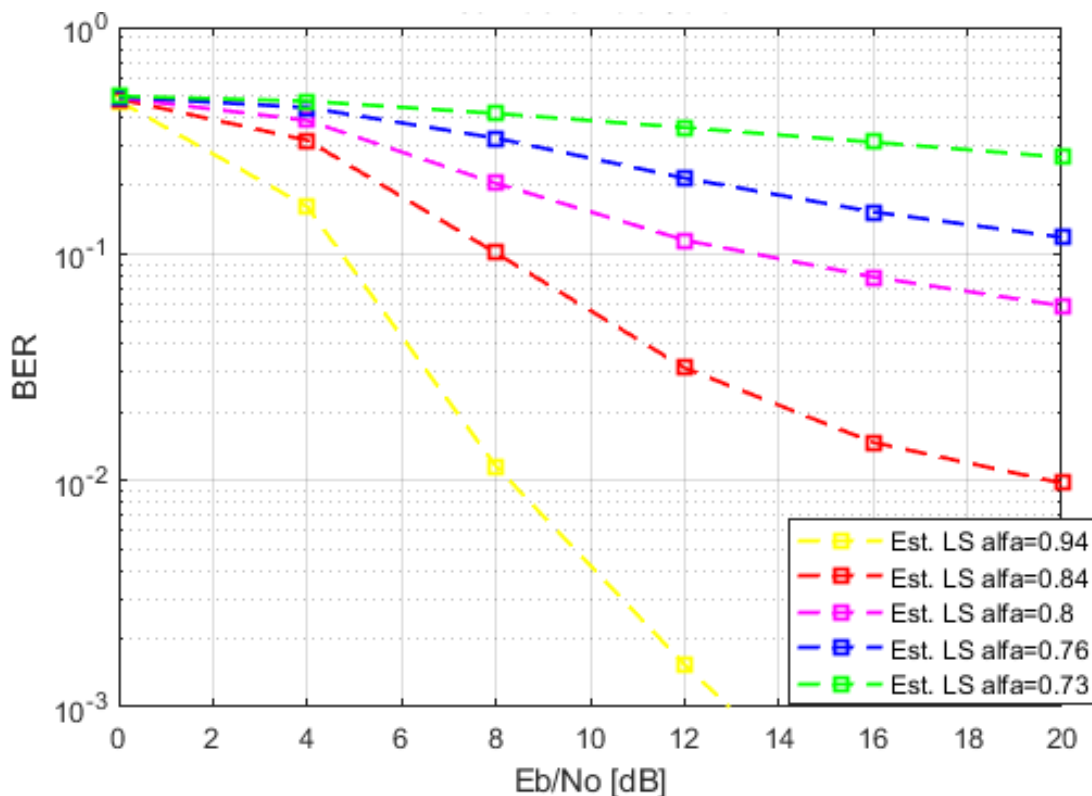
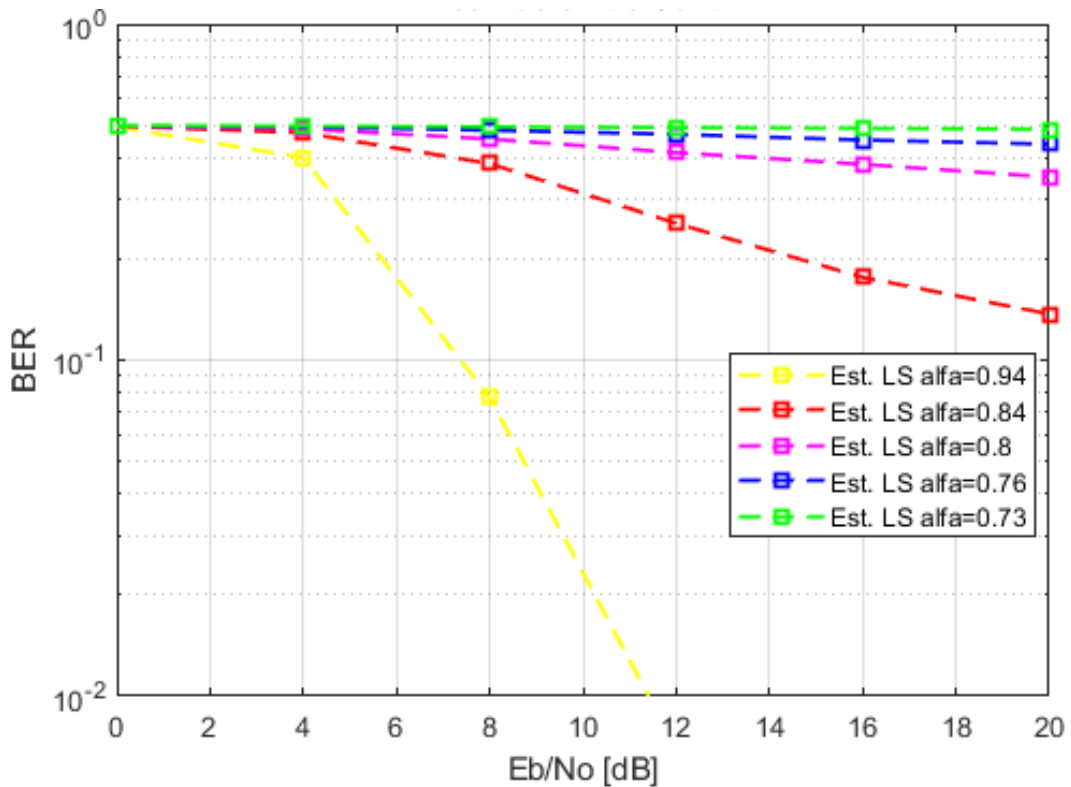


Figura 3.7. BER vs  $E_b/N_0$  estimación LS para diferentes alfas (QPSK).



**Figura 3.8.** BER vs  $E_b/N_0$  estimación LS para diferentes  $\alpha$ s (64QAM).

En la gráfica 3.7 se visualizan cinco curvas, cada una con un  $\alpha$  diferente, con valores de  $[0.94, 0.84, 0.8, 0.76, 0.73]$  para una modulación QPSK. A medida que el valor de  $\alpha$  tiene a acercarse a 1, es decir, se tiene menos ceros, la curva tiende a mejorar su rendimiento para los valores de BER; esto debido a que existe una menor compresión de ancho de banda, por lo cual, es más fácil compensar los efectos del canal al realizar un proceso de estimación de canal LS y detección MMSE. De esta manera, se puede comprobar cuál será el mejor  $\alpha$  para la transmisión sin que el valor de BER sea degradado en exceso.

En la figura 3.8 se visualiza las curvas de BER para los mismos valores de  $\alpha$ , pero usando una modulación 64QAM. A diferencia de las curvas de BER para QPSK donde tendían a mejorar cuando el valor de  $E_b/N_0$  aumentaba, para 64 QAM si el valor del  $\alpha$  es muy bajo, las curvas se mantienen como una línea prácticamente recta, por lo que demuestran un mal rendimiento cuando los valores de  $\alpha$  son muy bajos.

### 3.3 CONCLUSIONES

- Al usar un sistema SEFDM se logra reducir el ancho de banda utilizado para la transmisión de datos, por lo que, con un menor ancho de banda se transmiten el mismo número de datos que con un sistema OFDM. De esta manera, SEFDM alcanza una mejor eficiencia espectral que OFDM.
- SEFDM usa un factor de compresión para comprimir las subportadoras, al comprimir las subportadoras pierden ortogonalidad, por lo que, se produce una degradación de BER; dicho factor es representado por  $\alpha = \frac{N}{N+a}$ ,  $a$  es el número de ceros que serán añadidos para la definición de SEFDM. Si existe una mayor cantidad de ceros,  $\alpha$  decrecerá en magnitud lo que provoca una compresión mucho mayor en las subportadoras, pero también una degradación más grande de los valores de BER.
- Existen muchas técnicas de estimación de canal tanto para el dominio del tiempo y de la frecuencia, varían dependiendo de la complejidad o los algoritmos utilizados. La técnica de estimación LS con el método SEFDM-OFDM demostró tener un buen rendimiento; en muchas simulaciones llegó a ser igual a la técnica de estimación perfecta, especialmente para 16QAM y 64QAM. Sin embargo, para valores pequeños de  $E_b/N_0$  se observó que existen diferencias entre las técnicas utilizadas.
- Dependiendo del tipo de modulación usado se encuentra un mejor rendimiento en las curvas de BER; el mejor esquema de modulación es QPSK, esto debido, a que existe una mayor distancia entre los puntos del diagrama de constelación, por lo que, el ruido AWGN del canal inalámbrico afecta menos a la señal de datos que se recupera en el receptor.
- Al usar un canal inalámbrico de tipo Rician, se debe tener en cuenta el desvanecimiento selectivo en frecuencia provocado, el mismo es corregido por el detector MMSE, a pesar de la simpleza de dicho detector, demuestra tener un buen rendimiento para el canal con desvanecimiento de tipo Rician.
- Para los bloques  $\frac{N}{\alpha}$ IFFT y  $\frac{N}{\alpha}$ FFT mediante la variable *ceros* se añadieron los ceros correspondientes con  $a$  lo que permite simular el sistema SEFDM en el software de Matlab, por lo cual, el tamaño de los vectores para los bloques  $\frac{N}{\alpha}$ IFFT y  $\frac{N}{\alpha}$ FFT dependen de  $\frac{N}{\alpha}$ .



- Mediante el software de Matlab es posible realizar la simulación de un sistema SEFDM, ya que el programa tiene librerías como Communications Toolbox, que permite la simulación de ciertos parámetros de un sistema inalámbrico.

### **3.4 RECOMENDACIONES**

- Es recomendable utilizar una computadora con buenas características computacionales, debido a los requerimientos de hardware y software para la simulación.
- Para que se generen las curvas de BER es recomendable realizar una gran número de iteraciones, al menos de 8000 iteraciones; teniendo en cuenta que la simulación demorará más tiempo al aumentar el número de iteraciones.
- Para tener una mejor visualización de las gráficas de las curvas de resultados es recomendable modificar los valores de los ejes.
- Se recomienda realizar un trabajo de integración curricular basado en la comparación de diferentes técnicas de estimación de canal para diferentes tipos de canal inalámbrico como AWGN, Rayleigh y Rician.

## 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] I. Darwazeh, . H. Ghannam y T. Xu, "The First 15 Years of SEFDM: A Brief Survey", *2018 11th International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP)*, pp. 1-7, 2018.
- [2] I. Darwazeh y M. R. Rodriguez, "Fast OFDM: A Proposal for Doubling the Data Rate of OFDM Schemes", *IEEE/IEE International Conference on Telecommunications (ICT 2002)*, vol. III, pp. 484 - 487, 2002.
- [3] I. Darwazeh y . M. R. D. Rodrigues, "A Spectrally Efficient Frequency Division", *8th International OFDM-Workshop, Proceedings, Hamburg*, p. 70 74, 2003.
- [4] T. Xu y I. Darwazeh, "Spectrally efficient FDM: Spectrum saving technique for 5G?", *1st International Conference on 5G for Ubiquitous Connectivity*, pp. 273-278, 2014.
- [5] S. Isam y I. Darwazeh, "Robust channel estimation for Spectrally Efficient FDM system", *2012 19th International Conference on Telecommunications (ICT)*, pp. 1-6, 2012.
- [6] I. Kanaras, A. Chorti, M. R. Rodrigues y I. Darwazeh, "Spectrally Efficient FDM Signals: Bandwidth Gain at the Expense of Receiver Complexity", *in Communications, 2009. ICC '09. IEEE International Conference on*, pp. 1-6, 2009.
- [7] I. Darwazeh y T. Xu, "Transmission Experiment of Bandwidth Compressed Carrier Aggregation in a Realistic Fading Channel", *IEEE Transactions on Vehicular Technology*, vol. 66, p. 4087–4097, 2017.
- [8] S. Pupiales, "ANÁLISIS DE GANANCIA DE ANCHO DE BANDA Y DEGRADACIÓN DE LA BER EN UN SISTEMA SEFDM CON RESPECTO A OFDM USANDO MATLAB", *ESCUELA POLITÉCNICA NACIONAL, Quito*, pp. 1-98, 2020.
- [9] H. Ghannam y I. Darwazeh, "Signal coding and interference cancellation of spectrally efficient FDM systems for 5G cellular networks", *2017 24th International Conference on Telecommunications (ICT)*, pp. 1-6, 2017.
- [10] H. Ghannam y I. Darwazeh, "SEFDM: Spectral Efficiency Upper Bound and Interference Distribution.", *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP).*, pp. 1-6, 2018.
- [11] E. Espinoza, "SIMULACIÓN DE LOS MÉTODOS LS Y MMSE PARA ESTIMACIÓN DE CANAL EN SISTEMAS OFDM", *ESCUELA POLITÉCNICA NACIONAL, Quito*, pp. 1-99, 2019.
- [12] H. Ghannam y I. Darwazeh, "Robust Channel Estimation Methods for Spectrally Efficient FDM Systems", *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pp. 1-6, 2018.

[13] G. A. Logacho Quishpe, "SIMULACIÓN EN MATLAB DE LOS DETECTORES ZF, MMSE E ID PARA UN SISTEMA SEFDM UTILIZANDO UN CANAL CON DESVANECIMIENTO", *ESCUELA POLITÉCNICA NACIONAL*, pp. 1-96, 2021.

[14] A. F. Conterón Picuas, "SIMULACIÓN DE ESTIMACIÓN DE CANAL EN LA TÉCNICA NOMA", *ESCUELA POLITÉCNICA NACIONAL*, pp. 1-54, 2022.

## **5 ANEXOS**

ANEXO I. Programa principal y funciones utilizadas para la simulación del sistema SEFDM con estimación de canal, son presentadas en el Anexo Digital.