

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**ANÁLISIS DE GANANCIA DE ANCHO DE BANDA Y
DEGRADACIÓN DE LA BER EN UN SISTEMA SEFDM CON
RESPECTO A OFDM USANDO MATLAB**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES**

SANDRA MARISOL PUPIALES ANGAMARCA

sandra.pupiales@epn.edu.ec

DIRECTOR: ING. MARTHA CECILIA PAREDES PAREDES, Ph.D.

cecilia.paredes@epn.edu.ec

Quito, febrero 2020

AVAL

Certifico que el presente trabajo fue desarrollado por Sandra Marisol Pupiales Angamarca, bajo mi supervisión.

Ph.D. Martha Cecilia Paredes Paredes
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Sandra Marisol Pupiales Angamarca, declaro bajo juramento que el trabajo 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.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Sandra Marisol Pupiales Angamarca

DEDICATORIA

Con mucho amor a Dios, mis padres y hermanos.

AGRADECIMIENTO

A Dios y la Virgencita, quienes me acompañan en todos los momentos de mi vida. Por darme fortaleza y sabiduría.

A mis padres Luis Alberto y Silvia Esperanza por su apoyo incondicional, por su esfuerzo, ejemplo, amor y comprensión.

A mis hermanos Silvia y Pablo por su apoyo y cariño.

A mi compañero de vida Danilo por su amor sincero.

Un profundo agradecimiento a mi directora, Ph.D. Cecilia Paredes, por su tiempo, confianza, apoyo, paciencia y conocimientos oportunos dedicados durante la realización de este trabajo de titulación, gracias por su amistad.

A los docentes de la EPN de manera especial al Ph.D. Felipe Grijalva y M.Sc. Jorge Carvajal por compartir su tiempo y conocimientos conmigo durante esta etapa.

A mis amigos por compartir tantas alegrías y por su apoyo de una u otra manera durante la universidad.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
RESUMEN	VII
ABSTRACT	VIII
1. INTRODUCCIÓN	1
1.1. OBJETIVOS.....	2
1.2. ALCANCE	2
1.3. MARCO TEÓRICO	3
1.3.1. SISTEMA OFDM	3
1.3.1.1. Transmisor y receptor de un sistema OFDM.....	5
1.3.1.2. Ventajas e inconvenientes de OFDM	10
1.3.2. SISTEMA SEFDM.....	10
1.3.2.1. Transmisor y receptor de un sistema SEFDM	13
1.3.2.2. Ventajas e inconvenientes de SEFDM	22
1.3.3. PARÁMETROS INVOLUCRADOS PARA LA MEDICIÓN DEL REDIMIENTO DE OFDM Y SEFDM	22
2. METODOLOGÍA	25
2.1. SISTEMA OFDM.....	25
2.1.1. TRANSMISOR OFDM.....	26
2.1.2. CANAL AWGN	28
2.1.3. RECEPTOR OFDM.....	28
2.1.4. RESULTADOS EN OFDM	30
2.1.4.1. Cálculo de errores y BER en OFDM.....	30
2.1.4.2. Determinación de la PSD en OFDM.....	32
2.2. SISTEMA SEFDM.....	33
2.2.1. TRANSMISOR SEFDM.....	34
2.2.2. RECEPTOR SEFDM.....	36
2.2.3. RESULTADOS EN SEFDM	42
2.2.3.1. Cálculo de errores y BER en SEFDM.....	42
2.2.3.2. Determinación de la PSD en SEFDM.....	44
3. RESULTADOS Y DISCUSIÓN	48
3.1. ESCENARIO DE SIMULACIÓN	48

3.1.1.	PRUEBAS CON OFDM	48
3.1.2.	PRUEBAS CON SEFDM	50
3.1.3.	SELECCIÓN DEL NÚMERO DE ITERACIONES v DEL DETECTOR....	52
3.2.	RESULTADOS DE BER vs E_b/N_0	54
3.2.1.	MODULACIÓN 4-QAM	54
3.2.2.	MODULACIÓN 16-QAM	57
3.2.3.	MODULACIÓN 64-QAM	59
3.2.4.	SIMULACIÓN CON Y SIN DETECTOR	61
3.2.5.	DEGRADACIÓN DE LA BER vs ALFA α	62
3.3.	RESULTADOS DE PSD	63
3.3.1.	RESULTADOS CON OFDM	63
3.3.2.	RESULTADOS CON SEFDM	64
4.	CONCLUSIONES Y RECOMENDACIONES	69
4.1.	CONCLUSIONES	69
4.2.	RECOMENDACIONES.....	70
5.	REFERENCIAS BIBLIOGRÁFICAS	72
	ANEXOS.....	76
	ANEXO A: RESULTADOS DE BER vs E_b/N_0 EN OFDM	76
	ANEXO B: RESULTADOS DE PSD EN OFDM	78
	ANEXO C: FUNCIÓN DETECTOR ID.....	79
	ANEXO D: RESULTADOS DE BER vs E_b/N_0 EN SEFDM.....	82
	ANEXO E: FUNCIÓN SEFDM.....	84
	ANEXO F: RESULTADOS DE PSD EN SEFDM	85
	ANEXO G: FUNCIÓN SEFDM_tx.....	87

RESUMEN

Las técnicas de transmisión multiportadora consisten en dividir el ancho de banda en varios canales, llamados subportadoras. Estas técnicas son muy utilizadas en varios estándares de comunicaciones. SEFDM (*Spectrally Efficient Frequency Division Multiplexing*) es una técnica de transmisión multiportadora no-ortogonal que proporciona una reducción de la utilización del espectro comparado con OFDM (*Orthogonal Frequency Division Multiplexing*). Esta mejora en la eficiencia espectral se consigue a costa de degradar la BER (*Bit Error Rate*) debido a la pérdida de la ortogonalidad de las subportadoras.

El presente proyecto técnico tiene como objetivo comparar la técnica de transmisión SEFDM frente a OFDM en términos de espectro y degradación de la BER, para lo cual se realiza la implementación de estos sistemas en MATLAB a fin de examinar las mejoras e inconvenientes de esta nueva técnica con relación a OFDM a través de simulaciones.

El desarrollo del proyecto técnico se divide en cuatro capítulos. En el primer capítulo se revisan los conceptos teóricos relacionados con OFDM y SEFDM, dentro de SEFDM se mencionan varios detectores. Un detector permitirá compensar la degradación de la BER al perder la ortogonalidad de las subportadoras. Luego, en el capítulo 2 se detalla la implementación en MATLAB del transmisor y receptor de los dos sistemas, en el receptor de SEFDM se implementa el detector ID (*Iterative Detection*), el cual ayuda a mejorar la BER. Posteriormente, en el capítulo 3 se muestran los resultados de las simulaciones en términos de BER vs E_b/N_0 (*Energy per-bit to noise power spectral density ratio*) y de PSD (*Power Spectral Density*). Finalmente, en el capítulo 4 se presentan las conclusiones y recomendaciones del desarrollo del proyecto técnico.

PALABRAS CLAVE: OFDM, SEFDM, ID, BER, PSD.

ABSTRACT

The multicarrier transmission techniques consist of dividing the available bandwidth into several channels, called subcarriers. These techniques are used widely in several communications standards. SEFDM (*Spectrally Efficient Frequency Division Multiplexing*) is a non-orthogonal multicarrier transmission technique that provides a reduction in spectrum utilization compared to OFDM (*Orthogonal Frequency Division Multiplexing*). This improvement in spectral efficiency is achieved at the expense of BER (*Bit Error Rate*) degradation due to the loss of orthogonality of subcarriers.

This technical project aims to compare the SEFDM transmission technique against OFDM in terms of spectrum gain and the BER degradation, for which the implementation of these systems is carried out in MATLAB to examine through simulations the improvements and disadvantages of this new technique in relation with OFDM.

This technical project is organized into four chapters. The first chapter reviews the theoretical concepts related to OFDM and SEFDM, within SEFDM mentions several detectors. A detector allows to compensate BER degradation due to the loss of subcarriers orthogonality. Then, chapter 2 details the implementation in MATLAB of the transmitter and receiver of the two systems also, in the SEFDM receiver the ID (Iterative detection) detector is implemented, which helps to minimize the BER. Subsequently, chapter 3 shows the results of the simulations in terms of BER vs E_b/N_0 (*Energy per-bit to noise power spectral density ratio*) and PSD (*Power Spectral Density*). Finally, chapter 4 presents the conclusions and recommendations for the development of this technical project.

KEYWORDS: OFDM, SEFDM, ID, BER, PSD.

1. INTRODUCCIÓN

Actualmente las comunicaciones inalámbricas son parte fundamental de la vida cotidiana de una sociedad. La mayoría de los estándares para estas comunicaciones hacen uso de OFDM (*Orthogonal Frequency Division Multiplexing*), la cual es una técnica de acceso o técnica de transmisión multiportadora, ortogonal, que se encuentra dentro de la capa física. La utilización de esta técnica es debido a su alta eficiencia espectral y por ende elevadas tasas de transmisión y su robustez contra el desvanecimiento de frecuencia selectiva [1].

Sin embargo, con el incremento de la demanda de servicios inalámbricos rápidos, y al mismo tiempo el aumento de dispositivos conectados, los cuales son mayor al número de usuarios conectados, conduce a la generación de nuevos sistemas de comunicaciones inalámbricos que buscan aumentar aún más la eficiencia espectral, teniendo en cuenta que el espectro es un recurso cada vez más limitado y costoso [2]. Por lo tanto, se deben analizar nuevas técnicas de acceso que solventen estas nuevas necesidades.

Por ejemplo, en las comunicaciones celulares está por surgir la quinta generación (5G). El desarrollo de 5G pretende tener mejoras en tasas de datos, alta eficiencia espectral, mayor cantidad de dispositivos conectados, ahorro de energía, reducción de latencia, etc. Para obtener dichas mejoras se debe tener una gran disponibilidad del espectro, la generación de la señal debe ser flexible para combinar distintas tasas de bits dependiendo de la aplicación, modificar la estructura de las redes reduciendo el tamaño de las celdas para soportar mayor capacidad en tráfico de datos y número de usuarios [3].

En esta expectativa surge SEFDM (*Spectrally Efficient Frequency Division Multiplexing*), que es una técnica de acceso multiportadora no-ortogonal que busca la posibilidad de mejorar la eficiencia espectral haciendo uso de la no-ortogonalidad, que reduce el espacio entre las subportadoras de OFDM dependiendo del factor de compresión que se utilice, esto implica que al enviar la misma cantidad de datos se puede ahorrar más ancho de banda en comparación con OFDM. Todas estas mejoras se producen a costa de la interferencia entre subportadoras que compromete la BER (*Bit Error Rate*), disminuir esta interferencia es un desafío para los sistemas SEFDM [2], [4].

En este proyecto se implementará en MATLAB un sistema OFDM y un SEFDM. Mediante simulaciones se obtendrán resultados que permitirán determinar las mejoras e inconvenientes del sistema SEFDM con respecto a OFDM considerando diferentes variantes como el número de subportadoras, diferentes tipos de modulaciones y diferentes factores de compresión.

1.1. OBJETIVOS

El objetivo general de este proyecto técnico consiste en:

- Analizar la ganancia de ancho de banda y la degradación de la BER en un sistema SEFDM para percibir mejoras en términos de espectro con respecto a OFDM.

Los objetivos específicos de este proyecto técnico son:

- Describir los conceptos relacionados con OFDM y SEFDM.
- Implementar en MATLAB un sistema OFDM y SEFDM sobre un canal AWGN (*Additive White Gaussian Noise*).
- Evaluar los resultados en términos de BER vs E_b/N_0 (*Energy per-bit to noise power spectral density ratio*) y de PSD (*Power Spectral Density*) para OFDM y SEFDM con diferentes parámetros.
- Evaluar el ahorro de ancho de banda para varios factores de compresión.

1.2. ALCANCE

En primer lugar, se revisarán los conceptos teóricos asociados al proyecto. Se iniciará explicando OFDM y luego el nuevo sistema SEFDM, en donde se definirán los bloques que involucran a estos sistemas para la transmisión y recepción de este tipo de señales. Se revisará de la literatura existente los posibles detectores para SEFDM tal que ayude a compensar la degradación de la señal debido a la pérdida de la ortogonalidad. Luego de analizar los diferentes tipos de detectores se escogerá el detector más adecuado para su implementación en el sistema SEFDM, aquel que cuente con un mejor rendimiento en términos de BER de todos los posibles detectores que se mencionarán.

Luego, se realizará la implementación en MATLAB del transmisor y receptor de los sistemas OFDM y SEFDM, para conectar el transmisor y receptor se usará un canal

AWGN. En el sistema SEFDM se implementará el detector escogido previamente, el mismo que ayudará a la recuperación de la información y mejoramiento de la BER, debido a la pérdida de ortogonalidad de las subportadoras en el proceso de compresión.

Los resultados se presentarán en términos de BER en función de E_b/N_0 , y así establecer el grado de degradación de BER para el sistema SEFDM con respecto al sistema OFDM para diferentes valores de compresión de ancho de banda (alfa), diferentes tipos de modulaciones y para varios valores de subportadoras. Además, se presentarán resultados de BER vs E_b/N_0 sin el uso del detector a fin de ver la importancia de éste en el receptor.

Posteriormente, se comparará mediante simulaciones la PSD de los dos sistemas, en donde se visualizará el ahorro de ancho de banda en SEFDM con respecto a OFDM. Finalmente, se discutirá los resultados obtenidos en las diferentes simulaciones.

1.3. MARCO TEÓRICO

En esta sección se describen a los sistemas OFDM y SEFDM, en donde se definirán los bloques que involucran a éstos para la transmisión y recepción de este tipo de señales. Además, se explicarán brevemente las métricas usadas para medir el rendimiento de estas señales por medio de las cuales se presentarán resultados en términos de BER vs E_b/N_0 , PSD vs frecuencia.

1.3.1. SISTEMA OFDM

OFDM es una técnica de acceso multiportadora que es muy utilizada en varios tipos de comunicaciones de banda ancha, inalámbricas y móviles, las cuales ayudan al desarrollo de la sociedad [5].

Una técnica multiportadora divide al ancho de banda del canal (AB) en varias subportadoras paralelas (N) de igual ancho de banda, en donde el ancho de banda de cada subportadora está dado por $\Delta f = \frac{AB}{N}$ [6], las cuales llevarán flujos pequeños de bits paralelos resultado de repartir un flujo grande de bits, los canales se encuentran separados para que los espectros de la señal no se superpongan entre ellos [7] como se observa en la Figura 1.1 [1].

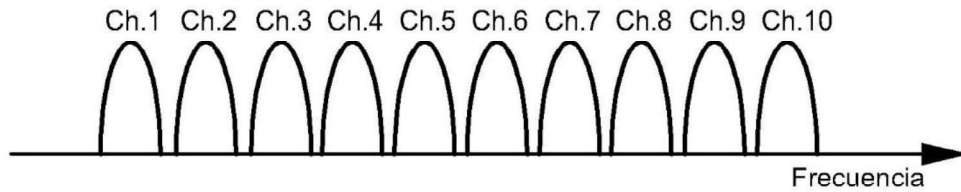


Figura 1.1. Concepto de multiprotadora

El primer concepto de transmisión multiprotadora OFDM fue introducido en el año 1966 por R. Chang de Laboratorios Bell [8] y patentado en 1970 [9], en el estudio de Chang se consiguió la transmisión simultánea de señales ortogonales de banda limitada a una velocidad de datos máxima sin interferencias entre símbolos y entre canales.

Luego, Saltzberg en el año 1967 continuando los resultados de Chang, analizó la transmisión paralela, la cual combate los efectos de distorsión de amplitud y retardo. Concluyó diciendo que se debe enfocar más en reducir el crosstalk entre canales adyacentes que en mejorar canales individuales para obtener un sistema paralelo eficiente puesto que las distorsiones debido al crosstalk tienden a dominar [10].

Posteriormente, S. Weinstein y P. Ebert plantearon la utilización de la FFT (*Fast Fourier Transform*) a los procesos de modulación y demodulación multiprotadora y así evitar el uso de bancos de osciladores de cada subportadora en el transmisor y receptor. Para combatir la ISI (*Inter Symbol Interference*) utilizaron un intervalo de guarda entre símbolos, y para la ICI (*Inter Carrier Interference*) se utilizó un enventanado de tipo coseno alzado, con esto no se obtuvo una ortogonalidad perfecta entre subportadoras, aunque se consideraba como un gran avance en aquel tiempo [11].

Finalmente, A. Peled y A. Ruiz en 1980 incluyeron el concepto de CP (*Cyclic Prefix*), el cual consiste en una extensión cíclica del símbolo OFDM que se localizaban en las bandas de guarda vacías, solucionando el problema de la ortogonalidad [12].

Por lo tanto, OFDM al ser una técnica multiprotadora divide el ancho de banda AB [Hz] en N subportadoras paralelas de igual ancho de banda, cada subportadora se asocia a una frecuencia f_k , el valor de k depende de la cantidad de subportadoras con $k = \{0, \dots, N - 1\}$. La separación entre subportadoras consecutivas es $\Delta f = AB/N$ [Hz]. En la Ecuación (1.1) [5] se presenta la expresión para determinar la f_k frecuencia de cada subportadora, donde f_0 es la frecuencia más baja. Se garantiza la ortogonalidad cuando

$T = \frac{1}{\Delta f}$ [s] o múltiplo de éste, donde T [s] es la duración del tiempo de símbolo OFDM [5].

$$f_k = f_0 + k \frac{1}{T} = f_0 + k \frac{AB}{N} \quad (1.1)$$

En efecto, cada subportadora es ortogonal al resto de tal manera que existe un ahorro de ancho de banda como se observa en la Figura 1.2, con esta superposición de subportadoras adyacentes se gana una alta eficiencia espectral [6]. De esta manera, se obtiene subcanales de banda estrecha, pudiendo existir en cada canal un desvanecimiento de frecuencia no selectivo o plano beneficiando al sistema multiportadora [7].

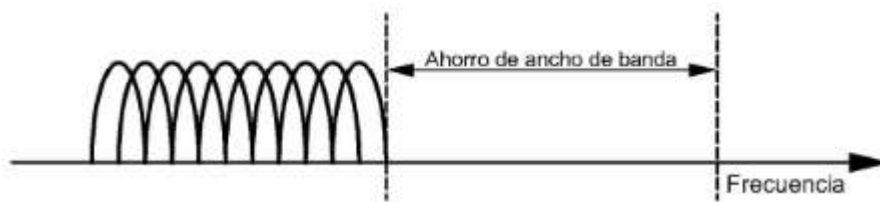


Figura 1.2. Ahorro de ancho de banda en OFDM

1.3.1.1. Transmisor y receptor de un sistema OFDM

En la Figura 1.3 se detalla un esquema general de comunicaciones que utiliza la técnica de transmisión multiportadora OFDM y a continuación se detallan cada uno de los bloques involucrados.

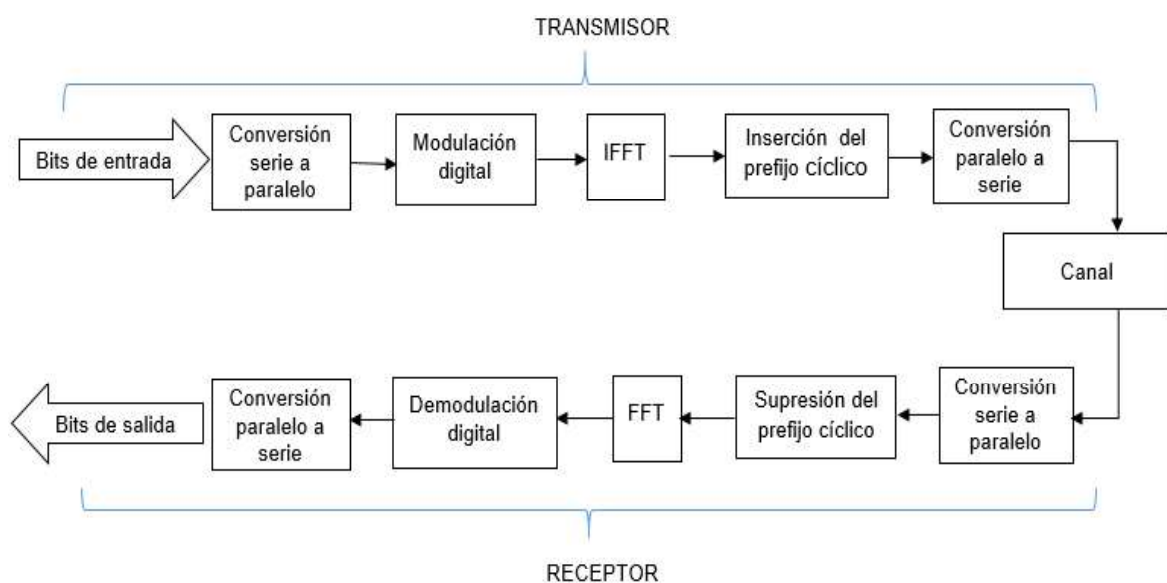


Figura 1.3. Transmisor y receptor de un sistema OFDM

a) Conversión serie a paralelo/ paralelo a serie

Los datos a la entrada del transmisor se convierten de serie a paralelo ya que posteriormente cada una de las N muestras moduladas se relacionará con las N subportadoras para formar un símbolo OFDM [13]. Al salir del transmisor se efectuará una conversión paralelo a serie para que los datos viajen por el canal. En el receptor se realizará el proceso inverso.

b) Modulación/ demodulación digital

La modulación digital se produce cuando la información digital de entrada en el transmisor se transporta utilizando una señal portadora ya sea en la amplitud, la frecuencia, la fase o una combinación de estos parámetros, la información se representa cuando la señal portadora varía de acuerdo a la información digital. Dependiendo del tipo de variación de la señal portadora se tiene diferentes tipos de modulación digital [14].

Los bits en paralelo ingresan al modulador y dependiendo del tipo de modulación, los datos se agrupan en bits para formar un número complejo usando una asignación que depende del diagrama de constelación [7]. Una de las modulaciones más populares dentro de los estándares inalámbricos y por ende de un sistema OFDM es QAM (*Quadrature Amplitude Modulation*) [1].

En una modulación QAM existe la variación simultánea de la fase y amplitud de la señal portadora, es decir, la información digital se encuentra tanto en amplitud como en fase de la portadora. Dependiendo de la agrupación de los bits, para grupos de m bits se puede obtener $M = 2^m$ salidas, donde M es el nivel o estado de modulación y cuyos valores usualmente son $M = \{4, 16, 64, \dots, M\}$ para formar modulaciones: 4-QAM, 16-QAM, 64-QAM, ..., M -QAM [15].

A medida que los niveles de modulación aumentan se transmite más bits por símbolo, pero con la desventaja de aumentar también la BER puesto que los puntos de la constelación estarán más cercanos siendo vulnerables a perturbaciones y ruido. A continuación, se explican las modulaciones que frecuentemente se utilizan:

4-QAM: Se agrupan $m = 2$ bits de entrada para formar $M = 4$ estados o niveles de modulación [15]. En la Figura 1.4 se observa la constelación 4-QAM, cuyos puntos

toman valores entre -1 y 1 tanto en el eje real (I) e imaginario (Q). Para normalizar los valores a una potencia promedio de uno, se debe multiplicar por el factor de normalización de $1/\sqrt{2}$ [1].

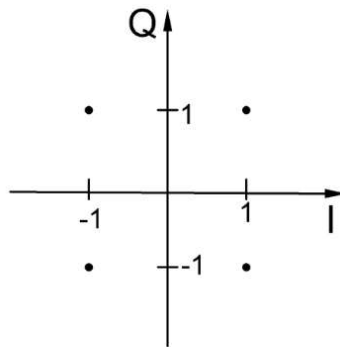


Figura 1.4. Diagrama de constelación 4-QAM

16-QAM: Se agrupan $m = 4$ bits de entrada para formar $M = 16$ estados o niveles de modulación [15]. En la Figura 1.5 se observa la constelación 16-QAM, donde los ejes I y Q toman valores: -3, -1 y 1, 3. Para normalizar los valores a una potencia promedio de uno, se debe multiplicar por el factor de normalización de $1/\sqrt{10}$ [1].

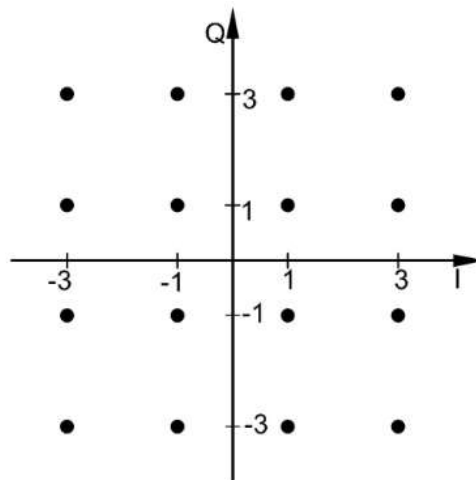


Figura 1.5. Diagrama de constelación 16-QAM

64-QAM: Se agrupan $m = 6$ bits de entrada para formar $M = 64$ estados o niveles de modulación [15]. En la Figura 1.6 se observa la constelación 64-QAM, donde los ejes I y Q toman valores: -7, -5, -3, -1 y 1, 3, 5, 7. Para normalizar los valores a una potencia promedio de uno, se debe multiplicar por el factor de normalización de $1/\sqrt{42}$ [1].

Al llegar la señal al demodulador, este es el encargado de recuperar la señal según el tipo de modulación.

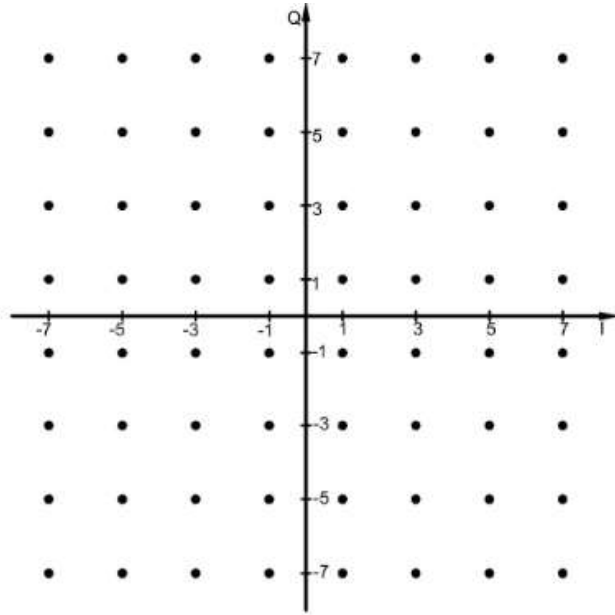


Figura 1.6. Diagrama de constelación 64-QAM

c) IFFT/FFT

Las muestras moduladas ingresan al bloque IFFT, que se transmiten sobre las N subportadoras de baja velocidad [5]. Una operación importante dentro del sistema OFDM es la DFT (*Discrete Fourier Transform*), que hace posible la representación del espectro de forma discreta [16]. La IDFT (*Inverse Discrete Fourier Transform*) se encuentra en el transmisor y la DFT en el receptor. Mediante la IDFT es posible pasar del dominio de la frecuencia al dominio del tiempo y en el receptor realizar el proceso contrario [5].

En las Ecuaciones (1.2) y (1.3) [16] se definen la señal OFDM en transmisión empleando la IDFT y la señal en el receptor utilizando DFT, respectivamente, en donde $X(k)$ es la señal en el dominio de la frecuencia, $x[n]$ es la señal en el dominio del tiempo, n es el índice temporal y k el índice frecuencial.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kn}{N}} \quad n = 0, \dots, N - 1 \quad (1.2)$$

$$X(k) = \sqrt{N} \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}} \quad k = 0, \dots, N - 1 \quad (1.3)$$

Para implementar la DFT/IDFT hay métodos eficientes que evalúan con un menor número de operaciones como la IFFT (*Inverse Fast Fourier Transform*) que se utiliza en

el transmisor y FFT en el receptor. El número de operaciones que se necesitan para implementar la DFT de N puntos es $N \log_2 N$ con la FFT, mientras que para hacerlo directamente se requieren N^2 operaciones [16].

Con la IFFT se toma una muestra de cada subportadora que resultan en un símbolo OFDM, calcula el flujo de datos del dominio de tiempo a transmitir, asegura que no existirán interferencias entre las subportadoras, se evitan transmisores y receptores costosos por el número de oscilares que se necesitaría para generar las frecuencias de las subportadoras. Con la FFT, las señales que ingresan al receptor se otorgan a su respectiva subportadora y así recuperar los flujos de datos de cada muestra tomada [17].

d) Inserción/ supresión del prefijo cíclico

Cuando la señal OFDM pasa a través de un canal multitrayecto o de tiempo disperso se produce ISI y si también se pierde la ortogonalidad de las subportadoras resulta la ICI, la solución a estos inconvenientes es la inserción de una banda de guarda al inicio de cada uno de los símbolos para evitar la ISI. Y para mitigar la ICI, a esta banda de guarda se llena con una extensión cíclica llamada prefijo cíclico (CP) que consiste en rellenar la banda de guarda con una copia de las últimas muestras del símbolo [6].

En la Figura 1.7, T_s es el tiempo o longitud del símbolo OFDM sin CP, T_{cp} es la longitud del prefijo cíclico, entonces $T = T_{cp} + T_s$ representa la longitud del símbolo OFDM transmitido con CP [6].

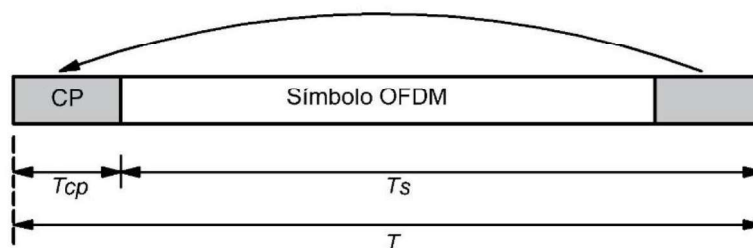


Figura 1.7. Símbolo OFDM con prefijo cíclico

En el receptor se realiza la supresión del prefijo cíclico que se insertó en el transmisor.

El prefijo cíclico debe ser igual o mayor a la parte significativa de la respuesta al impulso experimentada por la señal transmitida por el canal inalámbrico para evitar la ISI e ICI [6].

1.3.1.2. Ventajas e inconvenientes de OFDM

OFDM es una técnica de transmisión muy utilizada dentro de las comunicaciones inalámbricas ya que cuenta con las siguientes ventajas [1], [18]:

- Cuenta con una alta eficiencia espectral debido a la ortogonalidad de las subportadoras que hace que se ocupe un menor ancho de banda en comparación con otras técnicas multiportadoras, como FDM (*Frequency Division Multiplexing*) y así transmitir elevadas tasas de transmisión.
- Al ser un sistema multiportadora, existe un aumento de robustez contra el desvanecimiento de frecuencia selectiva, de esta manera no se verá afectada toda la información contenida en las subportadoras.
- Con el uso del prefijo cíclico se evita la ISI e ICI.
- Al utilizar los algoritmos de IFFT y FFT dentro del transmisor y receptor hacen posible que el sistema OFDM sea más simple y eficiente de implementar.

Sin embargo, también cuenta con las siguientes desventajas [1], [5]:

- Se requiere sincronización precisa en tiempo, STO (*Symbol Time Offset*), y frecuencia, CFO (*Carrier Frequency Offset*), para mantener la ortogonalidad.
- En el dominio del tiempo se produce la PAPR (*Peak-to-Average Power Ratio*), que son grandes fluctuaciones en la envolvente de la señal, las cuales afectan al rendimiento máximo de determinados componentes de hardware.
- Disminución de la eficiencia espectral por la adición del prefijo cíclico.
- Es un limitante en redes 5G, en donde se requiere un uso eficiente del espectro, SEFDM es una de las técnicas que busca ahorrar ancho de banda con respeto a OFDM y que podría ser una técnica de transmisión utilizada dentro de las redes 5G.

1.3.2. SISTEMA SEFDM

SEFDM es una técnica de acceso multiportadora que se fundamenta en OFDM, tiene una alta eficiencia espectral ya que al comprimir las subportadoras habrá un ahorro de ancho de banda en comparación con OFDM, esta gran ventaja se produce a costa de la degradación de la BER producida por la interferencia entre subportadoras al perder la ortogonalidad [19], [20].

La idea de aumentar la eficiencia espectral nace con la necesidad de mejorar para los futuros estándares inalámbricos, todo esto basado en sistemas multiportadoras que se utilizan actualmente en varios tipos de comunicaciones [4].

En el año 2002 fue propuesta una técnica semi-ortogonal llamada Fast-OFDM dicha técnica fue pensada originalmente para sistemas inalámbricos y en la actualidad tiene su aplicación en sistemas ópticos, brinda el doble de eficiencia espectral en comparación con OFDM [21].

Posteriormente en el año 2003 basado en la técnica anterior, se desarrolló el sistema SEFDM, este sistema fue propuesto por M. Rodrigues and I. Darwazeh [22], cumple el mismo objetivo de Fast-OFDM, de reducir el espacio entre subportadoras. Para lograr esto, SEFDM hace uso de un factor de compresión de ancho de banda denotado por α , el cual toma valores en el intervalo de $]0,1[$. Entonces, la separación entre subportadoras consecutivas está dada por $\Delta f = \frac{\alpha}{T}$, donde T [s] es el tiempo de símbolo SEFDM. Esta compresión implica que el espacio entre subportadoras de OFDM se comprimirá de acuerdo al factor α . De hecho, se puede decir que OFDM es un caso especial de SEFDM cuando se considera $\alpha=1$ [2], [4].

Esta compresión de las subportadoras quiere decir que, al enviar una misma cantidad de información, el ahorro de ancho de banda en SEFDM está dado por $(1-\alpha) * 100\%$ en comparación con OFDM. En la Figura 1.8 se observa el espectro de potencia de SEFDM para un sistema de $N = 1024$ subportadoras con diferentes valores de $\alpha = \{1, 0.8, 0.5\}$. Se observa que, mientras más bajo es el valor de alfa, más compresión se obtiene [4].

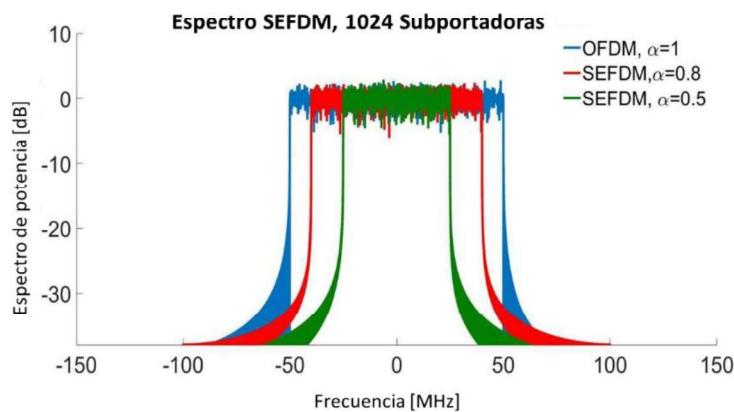


Figura 1.8. Power Spectral Density para varios valores de α

Por lo tanto, se tiene un mejoramiento en la eficiencia espectral, pero debido a la pérdida de ortogonalidad de las subportadoras en el proceso de compresión se produce ICI, lo que conlleva a una degradación de la BER. La eliminación de la ICI fue y continúa siendo un gran desafío en el sistema SEFDM por lo que surge la necesidad de introducir un detector en el receptor para compensar esta degradación y así recuperar la información y mejorar la BER [4].

Para realizar comparaciones entre los sistemas OFDM y SEFDM, estos sistemas deben usar el mismo ancho de banda y dependiendo de la compresión se tendrán más subportadoras ocupando el mismo ancho de banda [23], [24]. Esta relación está dada por $N_{SEFDM} = \frac{N_{OFDM}}{\alpha}$, donde N_{OFDM} representa el número de subportadoras en OFDM y N_{SEFDM} el nuevo número de subportadoras que se incrementarán con SEFDM.

En la Figura 1.9 [23] se aplica esta relación. Así, para sistemas con el mismo ancho de banda, con $N_{OFDM} = 16$ y $\alpha = 0.8$, el número de subportadoras se incrementa a $N_{SEFDM} = 20$. En consecuencia, la velocidad de datos se incrementa por un factor de $\frac{1-\alpha}{\alpha} * 100\%$ y por ende se alcanza una alta eficiencia espectral [23].

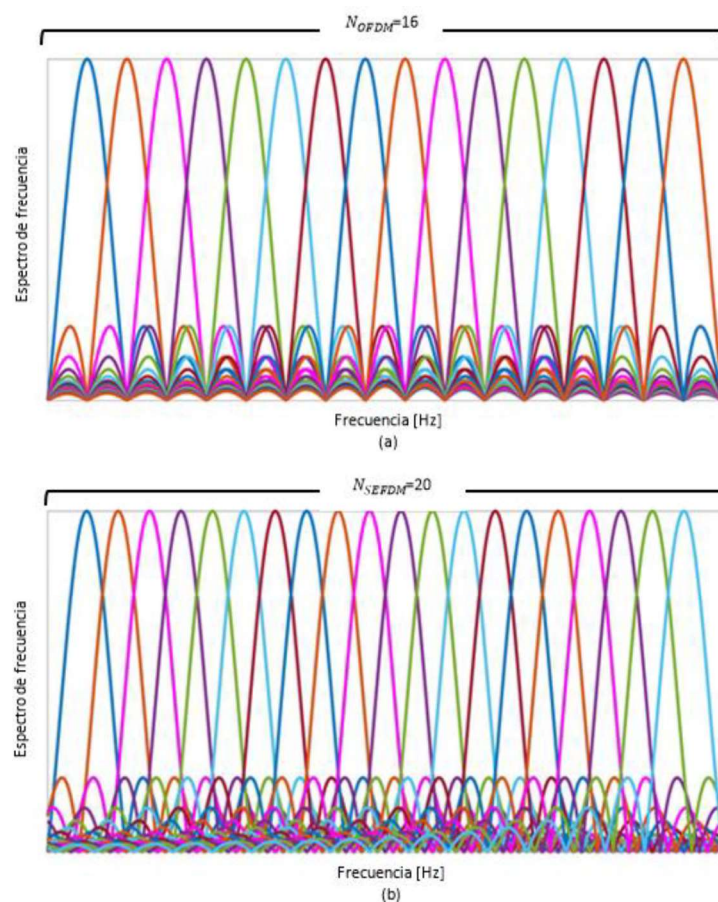


Figura 1.9. Espectro de frecuencia de a) OFDM ($\alpha = 1$) y b) SEFDM ($\alpha = 0.8$)

Como ya se había mencionado, OFDM al ser una técnica ortogonal es sensible a los errores de sincronización, los cuales producen la ICI al perder la ortogonalidad que a su vez se ve reflejado en la degradación de la BER. Sin embargo, SEFDM es una técnica no-ortogonal en la cual, los efectos de los errores en sincronización estarán presentes, es decir el sistema tendrá ICI en todos los casos [25].

1.3.2.1. Transmisor y receptor de un sistema SEFDM

En la Figura 1.10 se detalla un esquema general de una comunicación que utiliza la técnica de transmisión multiportadora SEFDM y a continuación se detallan cada uno de estos bloques.

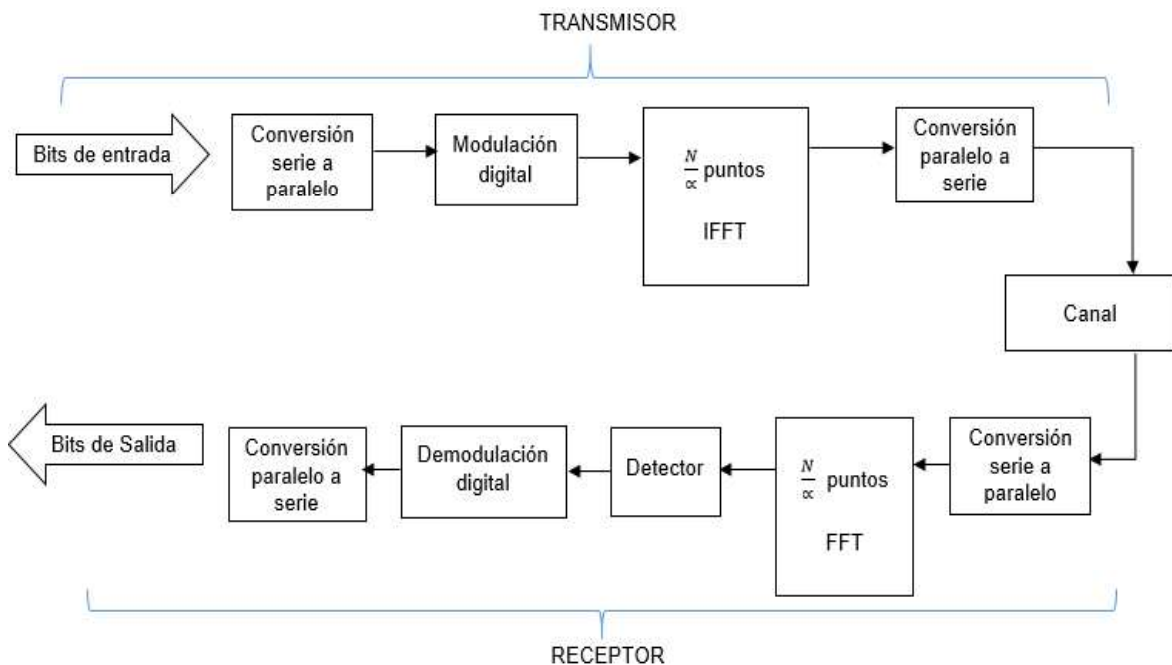


Figura 1.10. Transmisor y receptor de un sistema SEFDM

a) Conversión serie a paralelo/ paralelo a serie

Al ser SEFDM una técnica multiportadora, se requiere convertir los datos de serie a paralelo antes de ingresar al bloque IFFT y a la salida al canal se realiza el proceso inverso para que los datos viajen a través del canal en serie. En el receptor se aplicará el mismo concepto.

b) Modulación/ demodulación digital

Como ya se explicó en OFDM, los datos se agrupan en bits para formar un número complejo según el tipo de modulación (diagrama de constelación) [7]. En el sistema SEFDM también una de las modulaciones más populares es QAM.

Cuando el nivel de modulación incrementa, los sistemas son capaces de brindar una alta eficiencia espectral; sin embargo, aumenta también la BER ya que los puntos de la constelación estarán más cercanos, y así las modulaciones de mayor orden son consideradas menos resistentes al ruido e interferencia [24].

Al llegar la señal al demodulador, este es el encargado de recuperar la señal según el tipo de modulación.

c) IFFT/FFT

Para construir una señal SEFDM se multiplexa un grupo de símbolos complejos en subportadoras paralelas [26].

Como ya se detalló en OFDM, las muestras moduladas ingresan al bloque IFFT, que serán transmitidas sobre las N subportadoras de baja velocidad [7]. La DFT, que hace posible la representación del espectro de forma discreta, es también una operación importante dentro del sistema SEFDM [27]. La IDFT se encuentra en el transmisor y la DFT en el receptor. Mediante la IDFT es posible pasar del dominio de la frecuencia al dominio del tiempo y en el receptor realizar el proceso contrario. Para su fácil implementación se usan las operaciones como la IFFT que se utiliza en el transmisor y FFT en el receptor [28].

Dentro de este esquema las N subportadoras y el operador IFFT son usados para generar la señal en el dominio del tiempo SEFDM [28]. El objetivo es tener un vector de tamaño $\frac{N}{\alpha}$ con $\alpha < 1$ para aplicar la IFFT y posteriormente ignorar las últimas muestras.

Para mejor comprensión del bloque IFFT se explica en el siguiente ejemplo, con $N = 12$ y $\alpha = 0.8$. Como ya se mencionó en la Figura 1.10 antes del bloque IFFT se encuentra el bloque de modulación digital, a la salida de este bloque se tiene el vector A de tamaño 12×1 resultado de una modulación 4-QAM como se observa en la Figura 1.11.

```

A =
  1.0000 - 1.0000i
 -1.0000 - 1.0000i
  1.0000 + 1.0000i
 -1.0000 - 1.0000i
  1.0000 - 1.0000i
 -1.0000 - 1.0000i
  1.0000 + 1.0000i
  1.0000 + 1.0000i
 -1.0000 - 1.0000i
  1.0000 - 1.0000i
  1.0000 + 1.0000i
  1.0000 - 1.0000i

```

Figura 1.11. Vector a la salida del bloque de modulación digital

Luego, al final del vector A se inserta un número específico de ceros en función de $\frac{1-\alpha}{\alpha} * N$, es decir $ceros = \frac{1-0.8}{0.8} * 12$. Para este ejemplo, $ceros = 3$ como se observa en la Figura 1.12.

```

ceros =
  0
  0
  0

```

Figura 1.12. Vector ceros

Por lo tanto, se obtiene el vector B , resultado de insertar los ceros al final del vector A como se observa en la Figura 1.13.

```

B =
  1.0000 - 1.0000i
 -1.0000 - 1.0000i
  1.0000 + 1.0000i
 -1.0000 - 1.0000i
  1.0000 - 1.0000i
 -1.0000 - 1.0000i
  1.0000 + 1.0000i
  1.0000 + 1.0000i
 -1.0000 - 1.0000i
  1.0000 - 1.0000i
  1.0000 + 1.0000i
  1.0000 - 1.0000i
  0.0000 + 0.0000i
  0.0000 + 0.0000i
  0.0000 + 0.0000i

```

Figura 1.13. Vector B

Posteriormente, al vector B se le aplica la operación IFFT como se observa en la Figura 1.14. Por lo tanto, las muestras insertadas estarán ubicadas en cualquier posición luego de aplicar la IFFT [4]. C es el vector que se obtiene al aplicar la IFFT y es de tamaño $\frac{N}{\alpha}$, es decir $C = \frac{12}{0.8}$. Para este caso $C = 15$.

```
C =
 0.2667 - 0.2667i
-0.0417 - 0.2184i
 0.0678 + 0.1049i
-0.0917 - 0.0351i
 0.3114 - 0.4986i
-0.0488 + 0.0488i
 0.1193 - 0.1977i
 0.5670 - 0.0389i
 0.1873 + 0.0872i
-0.3096 + 0.3879i
 0.1821 - 0.1821i
-0.3079 - 0.0360i
 0.3486 - 0.2218i
-0.1082 + 0.0273i
-0.1424 + 0.0391i
```

Figura 1.14. Aplicación de la IFFT al vector B

Finalmente, serán ignoradas las últimas muestras del vector C y así formar el vector D de acuerdo a $\frac{N}{\alpha} - \left(\frac{N}{\alpha} - N\right)$, es decir $D = \frac{12}{0.8} - \left(\frac{12}{0.8} - 12\right)$, para este ejemplo $D = 12$, como se observa en la Figura 1.15. D es el vector que se transmite al canal.

```
D =
 0.2667 - 0.2667i
-0.0417 - 0.2184i
 0.0678 + 0.1049i
-0.0917 - 0.0351i
 0.3114 - 0.4986i
-0.0488 + 0.0488i
 0.1193 - 0.1977i
 0.5670 - 0.0389i
 0.1873 + 0.0872i
-0.3096 + 0.3879i
 0.1821 - 0.1821i
-0.3079 - 0.0360i
```

Figura 1.15. Vector D

Entonces, la señal SEFDM a la salida del transmisor $x[n]$ puede ser representada por la Ecuación (1.4) [28]:

$$x[n] = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn\alpha} \quad n = 0, \dots, N - 1, \quad (1.4)$$

donde, $X(k)$ es la señal en el dominio de la frecuencia, $x[n]$ es la señal en el dominio del tiempo, n es el índice para el tiempo, k el índice para frecuencia y α es el factor de compresión.

En el receptor se procederá de la misma manera que en el transmisor con la diferencia de que aquí se utiliza la operación FFT, y de esta manera pasar la señal al dominio de la frecuencia, dicha señal puede ser representada como en la Ecuación (1.5) [28]:

$$X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn\alpha} \quad k = 0, \dots, N - 1 \quad (1.5)$$

d) Canal

Para transmitir la señal al receptor en las comunicaciones inalámbricas, se usa como medio de transmisión el aire, por el cual viaja la señal convertida en energía electromagnética. Para representar un canal inalámbrico existen varios modelos. El modelo más simple es el canal AWGN, el cual a la señal que sale del transmisor $x(t)$ se añade ruido AWGN, denotado por $z(t)$ por lo tanto, al receptor llega la señal $r(t)$ dado por la Ecuación (1.6) [29].

$$r(t) = x(t) + z(t) \quad (1.6)$$

En consecuencia, el proceso de recepción de SEFDM puede ser representado por un modelo discreto utilizando matrices, donde la señal transmitida, utilizando SEFDM incluirá la interferencia entre subportadoras debido a la pérdida de la ortogonalidad. Esta señal recibida está dada por la Ecuación (1.7) [29], donde R es un vector N -dimensional que representa los símbolos recibidos distorsionados, después de la FFT, C describe la interferencia entre subportadoras, S denota el vector N -dimensional y representa los símbolos transmitidos y Z representa el vector N -dimensional del ruido Gaussiano [29].

$$R = CS + Z \quad (1.7)$$

e) Detector

La pérdida de la ortogonalidad a la que conlleva a la interferencia entre las subportadoras en el sistema SEFDM, convierte la recuperación de la señal en un problema demasiado complejo, para ello es necesario introducir un detector para mitigar dicha ICI. La complejidad de la detección crece con el aumento del tamaño del sistema (N) y el nivel de modulación [19]. Un detector intenta estimar la señal transmitida a partir de los datos recibidos o cierta información conocida.

Varios detectores para la señal SEFDM han sido propuestos. Se puede mencionar los detectores lineales: ML (*Maximum Likelihood*) [19], ZF (*Zero Forcing*), MMSE (*Minimum Mean Square Error*), también se ha encontrado a TSVD (*Truncated Singular Value Decomposition*) [27], [30], ID (*Iterative Detection*) [24].

ML

Para estimar la señal recibida dada por \hat{S}_{ML} se utiliza optimización matemática. La detección ML está dada por el problema de mínimos cuadrados detallado en la Ecuación (1.8), donde $\|\cdot\|$ es la norma euclidiana, S son los símbolos detectados, O es la cardinalidad de la constelación, N es el número de subportadoras y por lo tanto, O^N es el conjunto de todas las posibles combinaciones de símbolos [31].

El objetivo de ML es probar todas las posibles combinaciones de símbolos de entrada para cada símbolo SEFDM recibido, la solución adecuada es la que tiene la norma euclidiana más pequeña, es decir que la diferencia entre la señal transmitida que se le añadió una distorsión CS y la señal recibida R sea mínima [31].

La detección utilizando ML es demasiado compleja [32]. Ofrece un rendimiento óptimo, aunque en los sistemas, incluso en los de tamaño moderado, requieren grandes recursos que pueden ser poco realistas [27]. La complejidad de ML aumenta exponencialmente con el número de subportadoras N y la cardinalidad de la constelación O [4].

$$\hat{S}_{ML} = arg \min_{S \in O^N} \|R - CS\|^2 \quad (1.8)$$

ZF y MMSE

ZF se define en la Ecuación (1.9), la cual muestra la división de la señal recibida para la distorsión, es decir intenta quitarle la distorsión a la señal recibida y así estimar la señal transmitida, $\lfloor \cdot \rfloor$ es un operador que redondea al argumento al punto más cercano de la constelación [31]. MMSE se define en la Ecuación (1.10), en donde está quitando la distorsión a la señal recibida incluyendo un valor de SNR (*Signal to Noise Ratio*), $[\cdot]^*$ denota la transposición conjugada de C y el valor de I es la matriz identidad [33].

ZF y MMSE presentan restricciones en el ahorro de ancho de banda para producir una BER aceptable, esto debido a que la detectabilidad de las señales está restringida por el mal acondicionamiento de la matriz de correlación de subportadoras [27], estas dos técnicas son más deficientes que ML [4].

$$\hat{S}_{ZF} = \lfloor C^{-1}R \rfloor \quad (1.9)$$

$$\hat{S}_{MMSE} = \left\lfloor C^* \left(CC^* + \frac{1}{SNR} I \right)^{-1} R \right\rfloor \quad (1.10)$$

TSVD

TSVD es considerado un detector sub-óptimo que presenta mejores estimaciones según estudios de modelado y simulación proporcionando un atractivo rendimiento. En [28], el diseño de TSVD utiliza FPGAs (Field Programmable Gate Arrays) y los resultados se ven reflejados en la reducción en la complejidad [27], TSVD supera el problema del mal acondicionamiento de ZF y MMSE y de esta manera mejora el rendimiento en términos de BER manteniendo una complejidad lineal del sistema [4], [33].

TSVD es un método para invertir matrices mal acondicionadas y de esta manera realizar la detección de señales SEFDM, para lograr esto, genera una matriz inversa aproximada de una matriz con mal acondicionamiento [27], [33].

El acondicionamiento de una matriz se evalúa por su número de condición, la cual se define como la relación del valor singular máximo al mínimo [33]. Como el número de condición de una matriz crece, cualquier solución basada en el inverso de la matriz C (matriz de coeficiente de correlación cruzada) es muy sensible incluso a perturbaciones muy pequeñas [27], [33].

El detector TSVD se basa en la Ecuación (1.11) [34], donde C_ξ es pseudo-inverso aproximado de C .

$$\hat{S}_{TSVD} = |C_\xi R| \quad (1.11)$$

El pseudo-inverso aproximado de C se detalla en la Ecuación (1.12) [34]:

$$C_\xi = V \sum_{\xi}^{-1} U^* , \quad (1.12)$$

donde, $\sum_{\xi}^{-1} = \text{diag}\left(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_\xi}, 0, \dots, 0\right)$, ξ es el índice de truncamiento, U y V son matrices unitarias. El índice de truncamiento define el número de valores singulares que se aceptan y de esta manera se determina la calidad de la matriz generada la cual se verá reflejada en el rendimiento del sistema [33] [34].

ID

La ICI en SEFDM puede considerarse como una distorsión no lineal, ID es un método que ayuda a la compensación de la distorsión de señales causada por operaciones lineales o no lineales en el sistema SEFDM. De manera general, su idea se fundamenta en que si una señal S es distorsionada por una distorsión descrita en una matriz G entonces se tendrá una nueva señal distorsionada \hat{S} definida por la Ecuación (1.13), la señal original podrá ser recuperada a partir de la señal distorsionada siguiendo un proceso de iteración [28].

$$\hat{S} = GS \quad (1.13)$$

ID presenta una mejor inmunidad contra la interferencia. Su funcionamiento se basa en la recuperación de las señales distorsionadas de manera iterativa, la interferencia ocasionada en las señales puede ser descrita por una matriz de correlación C . El proceso de iteración se expresa en la Ecuación (1.14) [24], el objetivo es que, a través de esta ecuación, reducir de manera gradual la interferencia luego de cada iteración [20].

$$S_n = R + (e - C) S_{n-1} , \quad (1.14)$$

donde, S_n es un vector N -dimensional de símbolos recuperados luego de n iteraciones, S_{n-1} es un vector N -dimensional de símbolos estimados después de $n - 1$ iteraciones, R es un vector N -dimensional de símbolos distorsionados después de demodular los símbolos recibidos utilizando la FFT, e es una matriz de identidad $N \times N$, C es una

El mapeo suave tiene un mejor rendimiento que el mapeo duro, y si este mapeo suave es cambiado dinámicamente es mucho mejor, por lo tanto, este mapeo suave es de gran importancia dentro del detector iterativo para lograr una BER deseada [20].

Al realizar una comparación con TSVD, se puede decir que TVSD es un detector de mapeo duro mientras que ID al ser un método iterativo que utiliza un mapeo suave, genera una estimación inicial más precisa que TSVD [29]. La complejidad de ID se basa en el algoritmo iterativo mientras TVSD resulta por inversión directa de la matriz, por lo tanto, las operaciones que involucran a ID son menos que las de TSVD [24].

Luego de analizar los diferentes tipos de detectores en varias literaturas se escoge al detector ID para su implementación en el sistema SEFDM. Por ser el que cuenta con un mejor rendimiento en términos de BER de todos los detectores mencionados.

1.3.2.2. Ventajas e inconvenientes de SEFDM

- Para un mismo ancho de banda dado, SEFDM ofrece un mayor rendimiento que OFDM ya que se evidencia un ahorro en el espectro con respecto a OFDM, el cual es un recurso escaso y costoso [24], [29].
- Al ir reduciendo el valor de α , existe una mejora de la eficiencia espectral a costa de la complejidad de la detección [35].
- Al igual que en OFDM la PAPR causa una degradación significativa del rendimiento del amplificador la potencia [28].

1.3.3. PARÁMETROS INVOLUCRADOS PARA LA MEDICIÓN DEL RENDIMIENTO DE OFDM Y SEFDM

SNR

Es la relación del nivel de potencia de una señal, dado por P_s , con respecto al nivel de potencia del ruido denotado por P_r . Se mide en unidades de decibelios (dB), los cuales indican en que cantidad la señal deseada excede el nivel de ruido. Por lo tanto, establece la calidad de un sistema, mientras más alto sea el valor de SNR mejor será la calidad del sistema de comunicaciones. Generalmente este parámetro se evalúa en el receptor. La Ecuación (1.17) [37] muestra la expresión para calcular la SNR [14], [37].

$$SNR_{dB} = 10 * \log_{10} \frac{P_s}{P_r} \quad (1.17)$$

E_b/N_o

Este parámetro indica la relación de la energía de la señal por bit (E_b) y la densidad de potencia del ruido por hertz (N_o). Se utiliza para determinar las tasas de datos digitales y las tasas de error. Se relaciona con la SNR y también ayuda a evaluar el rendimiento de un sistema de comunicación digital. Al igual que la SNR, se mide en unidades de decibelios (dB) [37]. La Ecuación (1.18) [37] indica la fórmula para determinar el E_b/N_o , donde P_s es la potencia de la señal, T_b es el tiempo requerido para enviar un bit, K es la constante de Boltzmann, T es la temperatura, V_{Tx} es la tasa de datos [37].

$$\frac{E_b}{N_o} (dB) = \frac{P_s T_b}{K T} = \frac{P_s}{K T V_{Tx}} \quad (1.18)$$

Si el valor de $\frac{E_b}{N_o}$ aumenta el sistema será mejor en términos de BER. A mayor velocidad de transmisión las condiciones son más desfavorables, es decir habrá mayor cantidad de bits errados por lo tanto la BER aumenta.

BER

La BER está determinada por la relación entre el número de bits errados y número total de bits enviados en el transmisor. La Ecuación (1.19) [38] detalla este concepto. Para evaluar el número de bits errados se realiza una comparación entre los bits que ingresan al transmisor con los bits a la salida del receptor.

$$BER = \frac{\text{número de bits errados}}{\text{número total de bits enviados}} \quad (1.19)$$

PSD

La densidad espectral de potencia determina la contribución en potencia de las diferentes componentes de frecuencia que constituyen la señal.

El ancho de banda está determinado por el espectro de la señal modulada que generalmente se presenta como una gráfica de PSD vs frecuencia como se observa en la Figura 1.17 [39].

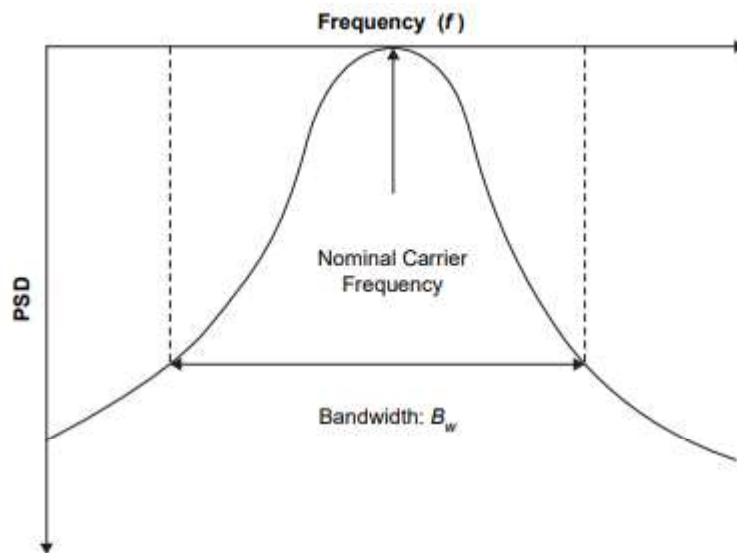


Figura 1.17. PSD vs frecuencia

De manera ideal la PSD debería ser cero fuera del ancho de banda; sin embargo, en la práctica esto no es posible ya que el espectro se extiende hasta infinitamente más allá del ancho de banda [39].

2. METODOLOGÍA


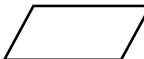
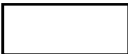

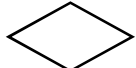


En este capítulo se explicará la programación necesaria para la implementación de los sistemas OFDM y SEFDM, para dicha implementación se utilizó el software de simulación MATLAB en su versión R2018b. Los procedimientos realizados en los scripts se detallarán a través de diagramas de flujo, tablas, en las cuales se indican varios parámetros que ayudan a entender los comandos y funciones utilizadas.

Los scripts que se detallarán posteriormente ayudarán a obtener resultados de:

- BER vs Eb/No en un sistema OFDM
- PSD en un sistema OFDM
- BER vs Eb/No en un sistema SEFDM
- PSD en un sistema SEFDM

La nomenclatura para los diagramas de flujo se especifica en la Tabla 2.1.

Tabla 2.1. Nomenclatura para diagramas de flujo

Figura	Significado
	Inicio y final del código
	Entrada de datos
	Procesamiento de datos
	Preparación para bucle
	Decisión
	Contador
	Resultados

2.1. SISTEMA OFDM

El sistema OFDM está compuesto por el transmisor y receptor. Para comprender la implementación de éste es necesario definir las variables involucradas, las cuales se detallan en la Tabla 2.2.

Tabla 2.2. Definición de variables de ingreso en sistema OFDM

Variable	Definición
N	Número de subportadoras
m	Número de bits agrupados según tipo de modulación: 2:4-QAM, 4:16-QAM, 6:64-QAM
N_{trials}	Número de iteraciones
E_b/N_o	Valor de E_b/N_o

2.1.1. TRANSMISOR OFDM

En el transmisor se ingresan bits aleatorios para entregar a la salida un símbolo OFDM. En la Figura 2.1 se detallan los bloques involucrados en el transmisor y la Figura 2.2 muestra el diagrama de flujo que involucran los diferentes procesos.

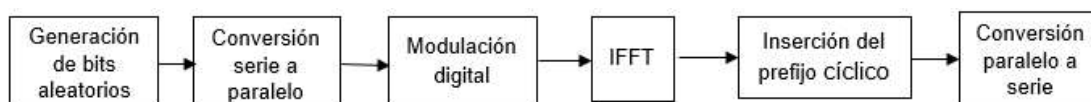


Figura 2.1. Diagrama de bloques del transmisor OFDM

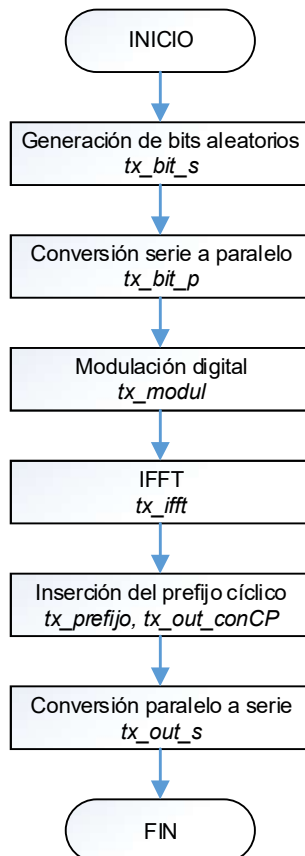


Figura 2.2. Diagrama de flujo del transmisor OFDM

a) Generación de bits aleatorios

Para la generación de bits aleatorios se utiliza el comando *randi*, el cual genera valores aleatorios sean 0 o 1 [40], por tanto, se crea un vector con nombre *tx_bit_s*, el tamaño del vector generado es de $N \times m$.

b) Conversión serie a paralelo

Para cambiar un vector fila a un vector columna se realiza mediante el comando *reshape* [41], este vector se almacena en la variable *tx_bit_p*.

c) Modulación digital

Dependiendo de la agrupación de los bits almacenados en la variable *tx_bit_p*, se tendrán diferentes tipos o niveles de modulación: 4-QAM, 16-QAM, 64-QAM. Para esto es necesario definir la variable $M = 2^m$, donde m indica el número de bits que se agrupan para formar una muestra modulada y M es el nivel, estado u orden de modulación. El comando *qammod* se utiliza para realizar la modulación del vector *tx_bit_p*, en donde se especifica el orden de modulación M y los argumentos '*InputType*', '*bit*' que indican que el vector que ingresa debe contener valores binarios y el tamaño del vector debe ser un múltiplo entero de $\log_2 M$, es decir múltiplo entero de m [42]. En la variable *tx_mod* se almacenan las muestras moduladas.

d) IFFT

Para que las muestras moduladas pasen del dominio de la frecuencia al dominio del tiempo se utiliza el comando *ifft*, el cual calcula la transformada discreta de Fourier inversa de una señal utilizando el algoritmo IFFT [43], se especifica el vector *tx_mod* y el tamaño de dicho vector. La variable *tx_ifft* almacena el símbolo OFDM generado.

e) Inserción del prefijo cíclico

La inserción del prefijo cíclico consiste en copiar las últimas muestras del símbolo OFDM al inicio de cada símbolo, con esto se evita la ISI e ICI. Cada tecnología utiliza un determinado tamaño del prefijo cíclico. En este trabajo el $CP = N/4$, ya que la mayoría de las aplicaciones utiliza esta duración de periodo de guarda ($1/4$ de símbolo OFDM) [3].

Por lo tanto se toman las $N/4$ muestras finales, es decir, desde la posición $(N - (CP - 1))$ hasta la posición N del vector tx_ifft , esto se almacena en la variable $tx_prefijo$, y luego se copian al inicio del símbolo OFDM y así formar un nuevo vector denominado tx_out_conCP .

f) Conversión paralelo a serie

Para cambiar un vector columna a un vector fila se realiza mediante el comando *reshape* [41], este vector se almacena en la variable tx_out_s .

2.1.2. CANAL AWGN

AWGN es un modelo de canal que representa los efectos de varios procesos aleatorios que suceden en la naturaleza, que se lo interpreta como ruido, debido a esto la señal sufre cierta degradación. Al símbolo OFDM que sale del transmisor se le añade el ruido AWGN y esta nueva señal es la que llega al receptor.

Para añadir ruido a la señal que sale del transmisor tx_out_s se utiliza el comando *awgn*, el canal AWGN es un canal aleatorio que se almacena en la variable *canal*, para su generación necesita de la señal tx_out_s , el valor de *SNR* y con los parámetros 'measured', 'db' que significa que el nivel de la señal de entrada se calcula para determinar el nivel de ruido apropiado en función del valor *SNR* [44]. Para determinar el valor de *SNR* se necesita los valores de $E_b N_0$ y m , en donde $SNR = E_b N_0 + 10 \log_{10} m$.

2.1.3. RECEPTOR OFDM

Al símbolo que atraviesa por el canal AWGN ingresa al receptor, en donde se realiza los procesos inversos a los explicados en el transmisor. En la Figura 2.3 se detallan cada uno de los bloques implicados en el receptor y la Figura 2.4 muestra el diagrama de flujo que involucra los diferentes procesos en el receptor OFDM.

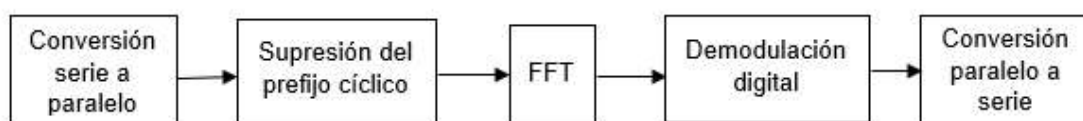


Figura 2.3. Diagrama de bloques del receptor OFDM

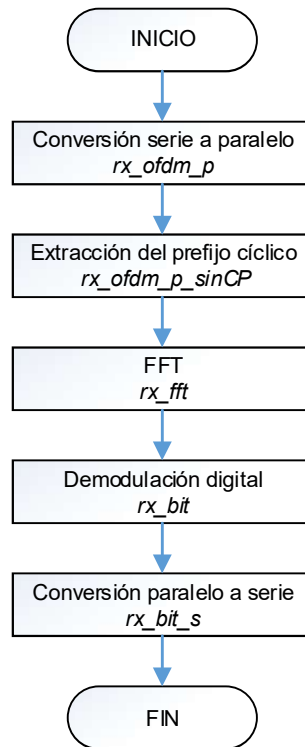


Figura 2.4. Diagrama de flujo del receptor OFDM

a) Conversión serie a paralelo

Para cambiar un vector fila a un vector columna se realiza mediante el comando *reshape* [41], este vector se almacena en la variable *rx_ofdm_p*.

b) Extracción del prefijo cíclico

Se almacena en la variable *rx_ofdm_p_sinCP*, el símbolo recibido sin prefijo cíclico, es decir se eliminan las copias de las muestras que fueron añadidas al inicio del símbolo, para esto se toman las muestras desde la posición $\left(\frac{N}{4} + 1\right)$ hasta la posición N del símbolo recibido.

c) FFT

Al símbolo sin CP, *rx_ofdm_p_sinCP*, se le aplica el comando *fft* que calcula la transformada discreta de Fourier de una señal utilizando un algoritmo FFT [45] y así pasar al símbolo del dominio del tiempo al dominio de la frecuencia, esto se almacena con el nombre *rx_fft* de tamaño N .

d) Demodulación digital

El proceso de demodulación se realiza mediante el comando *qamdemod* [46], en donde como resultado se obtendrán bits que se guardarán en la variable *rx_bit*, para esto se debe ingresar la señal a demodular *rx_fft*, el orden de modulación M , y se especifica la salida en bits con el argumento '*OutputType','bit*'.

e) Conversión paralelo a serie

Al final se realiza la conversión del vector columna *rx_bit*, a un vector fila mediante el comando *reshape* [41], este vector se almacena en la variable *rx_bit_s*.

2.1.4. RESULTADOS EN OFDM

Una vez que se ha implementado el transmisor y receptor se procede a evaluar los resultados en términos de BER vs E_b/N_0 y de PSD.

2.1.4.1. Cálculo de errores y BER en OFDM

La Figura 2.5 muestra el diagrama de flujo del sistema OFDM implementado en MATLAB para obtener resultados de BER vs E_b/N_0 , en este diagrama al inicio se ingresan los datos necesarios para la simulación que se explican en la Tabla 2.2, se tiene dos bucles, el primero hace referencia al valor de SNR , en donde la señal tendrá diferente comportamiento de acuerdo al valor del SNR aplicado, y el segundo se refiere el número de veces que se repite el código para cada valor de SNR , definido por la variable *Ntrials*, que se conoce como simulaciones Montecarlo y se realiza debido a que se trabaja con variables aleatorias.

Para determinar la BER, se debe comparar el vector de los datos transmitidos *tx_bit_s* con el vector de los datos a la salida del receptor *rx_bit_s* y así determinar el número de errores, para posteriormente calcular la BER.

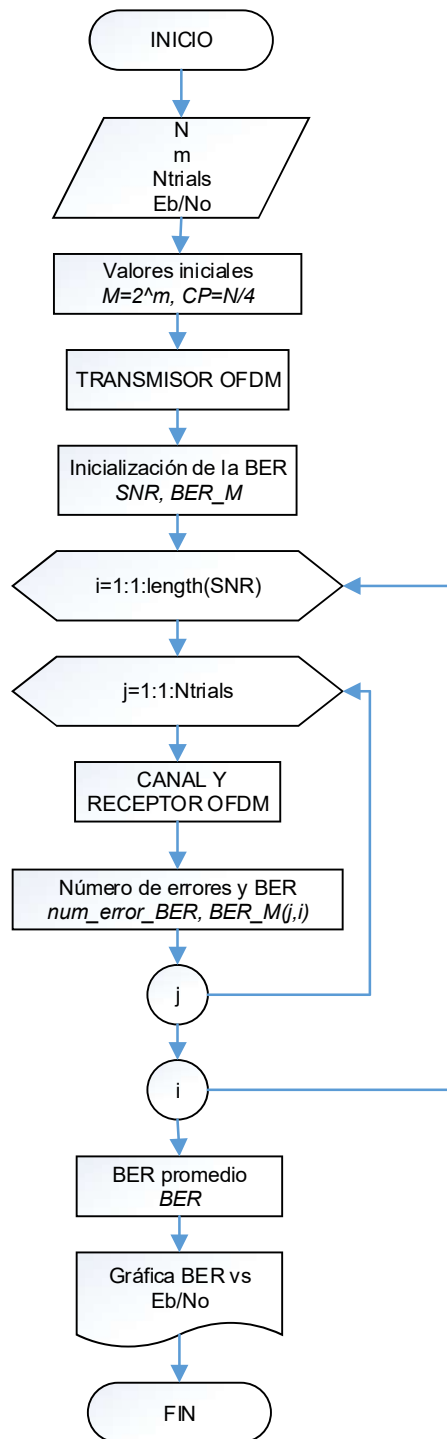


Figura 2.5 Diagrama de flujo para obtener resultados de BER vs Eb/No en OFDM

Con el comando *biterr* se determina el número de errores al comparar cada elemento de los dos vectores mencionados, generando el número de valores diferentes que serían el número de bits errados entre los dos vectores [47], los cuales se almacenan en la variable *num_error_BER*. Para calcular la BER se divide el número de bits errados para el número total de bits enviados en el transmisor, este cálculo se almacena en la variable

BER_M , la cual contiene una matriz de tamaño $(N_{trials}, length(SNR))$, es decir que para cada valor de SNR se calcula N_{trials} veces el valor de la BER.

Luego se realiza el promedio de cada una de las columnas de la matriz BER_M para obtener un vector denominado BER de tamaño de la longitud del vector SNR o del vector $EbNo$. Finalmente se utiliza el comando *semilogy* para obtener la curva BER vs Eb/No. La implementación en MATLAB para estos resultados se encuentra en el ANEXO A.

2.1.4.2. Determinación de la PSD en OFDM

Para poder observar el espectro de la señal se grafica la PSD en función de la frecuencia, la cual se obtiene a la salida del transmisor. La Figura 2.6 muestra el diagrama de flujo, indicando los procesos para obtener dicha gráfica.

En primer lugar, se ingresan los datos necesarios en la simulación que se explican en la Tabla 2.2, luego se crea un vector vacío con el nombre *total_simb*, el cual ayuda a la inicialización de los símbolos OFDM totales que se obtendrán al final del programa. Se tiene un bucle que ayuda a la formación de varios símbolos OFDM dependiendo de la variable *total*. Dentro del bucle consta el transmisor OFDM que involucran todos los procesos explicados anteriormente.

Para graficar la PSD vs frecuencia se utiliza el comando *pwelch*, en donde se especifica la señal de entrada que son los símbolos OFDM totales *total_simb*, el valor de la ventana de Hamming, el número de muestras superpuestas, si se especifica en vacío [] significa una superposición de muestras del 50%, el tamaño de las frecuencias normalizadas W y la frecuencia de muestreo fs que es el número de muestras por unidad de tiempo [48]. El comando *pwelch* devuelve dos vectores P_{xx} y W , de la estimación de PSD y de las frecuencias normalizadas, respectivamente, contienen elementos reales no negativos [47].

La implementación en MATLAB para los resultados de PSD se encuentra en el ANEXO B.

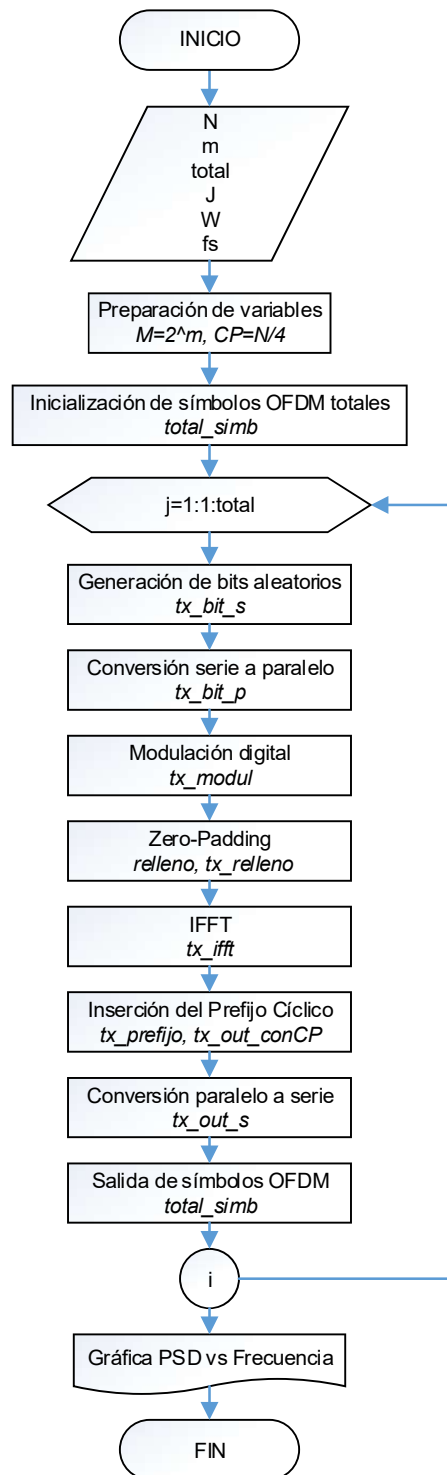


Figura 2.6. Diagrama de flujo para obtener resultados de PSD en OFDM

2.2. SISTEMA SEFDM

El sistema SEFDM está compuesto por el transmisor y receptor. Para comprender la implementación de este es necesario definir las variables involucradas, las cuales se detallan en la Tabla 2.3.

Tabla 2.3. Definición de las variables de ingreso para SEFDM

Variable	Definición
α	Factor de compresión de ancho de banda
N	Número de subportadoras
m	Número de bits agrupados según tipo de modulación: 2:4-QAM, 4:16-QAM, 6:64-QAM
v	Número de iteraciones del detector ID
N_{trials}	Número de iteraciones
E_b/N_0	Valor de E_b/N_0

2.2.1. TRANSMISOR SEFDM

Al transmisor se ingresan datos aleatorios, los cuales continúan con los diferentes procesos para entregar a la salida un símbolo SEFDM. En la Figura 2.7 se detallan los bloques involucrados en el transmisor y en la Figura 2.8 se muestra el diagrama de flujo que involucra a los diferentes procesos.



Figura 2.7. Diagrama de bloques del transmisor SEFDM

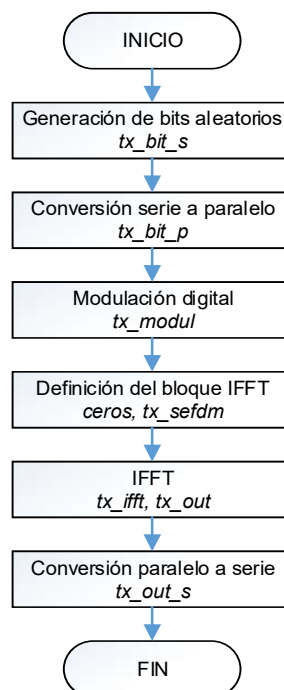


Figura 2.8. Diagrama de flujo del transmisor SEFDM

Los procesos de generación de bits aleatorios, conversión serie a paralelo, modulación digital son iguales a los explicados en el transmisor OFDM. En donde se tendrán vectores almacenados en las siguientes variables: tx_bit_s , tx_bit_p , tx_modul . En el proceso de generación de bits aleatorios antes de generar estos bits se utiliza el comando *rng default*, el cual ayuda a controlar la salida de números aleatorios, reestableciéndolos a sus valores predeterminados, de esta manera se producen los mismos números aleatorios como si reiniciara MATLAB [49], [50].

a) Definición del bloque IFFT

Luego de los anteriores procesos mencionados hay que definir el tamaño del bloque IFFT antes de realizar la operación *ifft*, el tamaño del bloque IFFT $IFFT_FFT_size$, se representa en la Ecuación (2.1), esta variable se define con la división de N para el valor de α , el valor resultante de esta división se redondea al entero más cercano mediante el comando *round*.

$$IFFT_FFT_size = \frac{N}{\alpha} \quad (2.1)$$

El objetivo es tener un bloque de tamaño $\frac{N}{\alpha}$, para esto se deben insertar una cantidad de ceros en función de la Ecuación (2.2), resolviendo esto, la cantidad de ceros también puede seguir la Ecuación (2.3), en donde $\frac{N}{\alpha}$ ya se definió anteriormente mediante la variable $IFFT_FFT_size$, este nuevo vector se guardará en la variable *ceros*.

$$ceros = \frac{1-\alpha}{\alpha} * N \quad (2.2)$$

$$ceros = \frac{N}{\alpha} - N \quad (2.3)$$

Como siguiente paso se crea otro vector con el nombre de tx_sefdm en donde se juntan los valores de los vectores tx_modul y *ceros*.

b) IFFT

Para que las muestras moduladas pasen del dominio de la frecuencia al dominio del tiempo se utiliza el comando *ifft*, se especifica el vector tx_sefdm y el tamaño de dicho vector, es decir el tamaño del vector $IFFT_FFT_size$. La variable tx_ifft almacena el

símbolo SEFDM generado. Luego de esto se crea un vector con el nombre de tx_out en donde se almacenarán los valores de tx_ifft desde el inicio hasta la posición N , es decir se ignoran las últimas muestras.

Y para finalizar con el transmisor se realiza el proceso de conversión paralelo a serie como ya se explicó anteriormente, obteniendo así el vector tx_out_s .

2.2.2. RECEPTOR SEFDM

Al símbolo que sale del transmisor se le añade el ruido AWGN en el canal, luego esta señal ingresa al receptor, en donde se realiza los procesos inversos a los detallados en el transmisor. En el receptor se adiciona un bloque, el cual involucra al detector ID. Para el desarrollo de éste se llama a la función ID. En la Figura 2.9 se detallan cada uno de los bloques implicados en el receptor y en la Figura 2.10 muestra el diagrama de flujo que involucra los procesos para el receptor SEFDM.

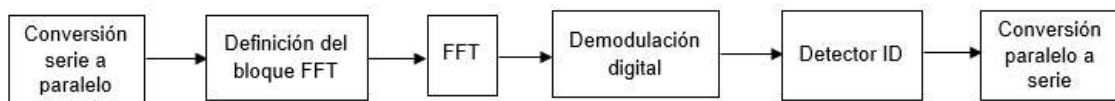


Figura 2.9. Diagrama de bloques del receptor SEFDM

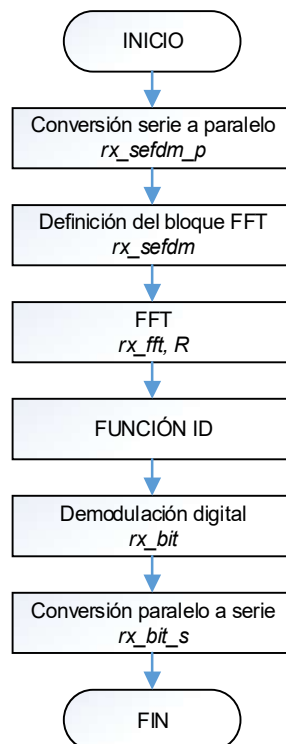


Figura 2.10. Diagrama de flujo del receptor SEFDM

Los procesos de canal AWGN, conversión serie a paralelo son iguales a los explicados en el receptor OFDM, las variables que se obtienen de estos procesos son: *canal*, *rx_sefdm_p*.

a) Definición del bloque FFT

Antes de realizar la operación *fft*, se define el tamaño del bloque FFT siendo igual al tamaño de bloque IFFT, por lo tanto, se define la cantidad de ceros en función de la Ecuación (2.3), y así se utiliza la misma variable *ceros* creada en el transmisor. Posterior a esto se crea otro vector con el nombre *rx_sefdm* para unir los valores de los vectores *rx_sefdm_p* y *ceros*.

b) FFT

Para pasar al vector *rx_sefdm* del dominio del tiempo al dominio de la frecuencia se aplica el comando *fft*, esto se almacena en un vector con el nombre *rx_fft* de tamaño *IFFT_FFT_size*. Luego de esto se crea un vector con el nombre de *R* en donde se almacenarán los valores de *rx_fft* desde el inicio hasta la posición *N*, es decir se ignoran las últimas muestras.

c) Detector ID

El detector iterativo ayuda a minimizar la ICI presente en el sistema SEFDM, por tal razón en el receptor se llama a la función ID, sus parámetros de entrada y salida se explican en la Tabla 2.4. La Figura 2.11 muestra el diagrama de flujo de dicha función ID, en donde se observa que dependiendo del valor de *m* que se inserte sea 2, 4 o 6 se realizará el algoritmo iterativo.

Tabla 2.4. Parámetros de la función ID

Tipo de Parámetro	Variables
Entrada	$R, v, parte_ID, m$
Salida	S_v

Correlación $N \times N$

Previamente, es necesario calcular la matriz de correlación cruzada C que hace referencia a la interferencia entre subportadoras. Para obtener C primero hay que determinar la matriz de subportadoras F y su conjugada F_c ya que la multiplicación de estas dos matrices dará como resultado C . Para la generación de los elementos de la matriz F intervienen las variables: α y N , que se puede observar en la Ecuación (2.4).

$$F_{k,n} = \frac{1}{\sqrt{N}} e^{\frac{i2\pi\alpha nk}{N}} \quad k, n = 0, 1, \dots, N - 1 \quad (2.4)$$

Después se calcula la matriz identidad con el comando `eye`, que necesita el valor de N para su ejecución [51], esta matriz se almacena en la variable I . Ahora I y C se utilizan para calcular una parte del algoritmo iterativo y se almacena en la variable $parte_ID$, consiste en la resta de $I - C$.

En la Figura 2.11, para todos los casos el cálculo de iteración es el mismo, se define la variable $Sv_a = R$, en donde Sv_a es el vector de símbolos estimados después de $v-1$ iteraciones y R es el vector a la salida del bloque FFT, luego se tiene un bucle que da inicio en mm que es la actual iteración y termina en v que es el número de iteración total, mencionados valores son los que involucran al intervalo umbral que se describe en la variable d , definido en la Ecuación (1.16). Después se calcula el valor de Sv que es el vector de símbolos recuperados luego de v iteraciones, esto se explica en la Ecuación (2.5).

$$Sv = R + parte_ID * Sv_a \quad (2.5)$$

La implementación en MATLAB del detector ID se encuentra en el ANEXO C.

Posteriormente, el valor de d y Sv son necesarios para realizar el proceso de soft mapping. Del vector Sv se obtiene los vectores Svr y Svi que son la parte real e imaginaria de Sv . El objetivo de soft mapping es estimar luego de cada iteración al vector Sv_a con sus correspondientes puntos de constelación dependiendo del tipo de modulación.

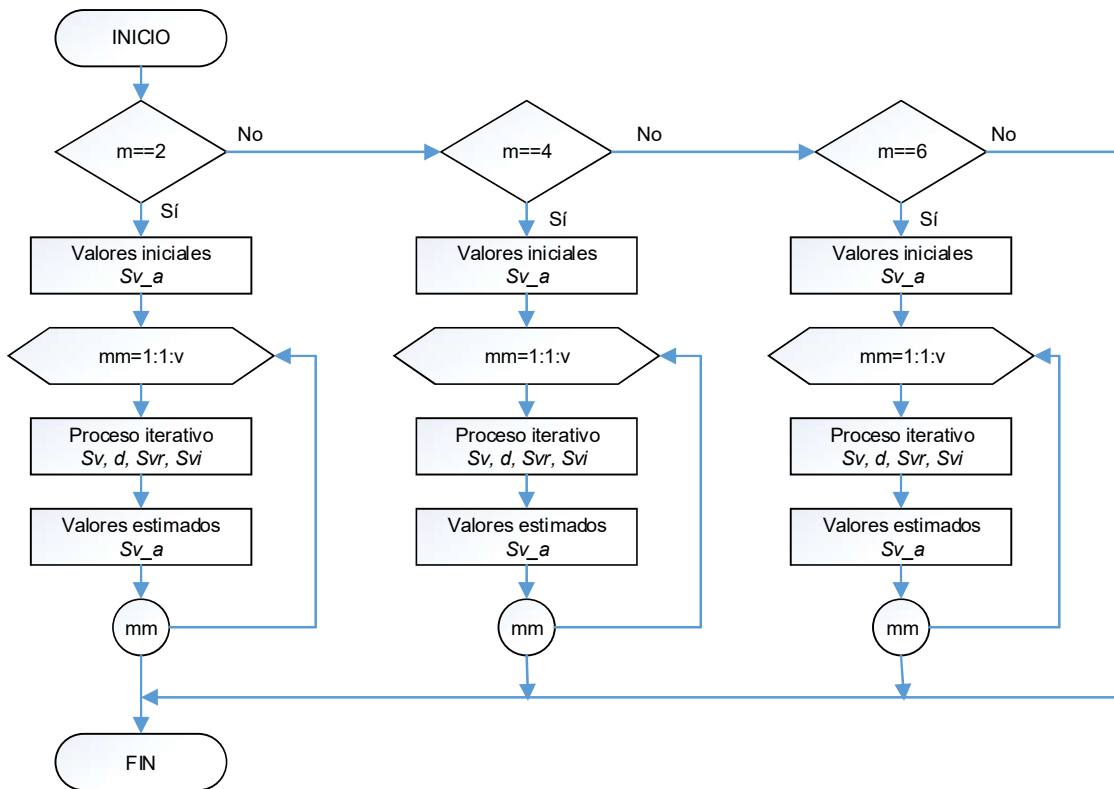


Figura 2.11. Diagrama de flujo de la función ID

A continuación, se explica la estimación del vector Sv_a según el tipo de modulación, para esto se comparan los valores de Svr y Svi con d .

4-QAM: En la Figura 2.12 se observan las zonas A y B de color naranja y celeste respectivamente. Cada punto en esta constelación tiene dos puntos adyacentes que influyen en la decisión de mapear los símbolos a su respectivo punto de constelación, el intervalo umbral ayuda a definir la zona A delimitada por un borde horizontal y un vertical.

La zona A es menos susceptible a la interferencia que la zona B, por lo tanto los puntos localizados dentro de esta zona serán cambiados o mapeados a sus correspondientes valores de la constelación, mientras que los puntos que se encuentran dentro de la zona B no se cambian y se dejan para la siguiente iteración de acuerdo al intervalo de incertidumbre, de esta manera la ICI se irá minimizando conforme dicho intervalo se reduce de forma gradual hasta el valor de cero [24].

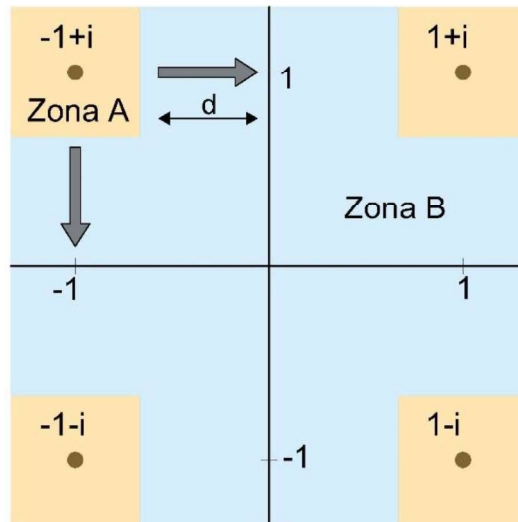


Figura 2.12. Mapeo suave para constelación 4-QAM

16-QAM: Los puntos en la constelación 16-QAM transportan más bits que 4-QAM, lo que hace más difícil el mapeo de símbolos a su respectiva constelación. Por lo tanto, la constelación consta de las áreas 1, 2 y 3 como se observa en la Figura 2.13.

El área 1 de color verde indica al área menos susceptible a la interferencia en comparación con las áreas 2 y 3 de color amarillo y lila respectivamente, ya que los puntos que se encuentren en esta área solo tienen dos puntos adyacentes como se observa en la figura, los cuales influyen al momento de tomar una decisión, de esta manera el intervalo umbral interviene para definir una zona de decisión denominada zona A que está limitado por un borde vertical y un horizontal, fuera de esta zona se encuentra la zona B.

Los puntos localizados en el área 2 tienen tres puntos adyacentes por lo que son más sensibles a la interferencia que los que se encuentran en el área 1 y menos susceptibles con los puntos en el área 3, aquí el intervalo umbral ayuda a definir la zona A y B, en donde la zona A se limita mediante tres bordes: dos horizontales y un vertical.

Finalmente, los puntos localizados en el área 3 tienen cuatro puntos adyacentes por lo que los puntos son más susceptibles a la interferencia en comparación con los puntos en las anteriores áreas, aquí el intervalo umbral interviene para definir cuatro bordes: dos horizontales y dos verticales y de esta manera definir la zona de decisión A y fuera de ésta la zona B.

Las zonas A y B en todas las áreas siguen el mismo esquema de 4-QAM en donde los puntos localizados dentro de la zona A serán cambiados o mapeados a sus correspondientes valores de la constelación, mientras que los puntos que se encuentran dentro de la zona B no se cambian y se dejan para la siguiente iteración de acuerdo al intervalo de incertidumbre hasta que el valor de intervalo de umbral sea 0 y así poder minimizar la ICI [24].

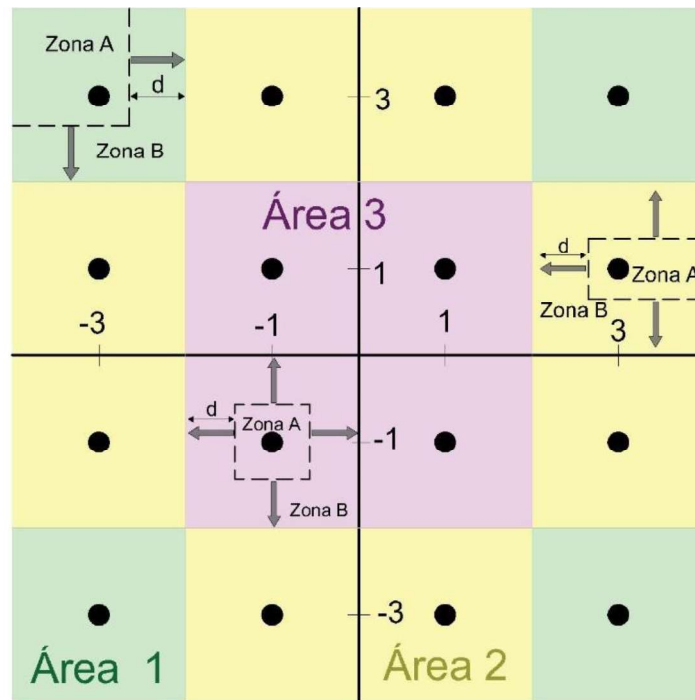


Figura 2.13. Mapeo suave para constelación 16-QAM

64-QAM: En la Figura 2.14 también se observan las áreas 1, 2 y 3 de color verde, amarillo y lila respectivamente como en 16-QAM, la diferencia es que aquí se tendrán que analizar más puntos dentro de cada área. El incremento del número de puntos en la constelación hace que este tipo de modulación sea muy susceptible a la interferencia, teniendo en cuenta que en mencionados puntos se transportan más bits que en 4-QAM y 16-QAM, lo que hace más difícil el mapeo de símbolos a su respectiva constelación.

Se sigue el mismo procedimiento ya explicado en 16-QAM teniendo en cuenta que hay que analizar más puntos y las nuevas posiciones de estos puntos para este tipo de modulación.

Los procesos que faltan para culminar en el receptor: demodulación digital, conversión paralelo a serie son iguales a los explicados en el receptor OFDM, las variables que se obtienen de estos procesos son: rx_bit y rx_bit_s .

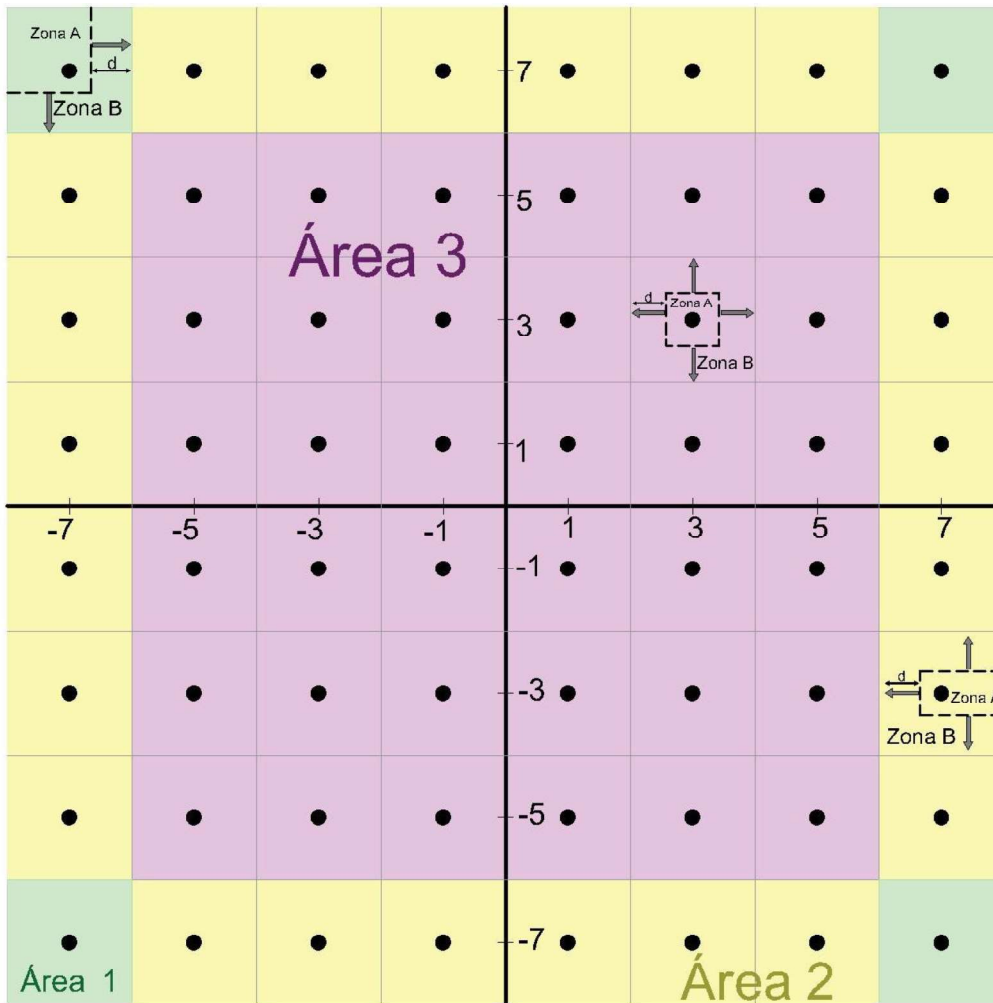


Figura 2.14. Mapeo suave para constelación 64-QAM

2.2.3. RESULTADOS EN SEFDM

Una vez que se ha implementado el transmisor y receptor se procede a evaluar los resultados en términos de BER vs E_b/N_0 y de PSD.

2.2.3.1. Cálculo de errores y BER en SEFDM

La Figura 2.15 muestra el diagrama de flujo para obtener resultados de BER vs E_b/N_0 para el sistema SEFDM implementado en MATLAB, el cual se encuentra en el ANEXO D, primero se ingresan los datos necesarios para la simulación que se detalló en la Tabla 2.3, luego se llama a la función SEFDM, sus parámetros de entrada y salida se explican en la Tabla 2.5, posteriormente se obtiene la gráfica de BER vs E_b/N_0 mediante el comando *semilogy*.

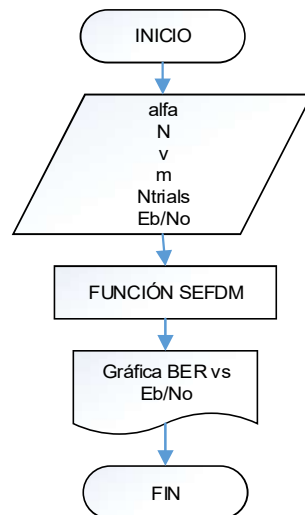


Figura 2.15. Diagrama de flujo para obtener resultados de BER vs Eb/No en SEFDM

Tabla 2.5. Parámetros de la función SEFDM

Tipo de Parámetro	Variables
Entrada	<i>alfa, N, m, v Ntrials</i>
Salida	<i>BER</i>

La Figura 2.16 muestra el diagrama de flujo para la función SEFDM, la implementación se encuentra en el ANEXO E, al inicio se calculan los valores iniciales como el orden de modulación M , el tamaño del bloque IFFT $IFFT_FFT_size$. En esta figura se tiene dos bucles, uno para cada valor de SNR y el otro para las $Ntrials$ veces que se repetirá el código para cada valor de SNR .

Los bits recibidos en la variable rx_bit_s se comparan con los bits transmitidos que se encuentran en la variable tx_bit_s para determinar el número de errores entre los dos vectores, cuyo valor posteriormente será necesario para determinar la BER. Los procedimientos para calcular los errores y la BER ya se detallaron en el sistema OFDM. El objetivo es obtener un vector denominado BER de tamaño de la longitud del vector SNR o del vector $EbNo$.

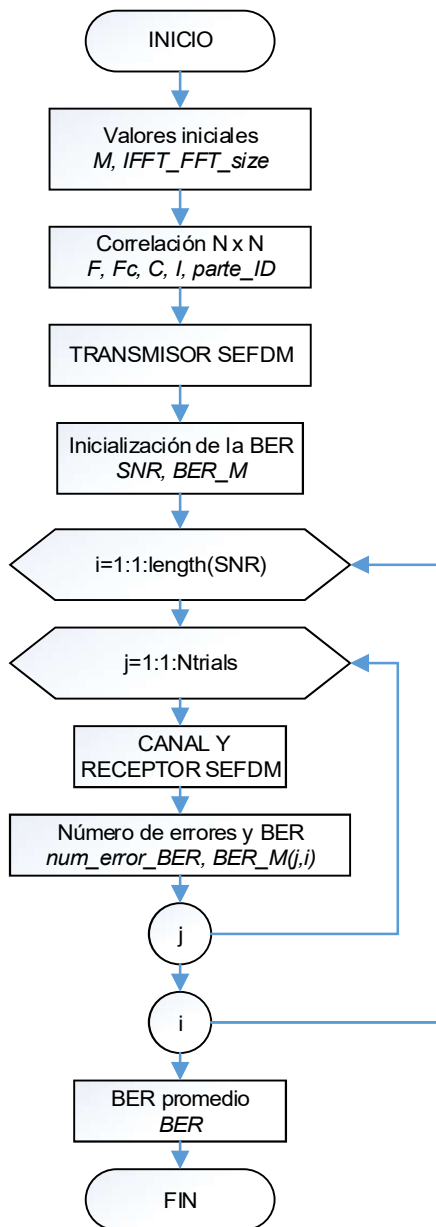


Figura 2.16. Diagrama de flujo de la función SEFDM

2.2.3.2. Determinación de la PSD en SEFDM

La Figura 2.17 muestra el diagrama de flujo con sus diferentes procesos para poder determinar la gráfica PSD vs frecuencia en el sistema SEFDM. Primero se ingresan los datos necesarios en la simulación, los cuales se explican en la Tabla 2.6, posteriormente se llama a la función SEFDM_tx para posteriormente obtener la PSD, el detalle de este código se encuentra en el ANEXO F.

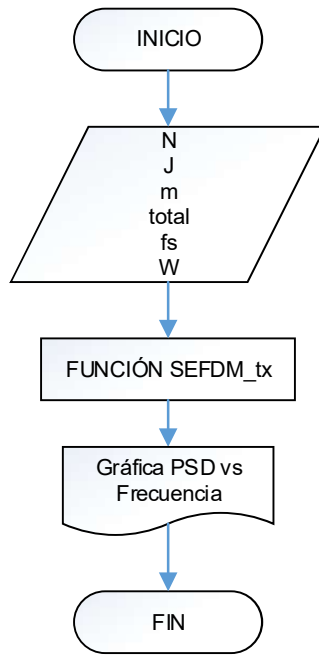


Figura 2.17. Diagrama de flujo para obtener resultados de PSD vs frecuencia en SEFDM

Tabla 2.6. Definición de las variables de ingreso para obtener PSD en SEFDM

Variable	Definición
N	Número de subportadoras
m	Número de bits agrupados según tipo de modulación: 2:4-QAM, 4:16-QAM, 6:64-QAM
$total$	Total de símbolos SEFDM
J	Factor de sobremuestreo
fs	Frecuencia de muestreo
W	Tamaño de las frecuencias normalizadas

En la Tabla 2.7 se detalla los parámetros de entrada y salida para la función SEFDM_tx y en la Figura 2.18 se muestra el diagrama de flujo para dicha función.

Tabla 2.7. Parámetros de la función SEFDM_tx

Tipo de Parámetro	Variables
Entrada	$alfa, N, m, J, total$
Salida	$total_simb$

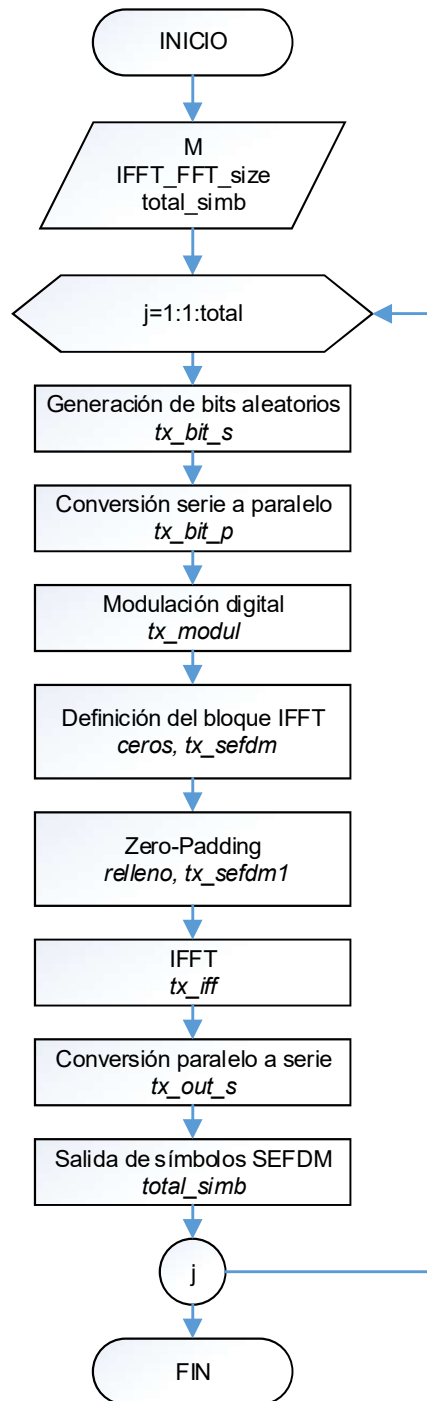


Figura 2.18. Diagrama de flujo de la función SEFDM_tx

En la función SEFDM_tx se definen los valores iniciales para luego ingresar a un bucle, el cual se utiliza para la generación de varios símbolos SEFDM. Antes de ingresar al bucle se crea un vector vacío con el nombre *total_simb*, el cual ayuda a la inicialización de los símbolos SEFDM totales que se obtendrán al final del programa. Dentro de este bucle se tiene al transmisor SEFDM que involucran todos los procesos explicados anteriormente, dentro del bloque IFFT se inserta un procedimiento antes de efectuar la

IFFT, que consiste en intercambiar las mitades izquierda y derecha del vector *tx_sefdm* para una mejor visualización del espectro, este vector se almacena en la variable *tx_shift*. Para graficar la PSD vs frecuencia se utiliza el comando *pwelch*, que ya fue detallado anteriormente. En el ANEXO G se detalla la función SEFDM_tx.

3. RESULTADOS Y DISCUSIÓN

En el presente capítulo se presentarán los resultados obtenidos de las simulaciones mediante curvas de BER vs E_b/N_0 y de PSD tanto para OFDM y SEFDM. Para una mejor visualización se realiza 10000 iteraciones en las diferentes simulaciones.

3.1. ESCENARIO DE SIMULACIÓN

Las simulaciones se realizaron en MATLAB, donde se implementó el transmisor y receptor OFDM y SEFDM, para conectar el transmisor y receptor se utilizó un canal AWGN. En el transmisor de los dos sistemas las variantes de simulación para la formación de símbolos OFDM como de SEFDM son los esquemas de modulación 4-QAM, 16-QAM y 64-QAM. En el transmisor del sistema SEFDM para cada esquema de modulación se tienen diferentes valores de subportadoras $N = \{16, 32, 48, 64, 128\}$, además de estas variantes se introdujo el factor de compresión $alfa = \{0.2, 0.4, 0.6, 0.65, 0.7, 0.8, 0.9, 0.95, 1\}$. Para el receptor SEFDM se ingresa como variante el número de iteraciones del detector ID $v = \{2, 5, 10, 15, 20, 30, 40\}$.

3.1.1. PRUEBAS CON OFDM

El esquema general de comunicaciones que utiliza la técnica de transmisión multiportadora OFDM se simula para obtener las curvas de BER vs E_b/N_0 , con subportadoras $N = 64$, un vector E_b/N_0 con valores de $E_b/N_0 = 0:2:30$ y para diferentes esquemas de modulación que se define con el valor de m . En el transmisor se genera un símbolo OFDM en base a bits aleatorios que ingresan al transmisor, a este símbolo OFDM se le añade ruido AWGN, luego en el receptor se recupera los bits transmitidos. Posteriormente se compara los bits de entrada y salida del sistema para obtener la BER. Se utiliza la variable $N_{trials} = 10000$ para una mejor visualización de las curvas BER vs E_b/N_0 .

Para verificar el correcto funcionamiento del sistema se compara con la curva que proporciona MATLAB de la BER para canales AWGN en un determinado esquema de modulación QAM mediante el comando *berawgn*, en donde hace uso del vector E_b/N_0 y el nivel de modulación M .

A continuación, se presentan las curvas de BER vs E_b/N_0 generada por la simulación del sistema OFDM a la que se denomina BER SIMULADO y la que se obtiene de forma

teórica, la cual permite calcular a través del comando *berawgn*; y se identifica con el nombre de BER TEÓRICO.

En las Figuras 3.1, 3.2 y 3.3 se observan de color amarillo la curva generada a través de las simulaciones por un sistema OFDM y de color rojo con línea entrecortada es la curva obtenida con el comando *berawgn*. Estas dos curvas son realizadas para una modulación 4-QAM, 16-QAM y 64-QAM, respectivamente.

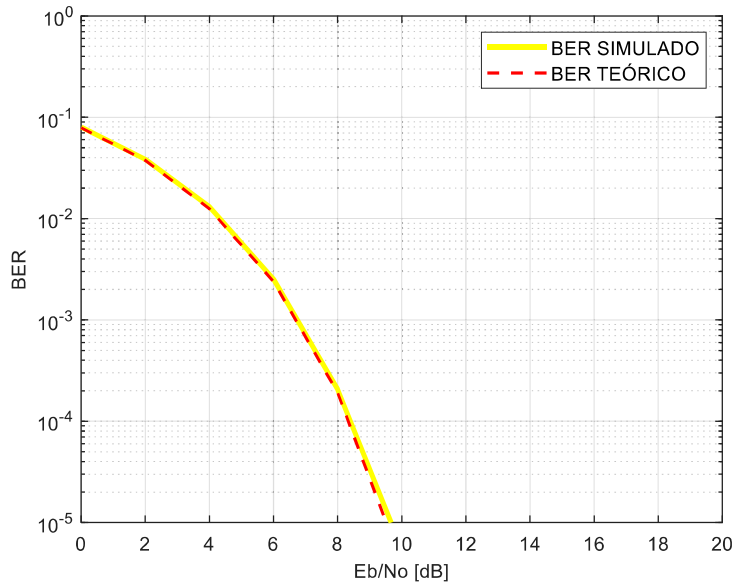


Figura 3.1. BER vs E_b/N_0 de un sistema OFDM con modulación 4-QAM

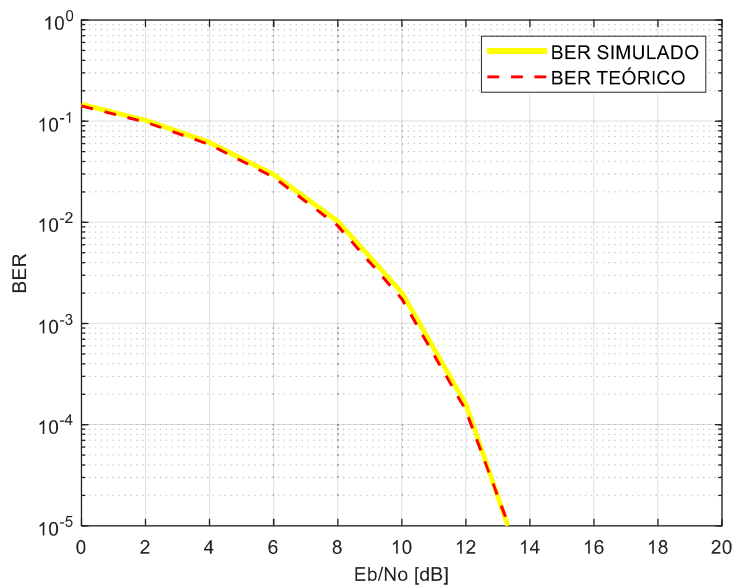


Figura 3.2. BER vs E_b/N_0 de un sistema OFDM con modulación 16-QAM

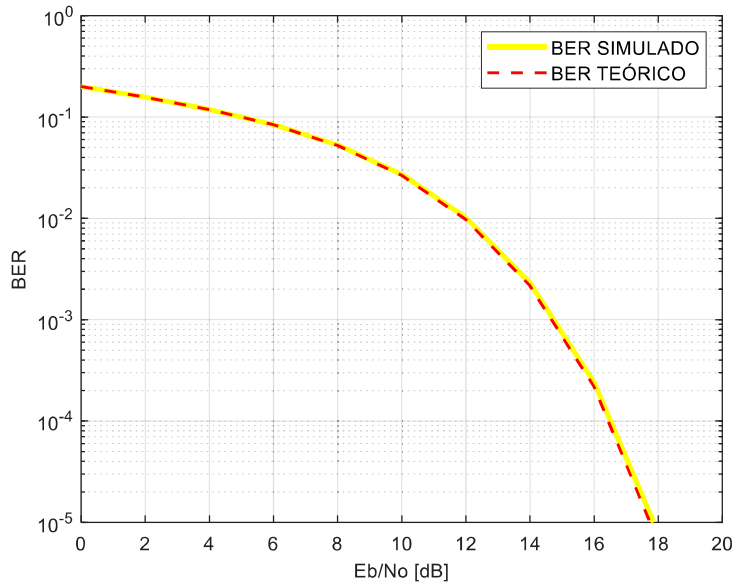


Figura 3.3. BER vs Eb/No de un sistema OFDM con modulación 64-QAM

En todas las figuras se observa que las dos curvas son idénticas, por lo tanto, el funcionamiento del sistema implementado es correcto.

3.1.2. PRUEBAS CON SEFDM

El esquema general de comunicaciones que utiliza la técnica de transmisión multiportadora SEFDM se simula para obtener las curvas de BER vs Eb/No. En el transmisor se genera un símbolo SEFDM en base a bits aleatorios que ingresan al transmisor, a este símbolo se le añade ruido AWGN, luego en el receptor se recupera los bits transmitidos. Posteriormente se compara los bits de entrada y salida del sistema para obtener la BER. Se utiliza la variable $N_{trials} = 10000$ para una mejor visualización de las curvas BER vs Eb/No.

Para evaluar el funcionamiento del sistema SEFDM se compara la curva de BER de SEFDM para un valor $\alpha = 1$ con la de OFDM, puesto que para este factor de compresión se asemeja mucho a OFDM. OFDM puede ser considerado un caso especial de SEFDM cuando $\alpha = 1$ [2]. En las Figuras 3.4, 3.5 y 3.6 se muestran los resultados de BER vs Eb/No para $N = 64$, $\alpha = 1$, $v = 20$, $EbNo = 0:2:30$ y para diferentes esquemas de modulación $m = 2, 4, 6$.

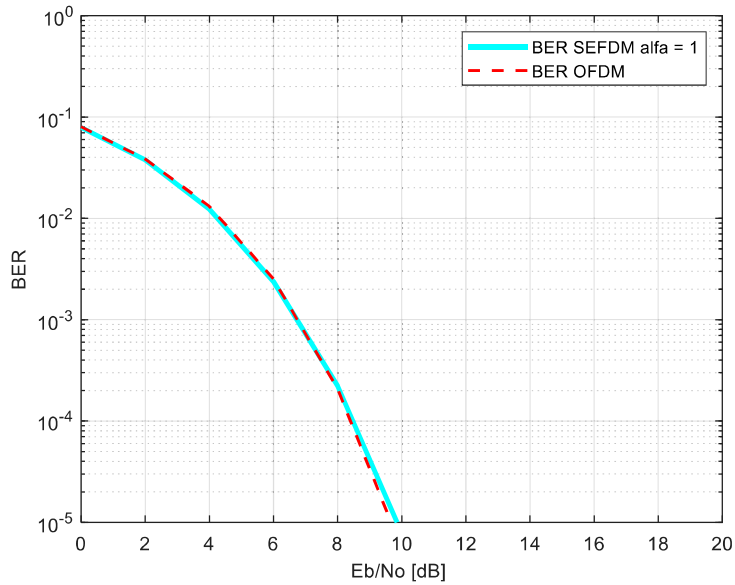


Figura 3.4. BER vs E_b/N_0 de un sistema SEFDM y de OFDM con modulación 4-QAM

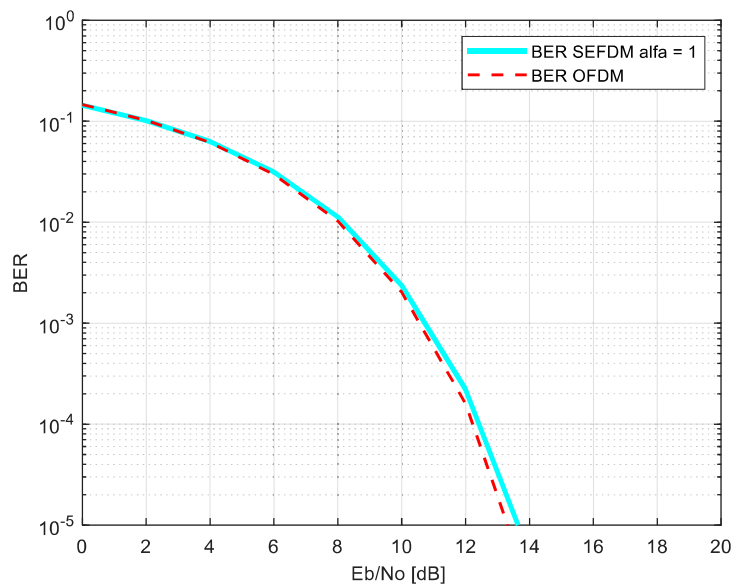


Figura 3.5. BER vs E_b/N_0 de un sistema SEFDM y de OFDM con modulación 16-QAM

En la Figuras 3.4, 3.5 y 3.6 se presentan las curvas de BER vs E_b/N_0 de color cian generada por la simulación del sistema SEFDM para un valor, $\alpha = 1$ y la curva de color rojo con línea entrecortada pertenece al sistema OFDM, en ambos casos para modulaciones 4-QAM, 16-QAM y 64-QAM respectivamente. En todas estas tres figuras se observa que las dos curvas son similares, por lo tanto, el funcionamiento del sistema SEFDM es correcto.

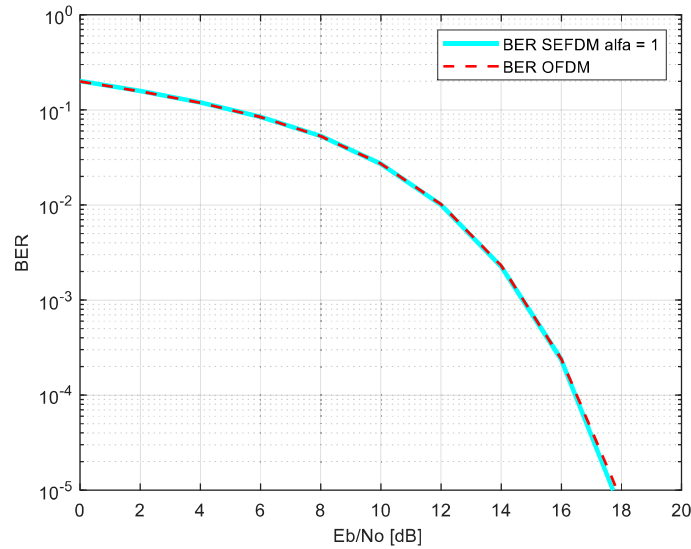


Figura 3.6. BER vs Eb/No de un sistema SEFDM y de OFDM con modulación 64-QAM

3.1.3. SELECCIÓN DEL NÚMERO DE ITERACIONES ν DEL DETECTOR

El detector ID es una parte importante dentro del receptor SEFDM, puesto que ayuda a minimizar la interferencia entre subportadoras, lo que se ve reflejado en un incremento de la BER. El detector entre otras variables necesita del valor de ν , que es el número de iteraciones de dicho detector, dependiendo del número de iteraciones la BER irá variando. Para determinar el valor adecuado de ν se simulará al sistema SEFDM con varios valores de iteraciones en diferentes esquemas de modulación.

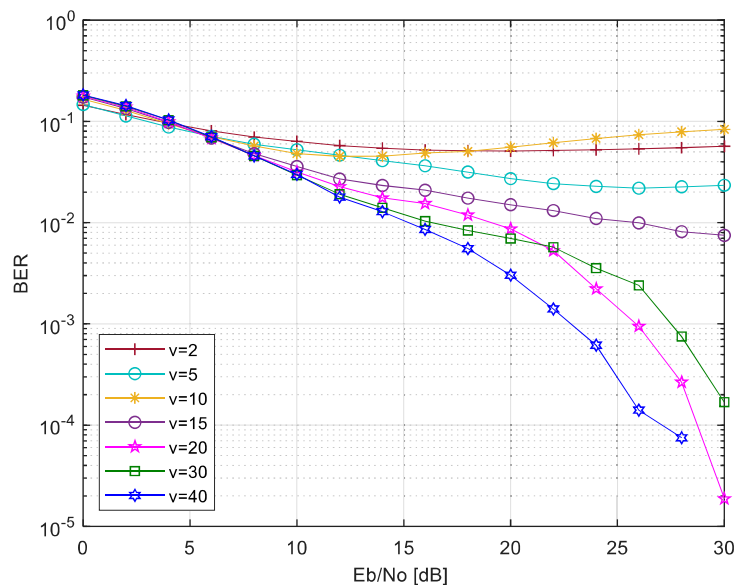


Figura 3.7. BER vs Eb/No con $N = 16$, $alfa = 0.65$, modulación 4-QAM para varias iteraciones del detector ID

En la Figura 3.7 se observan varias curvas con diferentes valores de $v = \{2, 5, 10, 15, 20, 30, 40\}$, $N = 16$, $\alpha = 0.65$, $m = 2$. En la Figura 3.8 los valores de entrada son los mismos que la Figura 3.7 a excepción del valor de $\alpha = 0.6$.

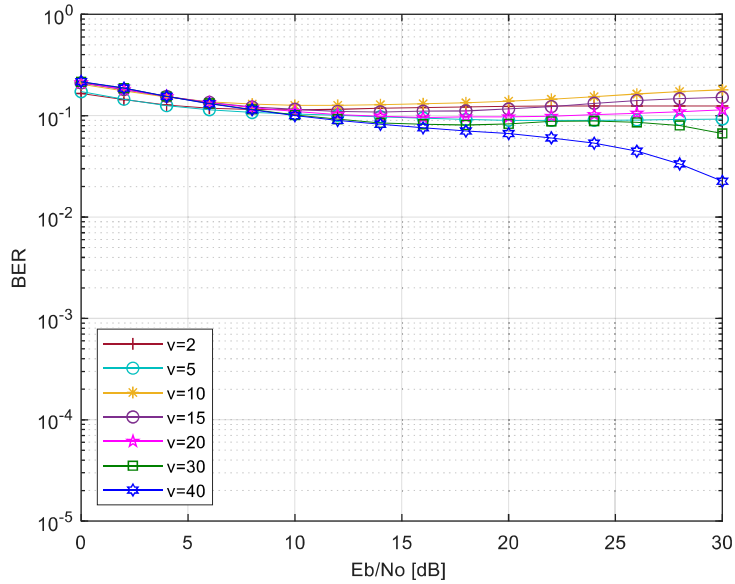


Figura 3.8. BER vs E_b/N_0 con $N = 16$, $\alpha = 0.6$, modulación 4-QAM, para varias iteraciones del detector ID

El número de iteraciones del detector ID v también se analiza para las modulaciones 16-QAM y 64-QAM como se observa en la Figura 3.9 y 3.10 respectivamente.

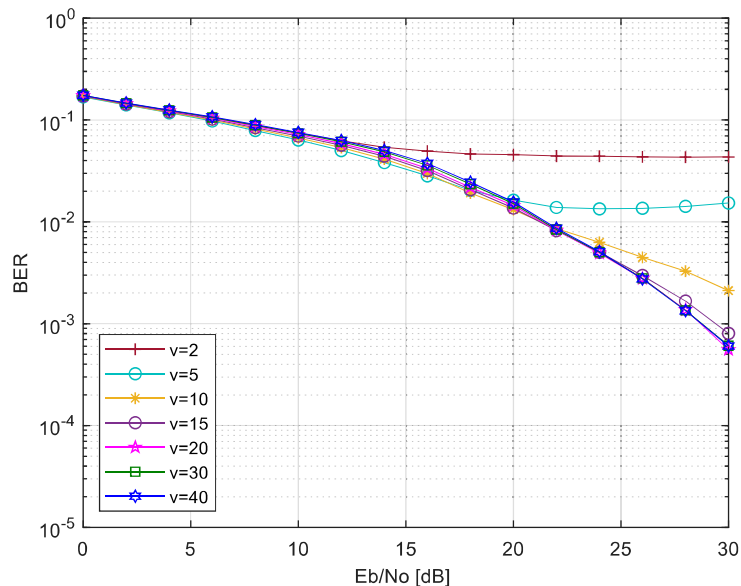


Figura 3.9. BER vs E_b/N_0 con $N = 16$, $\alpha = 0.8$, modulación 16-QAM, para varias iteraciones del detector ID

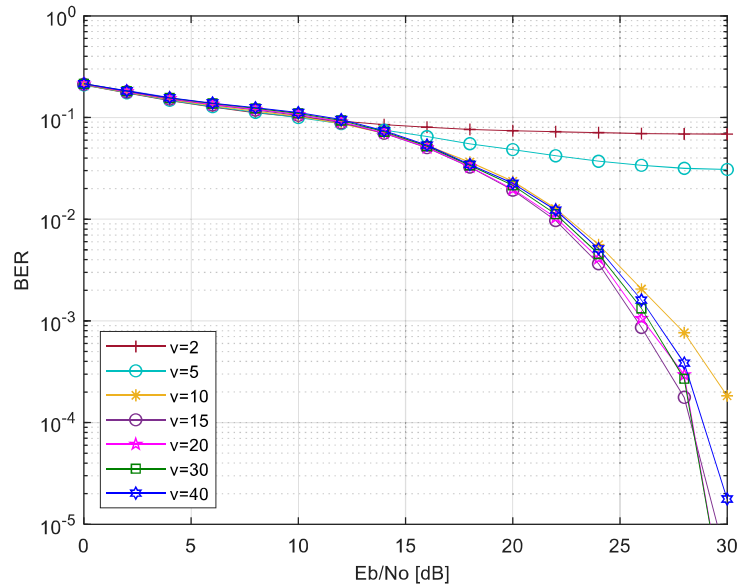


Figura 3.10. BER vs E_b/N_0 con $N = 16$, $\alpha = 0.95$, modulación 64-QAM, para varias iteraciones del detector ID

Como se observó en las Figuras 3.7, 3.8, 3.9 y 3.10 dependiendo del valor v que se utilice las curvas de BER también varían. De manera general, a mayor número de iteraciones del detector ID v , la BER va mejorando. Por lo tanto, las siguientes simulaciones se realizaron con un valor de $v = 40$.

3.2. RESULTADOS DE BER vs E_b/N_0

Se obtienen resultados de BER vs E_b/N_0 para diferentes valores de subportadoras $N = \{16, 32, 48, 64, 128\}$, factor de compresión $\alpha = \{0.2, 0.4, 0.6, 0.65, 0.7, 0.8, 0.9, 0.95, 1\}$, niveles de modulación $m = \{2, 4, 6\}$ con un número de iteraciones del detector ID $v = 40$.

3.2.1. MODULACIÓN 4-QAM

En la Figura 3.11, se obtienen las curvas BER vs E_b/N_0 para diferentes valores de compresión $\alpha = \{0.2, 0.4, 0.6, 0.65, 0.7, 0.8, 0.9, 1\}$ y con valores fijos de $N = 16, v = 40, m = 2$, aquí se observa la degradación de la BER para valores α más pequeños, es decir mientras más se comprime el espectro la BER aumentará significativamente.

La BER también depende del número de subportadoras N , por lo tanto, en las siguientes figuras se analiza este comportamiento para diferentes valores de α que no tengan mucha degradación de BER haciendo referencia a la Figura 3.11.

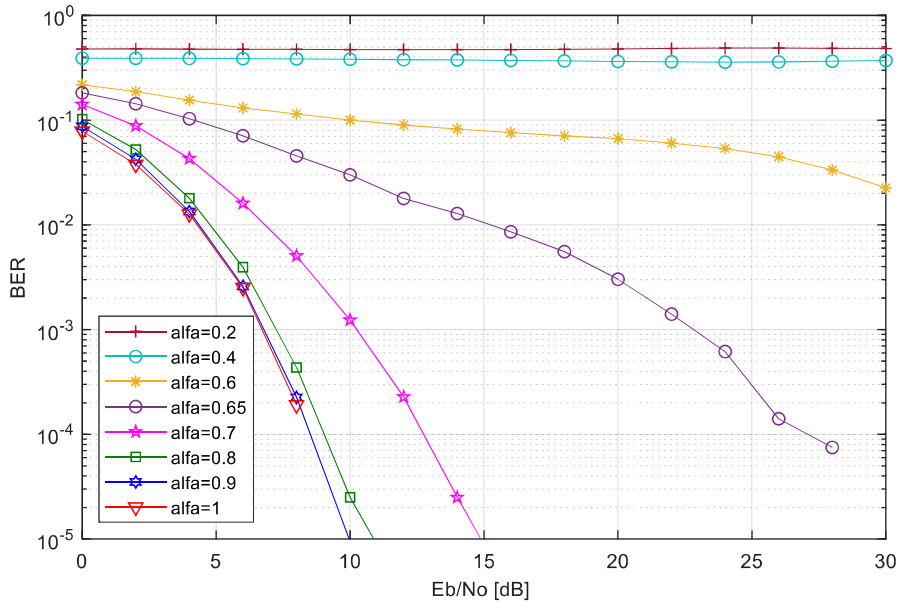


Figura 3.11. BER vs Eb/No con $N = 16$, $v = 40$, $m = 2$ para varios valores de α

A continuación, se presentan las curvas BER vs Eb/No en las Figuras 3.12, 3.13, 3.14 y 3.15 para valores $\alpha = \{0.95, 0.9, 0.8, 0.7\}$ respectivamente, cada una con diferentes valores de subportadoras $N = \{16, 32, 48, 64, 128\}$ y con valores fijos para todos los casos $v = 40$, $m = 2$. En estas figuras se puede apreciar como el valor de N influye al comportamiento de la BER.

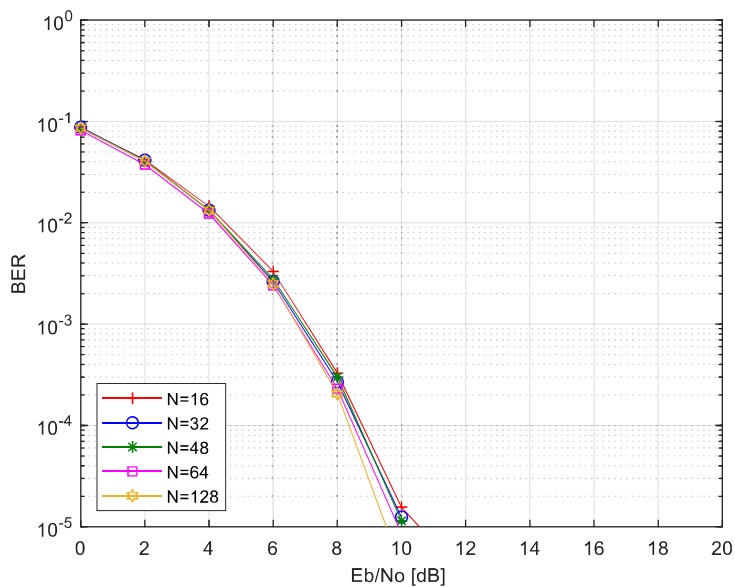


Figura 3.12. BER vs Eb/No con $\alpha = 0.95$, $v = 40$, $m = 2$ para varios valores de N

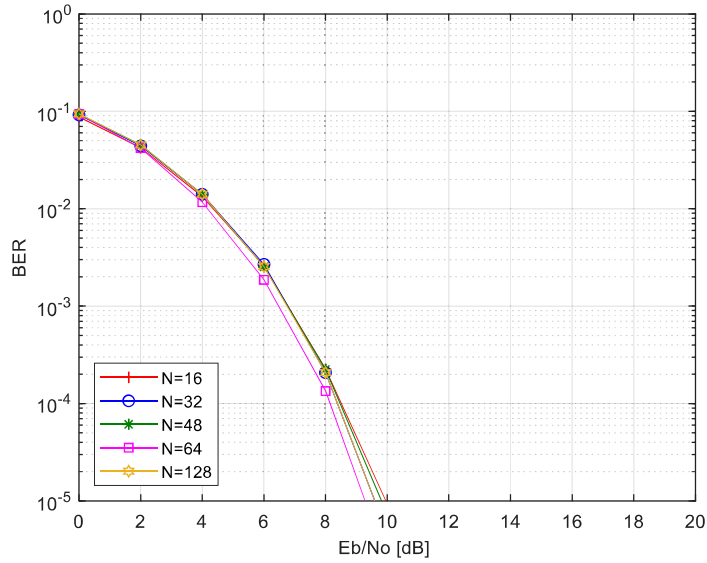


Figura 3.13. BER vs E_b/N_0 con $\alpha = 0.9$, $\nu = 40$, $m = 2$ para varios valores de N

En las Figuras 3.12 y 3.13 para cualquier valor de N dado las curvas de BER para todos los casos varía ligeramente, es decir coinciden los valores de E_b/N_0 para un valor de BER dado.

Mientras que en las Figuras 3.14 y 3.15 se observa que mientras se aplica un factor de compresión más pequeño existe una mayor degradación de las curvas de BER para los diferentes valores de N . Se observa también que existe una distorsión del sistema, el detector ya no funciona bien para valores de $N \geq 64$ en el caso de $\alpha = 0.8$, y para valores de $N \geq 32$ en el caso de $\alpha = 0.7$, por ejemplo en la Figura 3.14 la curva BER con $N = 64$ posee más degradación que la curva BER con $N = 128$.

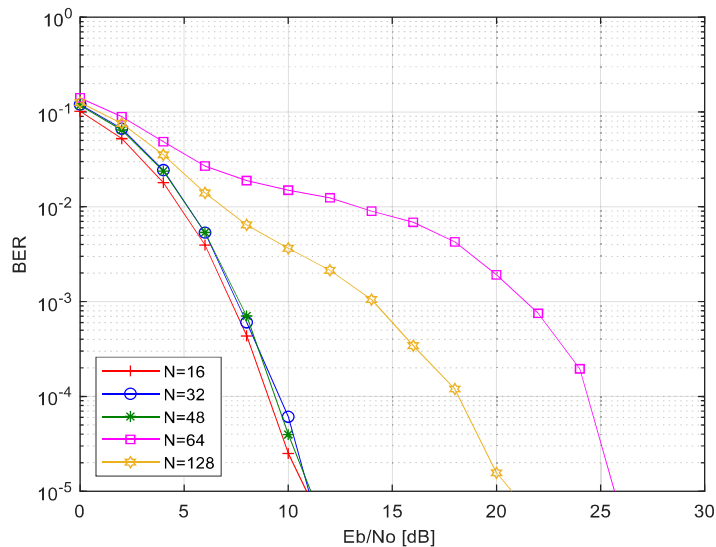


Figura 3.14. BER vs E_b/N_0 con $\alpha = 0.8$, $\nu = 40$, $m = 2$ para varios valores de N

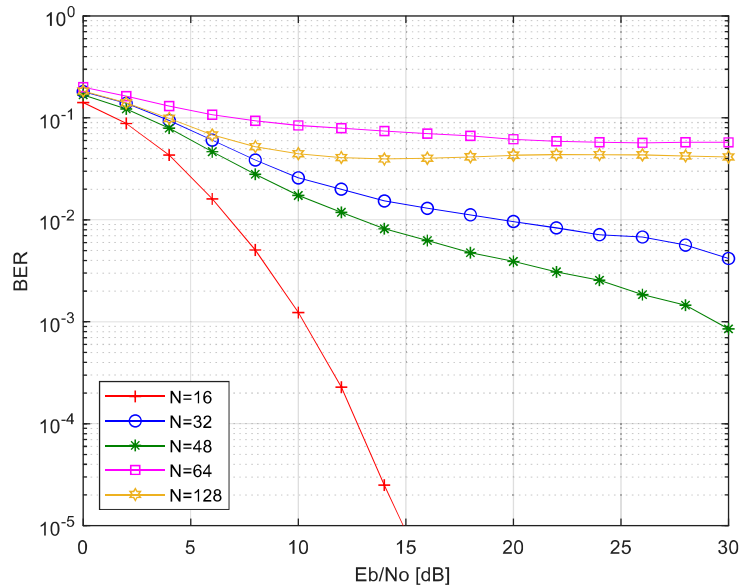


Figura 3.15. BER vs E_b/N_0 con $\alpha = 0.7$, $\nu = 40$, $m = 2$ para varios valores de N

3.2.2. MODULACIÓN 16-QAM

En la Figura 3.16 se obtiene la curva BER vs E_b/N_0 para diferentes valores de compresión $\alpha = \{0.2, 0.4, 0.6, 0.7, 0.8, 0.9, 0.95, 1\}$ y con valores fijos de $N = 16$, $\nu = 40$, $m = 4$, aquí también se observa la degradación de la BER para factores de compresión más pequeños como en el caso de 4-QAM.

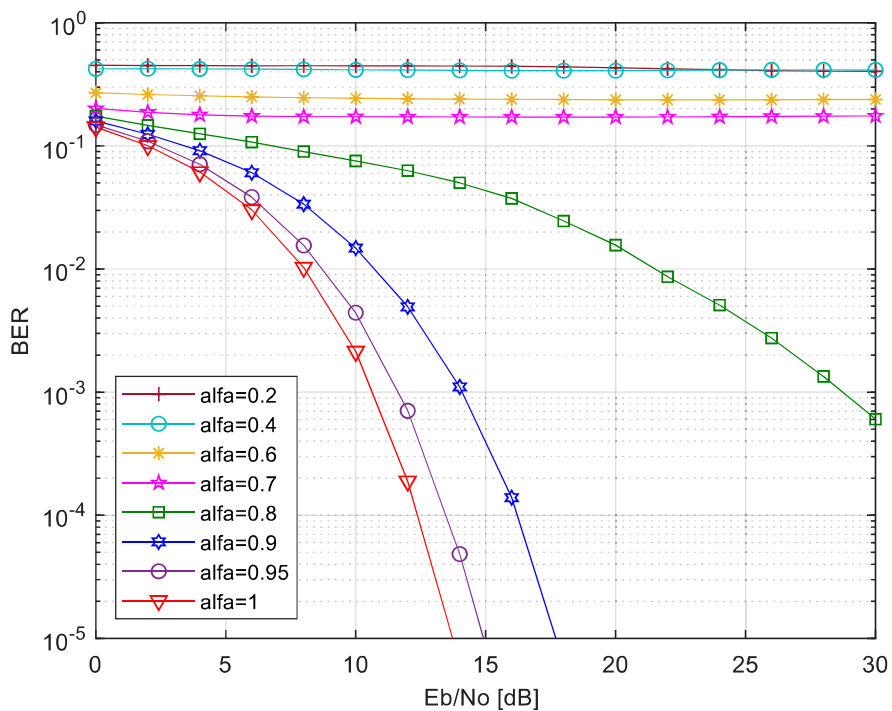


Figura 3.16. BER vs E_b/N_0 con $N = 16$, $\nu = 40$, $m = 4$, para varios valores de α

El número de subportadoras N , influye en el comportamiento de la BER y más aún si se trata de un nivel de modulación 16-QAM, por lo tanto, en las siguientes gráficas se analiza este comportamiento para diferentes valores de α que no tengan mucha degradación de BER haciendo referencia a la Figura 3.16.

A continuación, se presenta las curvas BER vs E_b/N_0 en las Figuras 3.17, 3.18 y 3.19 para valores $\alpha = \{0.95, 0.9, 0.8\}$ respectivamente con diferentes valores de subportadoras $N = \{16, 32, 48, 64, 128\}$ y con valores fijos para todos los casos $\nu = 40$, $m = 4$. En estas figuras se puede apreciar la influencia de N en las curvas BER.

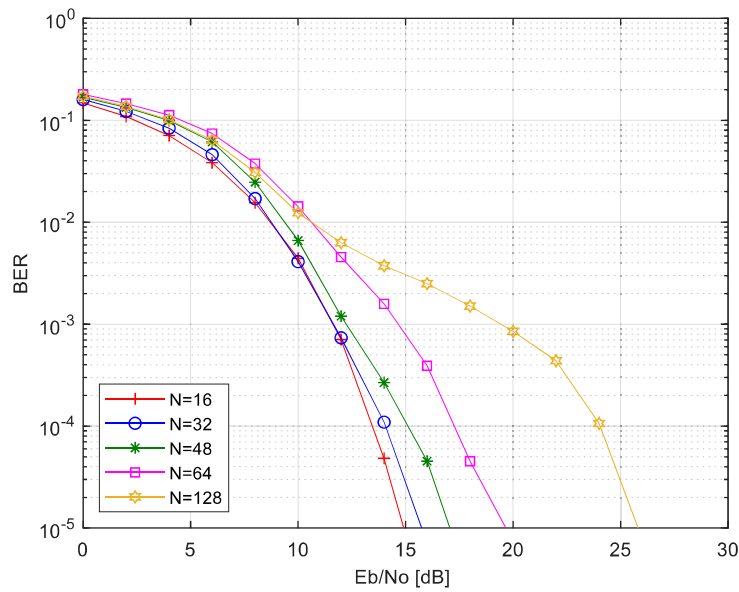


Figura 3.17. BER vs E_b/N_0 con $\alpha = 0.95$, $\nu = 40$, $m = 4$ para varios valores de N

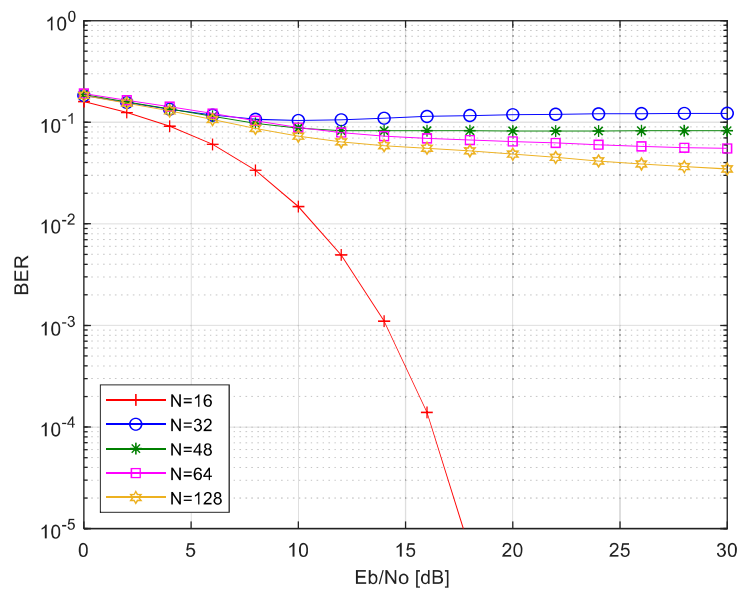


Figura 3.18. BER vs E_b/N_0 con $\alpha = 0.9$, $\nu = 40$, $m = 4$ para varios valores de N

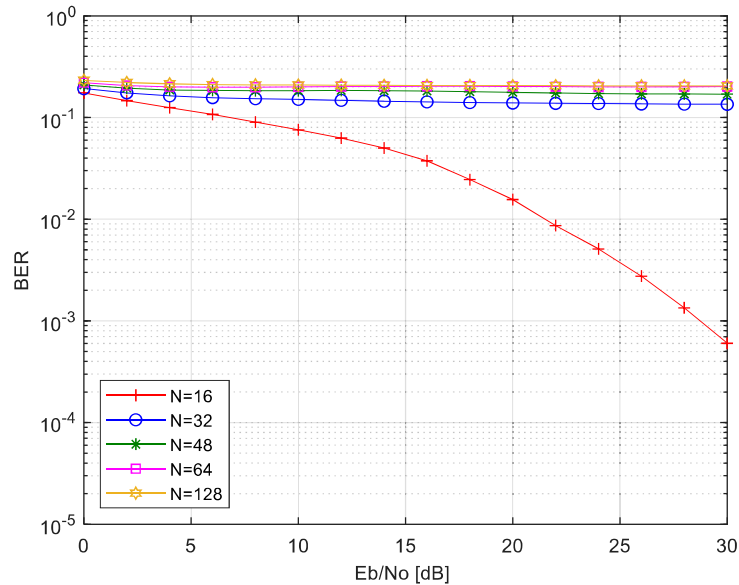


Figura 3.19. BER vs Eb/No con $\alpha = 0.8$, $v = 40$, $m = 4$ para varios valores de N

En las Figura 3.17 para $N = 16$ y $N = 32$ las curvas de BER son iguales, es decir coinciden los valores de Eb/No para un valor de BER dado, pero mientras el valor de N aumenta existe una mayor degradación de la BER. Mientras que en las Figuras 3.18 y 3.19 se observa que ya no hay coincidencia de las curvas para diferentes valores de N . Al aplicar un factor de compresión α más pequeño existe una mayor degradación de las curvas BER para los diferentes valores de N y mayor distorsión del sistema en comparación con 4-QAM, se deduce que ésta afectación se debe a la interferencia que aumenta mientras incrementa el nivel de modulación.

3.2.3. MODULACIÓN 64-QAM

En la Figura 3.20 se obtiene la curva BER vs Eb/No para diferentes valores de compresión $\alpha = \{0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 1\}$ y con valores fijos de $N = 16, v = 40, m = 6$, se observa la degradación de la BER para la mayoría de factores de compresión, esta degradación de BER es mayor que en los esquemas 4-QAM y 16-QAM, puesto que 64-QAM es el más alto nivel de modulación.

En la Figura 3.21 se analiza el comportamiento de la BER con diferentes subportadoras N para el factor de compresión con menos degradación de la BER $\alpha = 0.95$ de acuerdo a la Figura 3.20, se ve la afectación que tiene la BER con diferentes valores de N al estar en un esquema no tan robusto como es 64-QAM el detector ya no funciona bien.

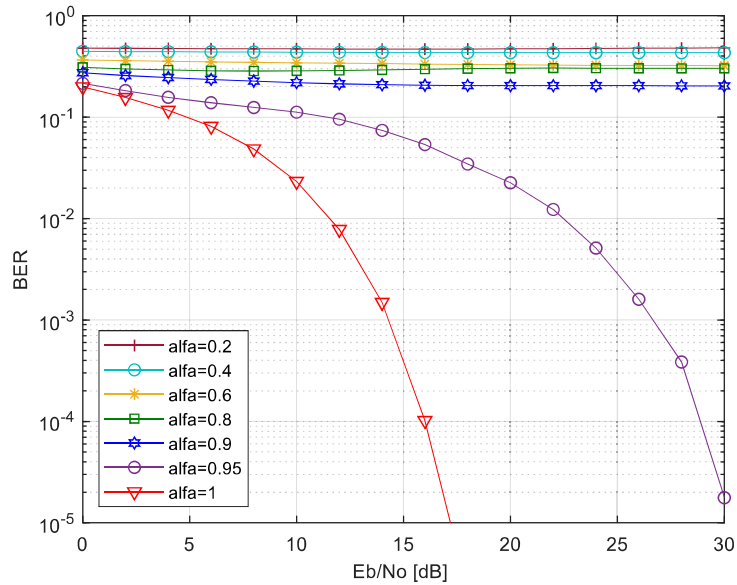


Figura 3.20. BER vs Eb/No con $N = 16$, $v = 40$, $m = 6$ para varios valores de α

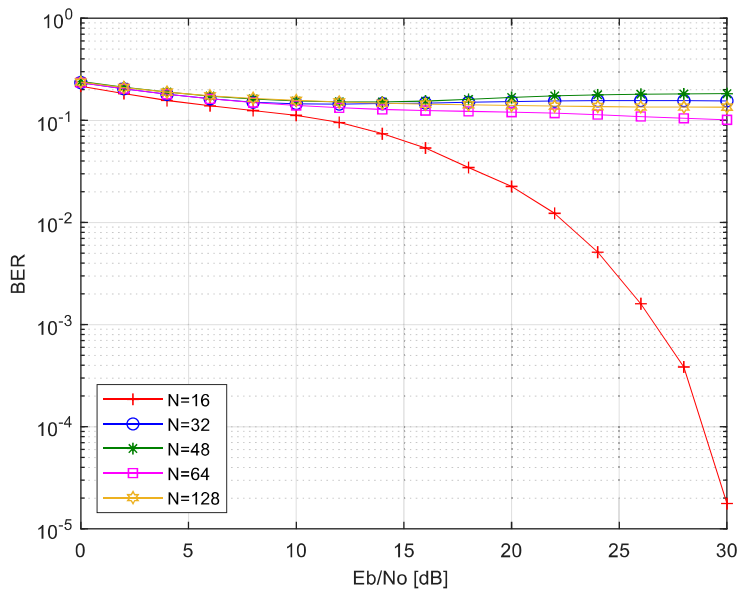


Figura 3.21. BER vs Eb/No con $\alpha = 0.95$, $v = 40$, $m = 6$ para varios valores de N

Por lo tanto, al observar los resultados de BER vs Eb/No para diferentes valores de α y m , se deduce que el detector ID cuenta con una limitación inherente para valores altos de N al aplicar un factor de compresión α cada vez más pequeño y más aun con el incremento del nivel de modulación m . Sin embargo, su presencia dentro del receptor es importante, la cual se analiza en el siguiente punto.

3.2.4. SIMULACIÓN CON Y SIN DETECTOR

En las Figuras 3.22, 3.23 y 3.24 se observa el efecto del detector dentro del sistema SEFDM para las modulaciones: 4-QAM, 16-QAM y 64-QAM respectivamente, el cual ayuda a minimizar la BER. Los valores fijos para estas gráficas son: $N = 16$, $v = 40$. Para la modulación 4-QAM se utiliza un $\alpha = 0.8$, para 16-QAM $\alpha = 0.9$ y para 64-QAM con un valor de $\alpha = 0.95$.

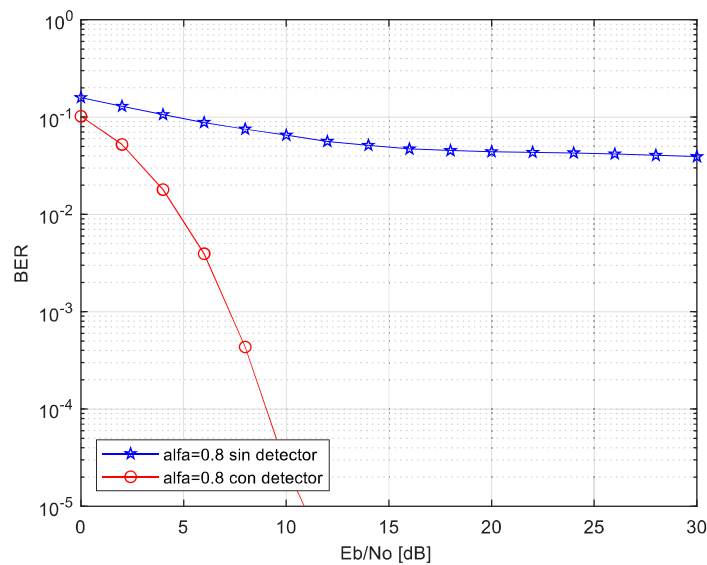


Figura 3.22. BER vs E_b/N_0 con $N = 16$, $\alpha = 0.8$, $v = 40$, $m = 2$ con y sin detector

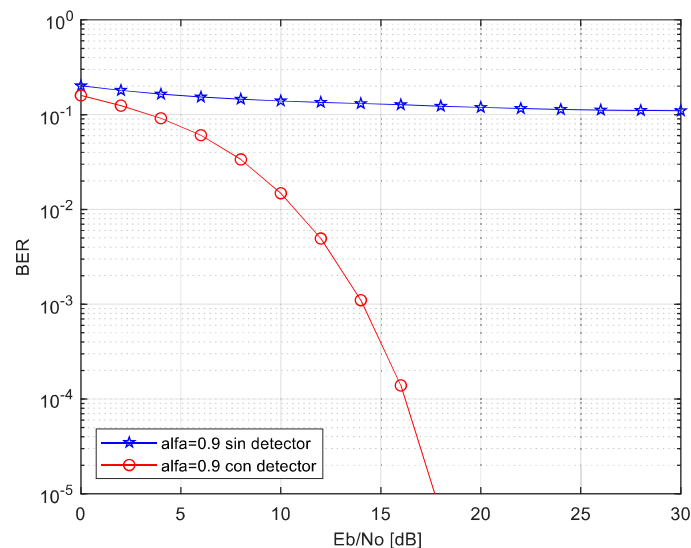


Figura 3.23. BER vs E_b/N_0 con $N = 16$, $\alpha = 0.9$, $v = 40$, $m = 4$ con y sin detector

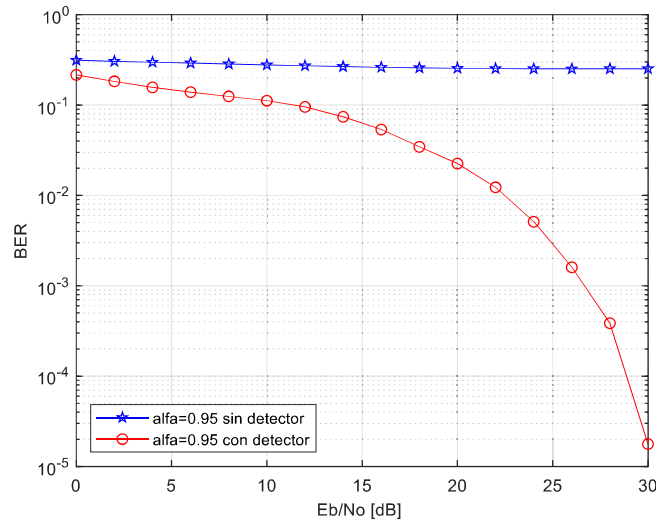


Figura 3.24. BER vs Eb/No con $N = 16$, $\alpha = 0.95$, $v = 40$, $m = 6$ con y sin detector

En todas estas figuras se observa la importancia del detector dentro del sistema SEFDM para los diferentes esquemas de modulación. La curva de color azul representa la BER sin detector mientras que la curva de color rojo es el resultado de la BER con el detector. Para todos los casos la minimización de la BER es muy evidente.

Por lo tanto, la inserción del detector dentro del receptor en este sistema es de vital importancia.

3.2.5. DEGRADACIÓN DE LA BER vs ALFA \propto

En la Figura 3.25 se detalla la degradación de la BER con $N = 16$, $v = 40$, $m = 2$ para varios valores de α , para obtener esta degradación se analiza los valores de Eb/No para cada valor de α a una BER de 10^{-3} , en donde se observa que mientras más pequeño es el valor de α la degradación de la BER aumenta significativamente. Los valores para cada α con su correspondiente degradación se encuentran en la Tabla 3.1.

Tabla 3.1. Valores para α y degradación BER [dB]

α	Degradación BER [dB]
0.65	16.14
0.7	3.58
0.8	0.54
0.9	0.08

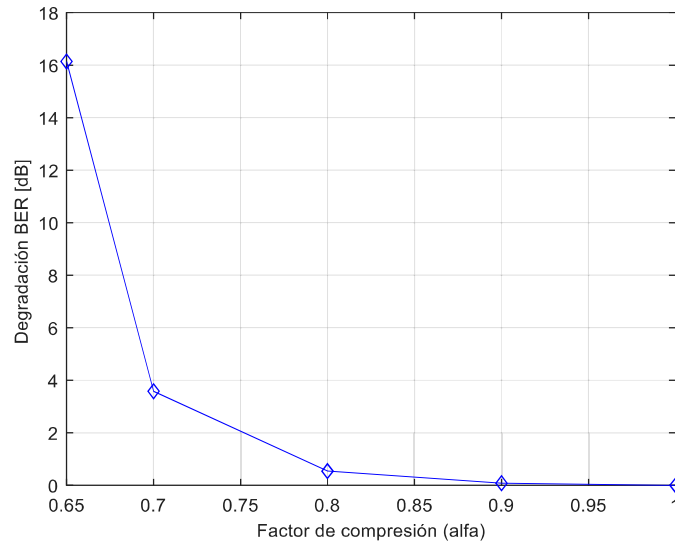


Figura 3.25. Degradación BER vs α con $N = 16$, $\nu = 40$, $m = 2$

3.3. RESULTADOS DE PSD

Para obtener resultados de PSD se realizan simulaciones en la parte del transmisor tanto para OFDM y SEFDM, para esto es necesario realizar el sobremuestreo de la señal en el dominio de la frecuencia, es decir antes de aplicar el proceso de *ifft*, lo cual se implementa mediante la técnica zero-padding, que consiste en el relleno de ceros en la mitad del símbolo, para ello se utiliza un factor de sobremuestreo, en este caso $J = 2$, por lo tanto a la salida del bloque *ifft* el símbolo en el dominio del tiempo tendrá un tamaño de $J * N$.

En el transmisor de los dos sistemas la variante de simulación tanto para la formación de símbolos OFDM como de SEFDM son los esquemas de modulación 4-QAM, 16-QAM y 64-QAM. En el transmisor del sistema SEFDM para cada esquema de modulación se introdujo el factor de compresión $\alpha = \{0.2, 0.4, 0.6, 0.8, 1\}$.

3.3.1. RESULTADOS CON OFDM

El esquema general de comunicaciones que utiliza la técnica de transmisión multiportadora OFDM se simula para obtener la PSD vs frecuencia, con subportadoras $N = 64$, un prefijo cíclico $PC = N/4$ y para diferentes esquemas de modulación que se define con el valor de m . En el transmisor se genera un símbolo OFDM en base a bits aleatorios que ingresan al transmisor. Para visualizar mejor el espectro se forman 500

símbolos OFDM para ello se utiliza la variable $total = 500$ que indica el número de símbolos OFDM que se genera en el transmisor.

A continuación, en la Figura 3.26 se presenta la PSD vs frecuencia para diferentes esquemas de modulación, aquí se observa que al incrementar el nivel de modulación m no afecta a la PSD, siendo la misma para los diferentes valores de m . El nivel de modulación no influye a la PSD, por lo tanto, las gráficas de PSD son correctas.

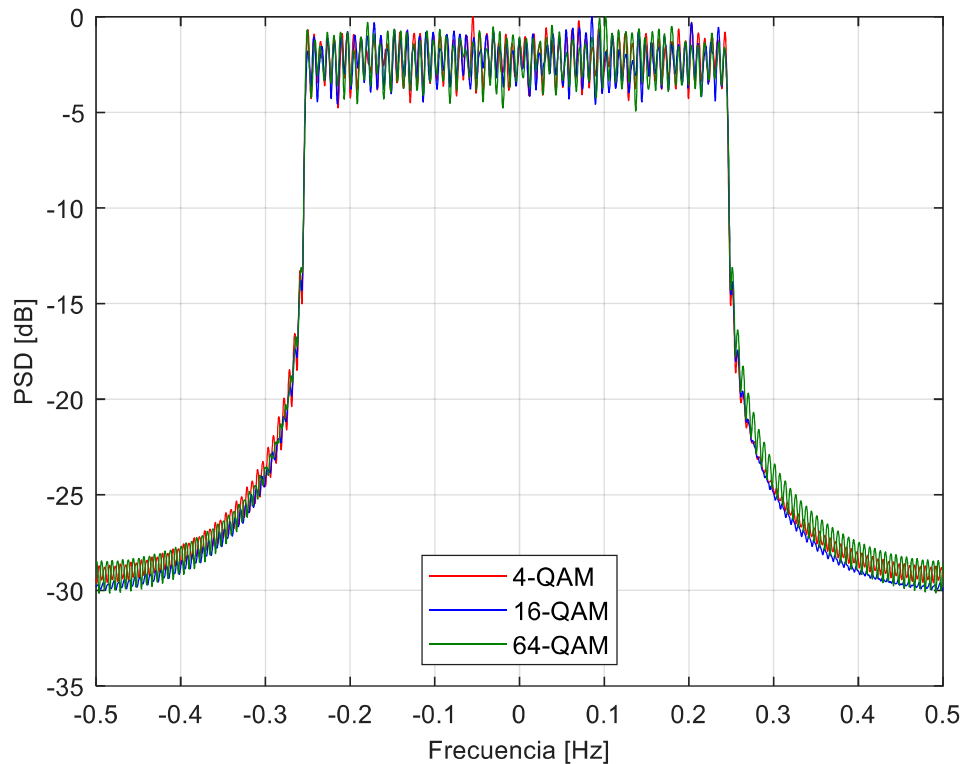


Figura 3.26. PSD vs frecuencia con $N = 64$ para diferentes esquemas de modulación

3.3.2. RESULTADOS CON SEFDM

El esquema general de comunicaciones que utiliza la técnica de transmisión multiportadora SEFDM se simula en el transmisor para obtener las gráficas de PSD vs frecuencia, con un valor de subportadora $N = 64$, factor de compresión $alfa = \{0.2, 0.4, 0.6, 0.8, 1\}$, niveles de modulación $m = \{2, 4, 6\}$. En el transmisor se genera un símbolo SEFDM en base a bits aleatorios que ingresan al transmisor.

Para visualizar mejor el espectro se forman 500 símbolos SEFDM para ello se utiliza la variable $total = 500$ que indica el número de símbolos SEFDM que se genera en el transmisor.

Para comprobar el correcto funcionamiento del sistema SEFDM se compara el espectro de SEFDM para $alfa = 1$ con el espectro de OFDM que se observa en la Figura 3.27, en este caso los dos espectros ocupan el mismo ancho de banda, por lo tanto, el funcionamiento de SEFDM es correcto. El espectro de los dos sistemas varía ligeramente debido a que en el sistema OFDM se introdujo el prefijo cíclico, mientras que en SEFDM no cuenta con dicho prefijo cíclico.

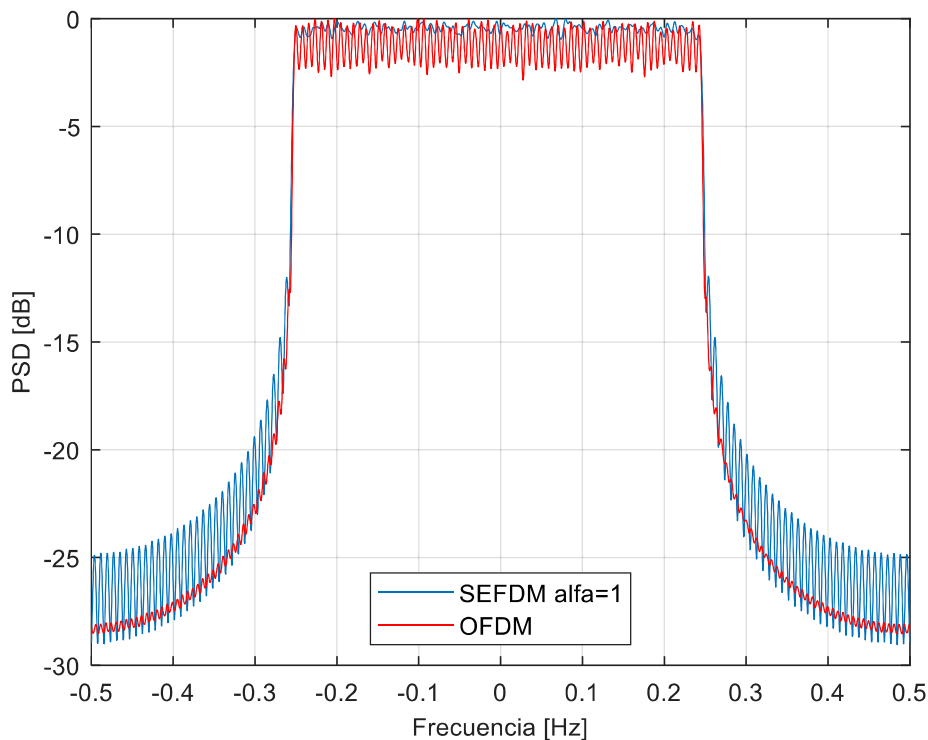


Figura 3.27. PSD vs frecuencia con $N = 64, m = 2$ para SEFDM $alfa = 1$ y OFDM

En la Figura 3.28 se tiene el espectro de SEFDM para $alfa = 1$ con el espectro de OFDM. Este espectro se asemeja con la Figura 3.27 con la diferencia que en los dos sistemas se introdujo el prefijo cíclico. Por tal razón los espectros son exactamente iguales.

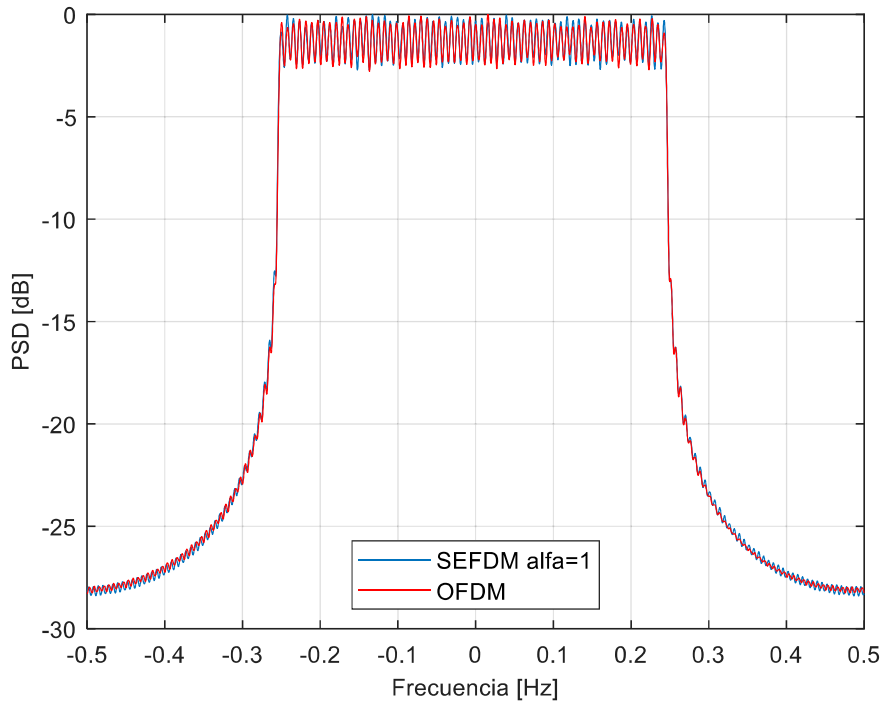


Figura 3.28. PSD vs frecuencia con $N = 64, m = 2$ para SEFDM $\alpha = 1$ y OFDM con prefijo cíclico

En las Figuras 3.29, 3.30 y 3.31 se obtiene las curvas PSD vs frecuencia en el sistema SEFDM para los diferentes esquemas de modulación 4-QAM, 16-QAM y 64-QAM respectivamente.

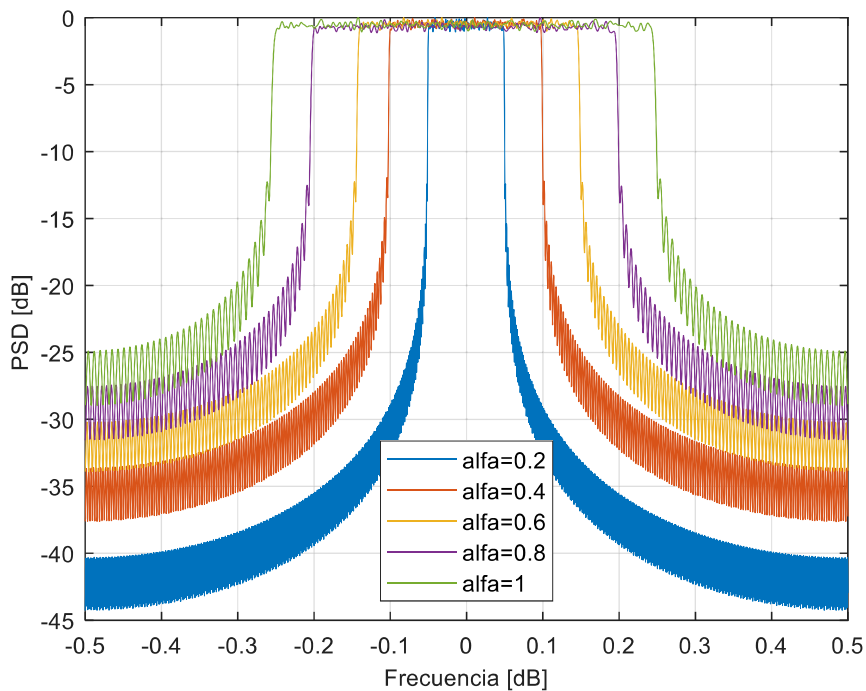


Figura 3.29. PSD vs frecuencia con $N = 64, m = 2$ para SEFDM

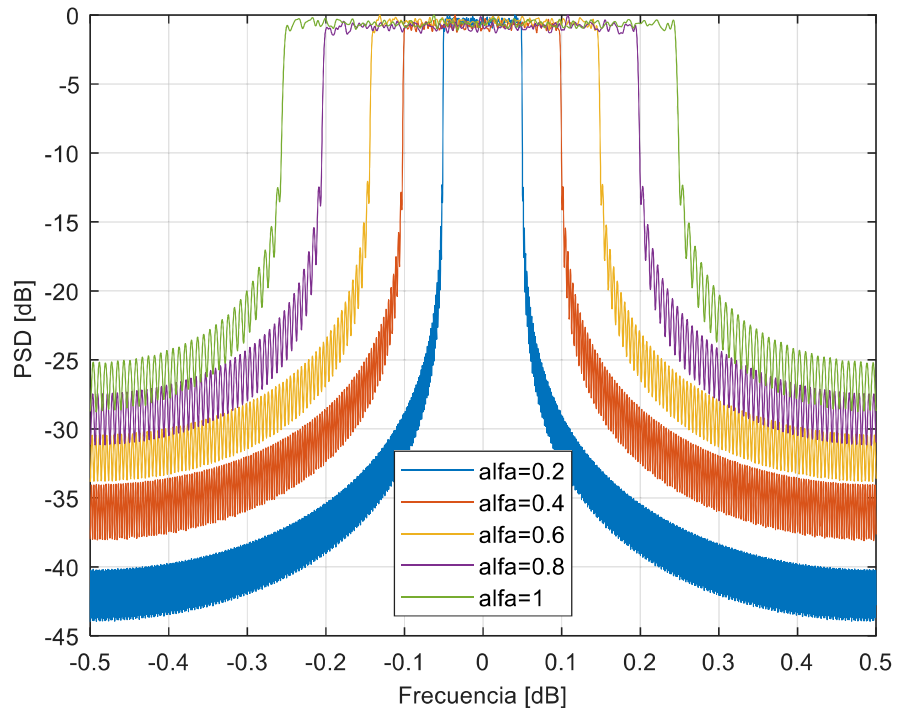


Figura 3.30. PSD vs frecuencia con $N = 64, m = 4$ para SEFDM

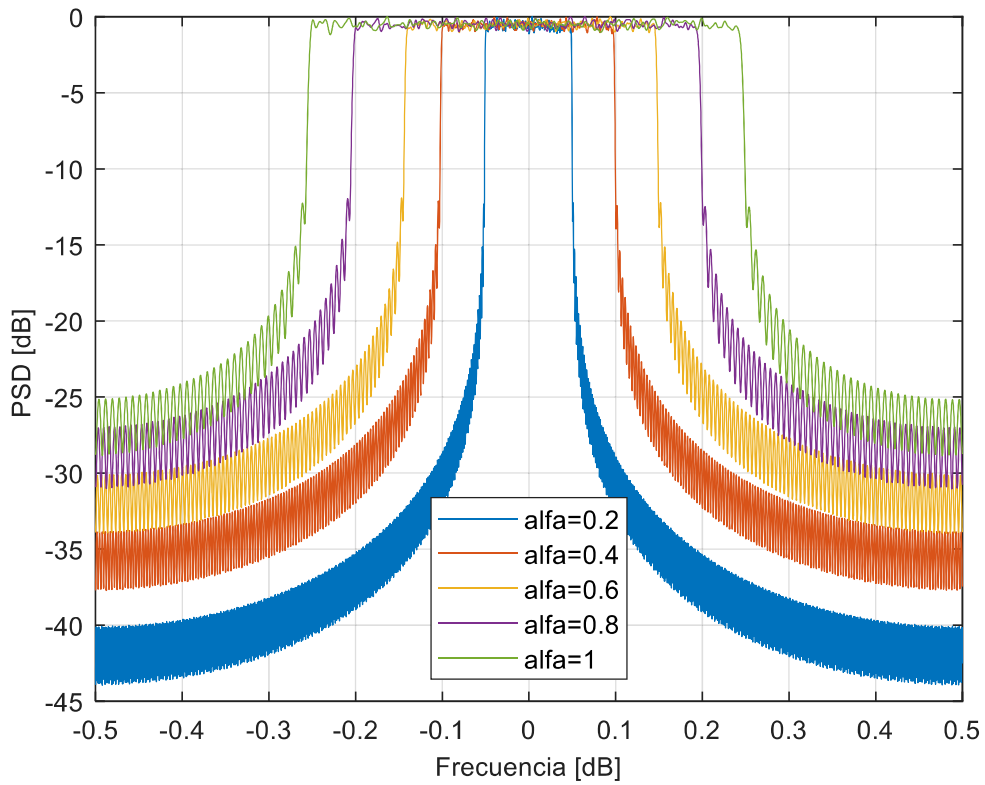


Figura 3.31. PSD vs frecuencia con $N = 64, m = 6$ para SEFDM

En las Figuras 3.29, 3.30 y 3.31 se observa el ahorro de ancho de banda para los diferentes valores de alfa. Mientras más pequeño es el valor de *alfa* existe más compresión del espectro.

En la Figura 3.32 se detalla el porcentaje de ahorro de ancho de banda con $N = 64$, $m = 2$ para varios valores de *alfa*, en donde se observa que mientras más pequeño es el valor de *alfa* se comprime más el espectro. Los valores para cada *alfa* con su correspondiente porcentaje de ahorro de ancho de banda se encuentran en la Tabla 3.2.

Tabla 3.2. Valores para *alfa* y ahorro de ancho de banda [%]

<i>alfa</i>	Ahorro de ancho de banda [%]
0.2	80.05
0.4	59.83
0.6	41.90
0.8	19.95

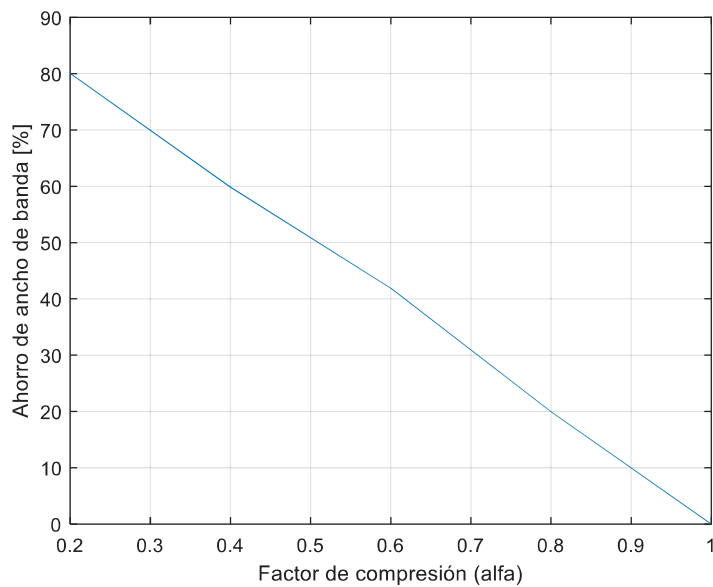


Figura 3.32. Ahorro de ancho de banda [%] vs *alfa* con $N = 64$, $m = 2$

4. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se detallan las principales conclusiones y recomendaciones obtenidas de este proyecto técnico.

4.1. CONCLUSIONES

En este proyecto técnico se implementó en MATLAB un sistema OFDM y un SEFDM, para mediante simulaciones obtener resultados de BER vs Eb/No y PSD que permitieron analizar los beneficios e inconvenientes del sistema SEFDM con respecto a OFDM. De este trabajo se obtuvo las siguientes conclusiones:

- La técnica de transmisión OFDM es la más utilizada dentro de las comunicaciones inalámbricas por sus diversas ventajas, sin embargo, las nuevas necesidades dentro de las comunicaciones inalámbricas, que son cada vez más exigentes, conducen a la búsqueda de nuevas técnicas de transmisión como SEFDM, la cual busca ahorrar el espectro con respecto a OFDM al enviar la misma cantidad de datos en un ancho de banda más pequeño.
- SEFDM, al perder ortogonalidad de las subportadoras, reduce el espacio entre subportadoras de OFDM, dependiendo del factor de compresión que se utilice $\alpha = \{0.2, 0.4, 0.6, 0.8, 1\}$. En donde, mientras más bajo es el valor de α se obtuvo más compresión del espectro, para $\alpha = 1$ no existe compresión lo que es equivalente a un sistema OFDM.
- En SEFDM, la mejora en la reducción del espectro se produce a costa del incremento de la BER con respecto a OFDM debido a la no-ortogonalidad de las subportadoras. Al tener más compresión del espectro las curvas de BER se van degradando, por ejemplo en un esquema 4-QAM, la BER se degrada en $\{0.08, 0.54, 3.58, 16.14\}$ [dB] para valores de $\alpha = \{0.9, 0.8, 0.7, 0.65\}$ respectivamente a una BER de 10^{-3} .
- La utilización del detector iterativo fue una parte importante en el receptor dentro del esquema SEFDM puesto que ayudó a la mejora de la BER. El número de iteraciones v del detector influye en la mejora de la BER. De manera general a un mayor número de iteraciones la BER va mejorando. Por lo tanto, se definió un valor de $v = 40$, con el cual se obtuvo mejores resultados de BER para los

diferentes esquemas de modulación. Por ejemplo para un esquema 4-QAM con $N = 16$, $\alpha = 0.65$, se mejoró la BER en $\{19.05, 16.26, 15.23\}$ [dB] para valores de $\nu = \{20, 30, 40\}$ respectivamente a una BER de 10^{-2} .

- El nivel de modulación m también afectó en la degradación de la BER, ya que la degradación de la BER es mayor en el nivel de modulación más alto 64-QAM, por lo tanto, al incrementar el valor de m se obtuvieron curvas BER con más degradación.
- Al analizar los diferentes tipos de modulaciones se concluye que para valores de α pequeños, es decir mientras más se comprime el espectro la BER aumentara significativamente. Por lo tanto para una modulación 4-QAM los valores de $\alpha = \{0.65, 0.7, 0.8, 0.9\}$ pueden considerarse factores de compresión aceptables sin que la BER se vea muy afectada. Para 16-QAM los valores de $\alpha = \{0.8, 0.9, 0.95\}$ son aceptables mientras que para 64-QAM la curva BER para $\alpha = \{0.95\}$ cuentan con poca degradación de la BER.
- El número de subportadoras es otro parámetro que influyó en la degradación de la BER, de manera general a mayor número de subportadoras N existe una mayor degradación de las curvas.
- Para valores $\alpha = \{0.2, 0.4, 0.6, 0.8\}$ el porcentaje de ahorro de ancho de banda es $\{80.05 \%, 59.83 \%, 41.90 \%, 19.95 \%\}$ con respecto a $\alpha = 1$ que se considera OFDM, es decir, mientras más pequeño es el valor de α se tendrá más ahorro de ancho de banda con respecto a OFDM.

4.2. RECOMENDACIONES

- SEFDM permite ahorrar el espectro, el cual es un recurso cada vez más limitado y costoso. Sin embargo, hay que tener en cuenta los factores de compresión para evitar mucha degradación de la BER.
- Para obtener resultados de las curvas BER confiables el número de iteraciones del sistema debe ser alto, con un valor de 10000 o más para obtener resultados adecuados a una BER de 10^{-3} .

- A pesar de que el detector ID cuente con un mejor rendimiento en términos de BER en comparación con ML, ZF, MMSE y TSVD, el sistema implementado con ID tiene una limitación inherente, en la cual no funciona para valores altos de N , por tal razón, se recomienda para otros trabajos probar la técnica SEFDM con otros detectores.
- Se recomienda para futuros trabajos o líneas de investigación analizar técnicas para la reducción de la PAPR en sistemas SEFDM ya que las señales de múltiples subportadoras son propensas a un PAPR alta.
- Otra línea de investigación interesante es el análisis de SEFDM en transmisión sobre canales MIMO (Multiple-input Multiple-output).

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] R. Nee and R. Prasad, "Introduction, Coding and Modulation," in *OFDM for Wireless Multimedia Communications*, Boston, USA: Artech House, 2000, pp. 1–62.
- [2] D. Rainnie, Y. Feng, and J. Bajcsy, "On capacity merits of spectrally efficient FDM," *Waveforms and Signal Processing*, Canada, pp. 581–586, 2015.
- [3] G. J. García Japon, "Estudio de forma de ondas candidatas para 5G: caso UFMC," *Universidad de Sevilla*, Sevilla, 2019.
- [4] I. Darwazeh, H. Ghannam, and T. Xu, "The first 15 years of SEFDM: a brief survey," *IEEE 11th international Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, 2018.
- [5] M. C. Paredes Paredes, "Algoritmos para la reducción de los picos de potencia en los sistemas OFMD," *Universidad Carlos III de Madrid*, Leganés, Madrid, 2014.
- [6] R. Ness, P. Linnartz, and L. Van der Perre, "The OFMD Principle," in *Wireless OFMD Systems*, vol. 692, Belgium: KluwerAcademic Publishers, 2002, pp. 33–51.
- [7] M. Ergen, "Principles of OFMD," in *Mobile Broadband*, 1st ed., Berkeley, USA: Springer, 2009, pp. 109–175.
- [8] R. W. Chang, "Synthesis of band-limited orthogonal signals for multichannel data transmission," *Bell Syst. Tech. J.*, vol. 45, no. 10, pp. 1775–1796, 1966.
- [9] R. W. Chang, "Orthogonal frequency multiplex data transmission system," no. 3 488 445, USA: United States Patent Office, 1970.
- [10] B. R. Saltzberg, "Performance of an efficient parallel data transmission system," *IEEE Trans. Commun. Technol.*, vol. COM-15, no. 6, pp. 508–811, 1967.
- [11] S. B. Weinstein and P. M. Ebert, "Data transmission by frequency-division multiplexing using the discrete fourier transform," *IEEE Trans. Commun. Technol.*, vol. COM-19, no. 5, pp. 628–634, 1971.
- [12] A. Peled and A. Ruiz, "Frequency domain data transmission using reduced computational complexity algorithms," *IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP '80)*, pp. 964–967, 1980.

- [13] L. A. Masabanda, "Comparación de las formas de onda FBMC y UFMC basadas en OFDM a través de simulaciones en Matlab," Escuela Politecnica Nacional, Quito, 2019.
- [14] M. S. Jimenez, "Introducción, Modulación Digital y Módems Telefónicos," *Comunicación Digital*, Quito, pp. 4–114.
- [15] A. Artés, F. Pérez González, J. Cid, R. López, C. Mosquera, and F. Pérez Cruz, "Modulaciones de amplitud," in *Comunicaciones Digitales*, España: Universidad Carlos III de Madrid, 2012, pp. 247–298.
- [16] A. Albiol, V. Naranjo, and J. Prades, "Transformada Discreta de Fourier," in *Tratamiento Digital de la Señal*, Valencia: Universidad Politécnica de Valencia, 2007, pp. 39–82.
- [17] C. Beard and W. Stallings, "Overview of Wireless Communication," in *Wireless Communication Networks and Systems*, Hoboken, USA: Pearson Higher Education, Inc., 2016, pp. 94–114.
- [18] V. N. Moreno Hermosa, "Simulación de un Sistema OFDM con diversidad de antena en recepción usando Matlab," Escuela Politécnica Nacional, Quito, 2018.
- [19] T. Xu and I. Darwazeh, "Spectrally efficient FDM: spectrum saving technique for 5G," *1st IEEE Int. Conf. 5G Ubiquitous Connect.*, pp. 273–278, 2014.
- [20] J. Heydari, F. Naeiny, and F. Marvasti, "Iterative detection with soft decision in spectrally efficient FDM systems," *Shahed University of Technology*, Tehran, Iran, pp. 1–6, 2016.
- [21] M. Rodrigues and I. Darwazeh, "Fast OFDM: a proposal for doubling the data rate of OFDM schemes," *Int. Conf. Telecommun.*, vol. 3, pp. 484–487, 2002.
- [22] M. Rodrigues and I. Darwazeh, "A spectrally efficient frequency division multiplexing based communications system," *8th Int. OFDM-Workshop, Proceedings, Hambg.*, pp. 70–74, 2003.
- [23] H. Ghannam and I. Darwazeh, "SEFDM: spectral efficiency upper bound and interference distribution," *2018 12th IEEE International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, Budapest, Hungary, pp. 1–6, 2018.
- [24] T. Xu and I. Darwazeh, "M-QAM signal detection for a non-orthogonal system using an improved fixed sphere decoder," *2014 9th IEEE/IET Int. Symp. Commun.*

Syst. Networks Digit. Sign Process., pp. 623–627, 2014.

- [25] S. Ahmed, “Spectrally efficient FDM communication signals and transceivers: design, mathematical modelling and system optimization,” University College London, London, 2011.
- [26] S. Ahmed and I. Darwazeh, “IDFT based transmitters for spectrally efficient FDM system,” *IDFT based transmitters for spectrally efficient FDM system*, London, pp. 1–4, 2009.
- [27] R. Grammenos, S. Isam, and I. Darwazeh, “FPGA design of a truncated SVD based receiver for the detection of SEFDM signals,” *2011 IEEE 22nd Int. Symp. Pers. Indoor Mob. Radio Commun.*, pp. 2085–2090, 2011.
- [28] M. Jia, Z. Yin, Q. Guo, G. Liu, and X. Gu, “Waveform design of zero head DFT spread spectral efficient frequency division multiplexing,” *IEEE Access*, vol. 5, pp. 16944–16952, 2017.
- [29] T. Xu, R. Grammenos, F. Marvasti, and I. Darwazeh, “An improved fixed sphere decoder employing soft decision for the detection of non-orthogonal signals,” *IEEE Commun. Lett.*, vol. 17, no. 10, pp. 1964–1967, 2013.
- [30] S. Isam and I. Darwazeh, “Robust channel estimation for spectrally efficient FDM system,” *19th Int. Conf. Telecommun. (ICT 2012)*, pp. 1–6, 2012.
- [31] T. Xu, “Bandwidth compressed waveform and system design for wireless and optical communications: theory and practice,” University College London, London, 2017.
- [32] I. Kanaras, A. Chorti, M. Rodrigues, and I. Darwazeh, “Spectrally efficient FDM signals: bandwidth gain at the expense of receiver complexity,” *IEEE ICC*, pp. 1–6, 2009.
- [33] S. Isan, I. Kanaras, and I. Darwazeh, “A truncated SVD approach for fixed complexity spectrally efficient FDM receivers,” *IEEE Wirel. Commun. Netw. Conf.*, pp. 1584–1589, 2011.
- [34] R. Grammenos and I. Darwazeh, “Performance trade-offs and DSP evaluation of spectrally efficient FDM detection techniques,” *IEEE ICC- Signal Process. Commun. Symp.*, pp. 4781–4786, 2013.
- [35] I. Darwazeh, T. Xu, T. Gui, Y. Bao, and Z. Li, “Optical SEFDM system: bandwidth saving using non-orthogonal sub-carriers,” *IEEE Photonics Technol. Lett.*, vol. 26,

no. 4, pp. 352–355, 2014.

- [36] J. Huang, Q. Sui, Z. Li, and F. Ji, “Experimental demonstration of 16-QAM DD-SEFDM with cascaded BPSK iterative detection,” *IEEE Photonics J.*, vol. 8, no. 3, pp. 1–9, 2016.
- [37] W. Stallings, “Transmission Fundamentals,” in *Wireless Communications and Networks*, 2nd ed., United States of America: Pearson Education, Inc, 2005, pp. 14–43.
- [38] G. Breed, “Bit Error Rate: Fundamental concepts and measurement issues,” *High Frequency Electron.*, pp. 46–48, 2003.
- [39] V. K. Garg, “An overview of digital communication and transmission,” in *Wireless Communications and Networking*, Amsterdam: Elsevier Inc, 2007.
- [40] “Uniformly Distributed Pseudorandom Integers, MathWorks.” .
- [41] “Reshape Symbolic Array, MathWorks.” .
- [42] “Quadrature amplitude modulation (qammod), MathWorks.” .
- [43] “Inverse fast Fourier transform (ifft), MathWorks.” .
- [44] “Add white Gaussian noise to signal (awgn), MathWorks.” .
- [45] “Fast Fourier transform (fft), MathWorks.” .
- [46] “Quadrature amplitude demodulation (qamdemod).” .
- [47] “Number of bit errors and bit error rate (BER), MathWorks.” .
- [48] “Estimación de densidad espectral de potencia de Welch (pwelch).” .
- [49] “Control random number generation (rng), MathWorks.” .
- [50] “Controlling Random number Generation, MathWorks.” .
- [51] “Identity matrix, MathWorks.” .

ANEXOS

En esta sección se detalla el código de MATLAB realizado que se describe de la siguiente manera:

ANEXO A. Resultados de BER vs Eb/No en OFDM.

ANEXO B. Resultados de PSD en OFDM.

ANEXO C. Función detector ID.

ANEXO D. Resultados de BER vs Eb/No en SEFDM.

ANEXO E. Función SEFDM

ANEXO F. Resultados de PSD en SEFDM.

ANEXO G. Función SEFDM_tx

ANEXO A: RESULTADOS DE BER vs Eb/No EN OFDM

En este anexo se detalla el código para obtener resultados de BER vs Eb/No en OFDM.

a) Ingreso de variables

```
N=64;           % Número de subportadoras
m=2;           % Agrupación de bits según modulación 2(4-QAM)/4(16-
QAM)/6(64-QAM)
Ntrials=10000; % Número de iteraciones para cada valor de SNR
EbNo=0:2:30;   % Valor Eb/No
```

b) Valores iniciales

```
M=2^m;         % modulación 4-QAM/16-QAM/64-QAM
CP=N/4;       % Prefijo cíclico
```

c) Transmisor OFDM

```
% Generación de bits aleatorios
tx_bit_s=randi([0 1],1,N*m);
% Conversion serie a paralelo
tx_bit_p=reshape(tx_bit_s,N*m,1);
% Modulación digital
tx_modul=qammod(tx_bit_p,M,'InputType','bit');
% IFFT
tx_ifft=ifft(tx_modul,N);
% Toma de N/4 muestras finales(CP)
tx_prefijo=tx_ifft(N-(CP-1):N);
% Símbolo OFDM con prefijo cíclico(CP)
tx_out_conCP=[tx_prefijo;tx_ifft];
% Conversion paralelo a serie
```

```
tx_out_s=reshape(tx_out_conCP,1,length(tx_out_conCP));
```

d) Inicialización de la BER

```
SNR=EbNo+10*log10(m);  
  
BER_M=zeros(Ntrials,length(SNR));
```

e) Canal y Receptor

```
for i=1:length(SNR)  
    for j=1:Ntrials  
        %Ruido AWGN en el canal  
        canal=awgn(tx_out_s,SNR(i),'measured','db');  
  
        %Receptor OFDM  
  
        % Conversión serie a paralelo  
        rx_ofdm_p=reshape(canal,length(canal),1);  
        % Extracción del prefijo cíclico (CP)  
        rx_ofdm_p_sinCP=rx_ofdm_p(CP+1:length(rx_ofdm_p));  
        % FFT  
        rx_fft=fft(rx_ofdm_p_sinCP,N);  
        % Demodulación  
        rx_bit=qamdemod(rx_fft,M,'OutputType','bit');  
        % Conversión paralelo a serie  
        rx_bit_s=reshape(rx_bit,1,length(rx_bit));
```

f) Cálculo de errores y BER

```
        % número de errores  
        num_error_BER=biterr(tx_bit_s,rx_bit_s);  
        % matriz BER de tamaño Ntrials X longitud del vector EbNo  
        BER_M(j,i)= num_error_BER./(N*m);  
    end  
end  
% Promedio de columnas de la matriz BER M,  
BER=mean(BER_M);  
% BER teórico  
BER_t=berawgn(EbNo,'qam',M);
```

g) Gráfica de BER simulado vs BER teórico

```
semilogy(EbNo,BER,'Color','y','Linewidth',2.5);  
axis([0 20 10^-5 1])  
xlabel('Eb/No [dB]','FontSize',10)  
ylabel('BER','FontSize',10)
```

```

grid on
hold on
semilogy(EbNo,BER_t,'r--','Linewidth',1.5)
legend({'BER SIMULADO','BER TEÓRICO'],'FontSize',10)
'FontSize',10)

```

ANEXO B: RESULTADOS DE PSD EN OFDM

En este anexo se detalla el código para obtener resultados de PSD en OFDM.

a) Ingreso de variables

```

N=64;           % Número de subportadoras
m=2;           % Agrupación de bits según modulación 2(4-QAM)/4(16-
QAM)/6(64-QAM)
J=2;           % Factor de sobremuestreo
total=500;     % Cantidad de símbolos OFDM

```

b) Valores iniciales

```

M=2^m;         % modulación 4-QAM/16-QAM/64-QAM
CP=(J*N)/4;   % Prefijo ciclico

```

c) Transmisor OFDM

```

% Vector que almacena los símbolos totales
total_simb=[];

for j=1:1:total

% Generación de bits aleatorios
tx_bit_s=randi([0 1],1,N*m);
% Conversion serie a paralelo
tx_bit_p=reshape(tx_bit_s,N*m,1);
% Modulación digital
tx_modul=qammod(tx_bit_p,M,'InputType','bit');
% Zero-Padding con J=2
relleno=zeros(N*(J-1),1);
tx_relleno=[tx_modul(1:N/2);relleno;tx_modul(N/2+1:N)];
% IFFT
tx_ifft=ifft(tx_relleno,J*N);
% Toma de N/4 muestras finales(CP)
tx_prefijo=tx_ifft(J*N-(CP-1):J*N);
% Símbolo OFDM con prefijo cíclico(CP)
tx_out_conCP=[tx_prefijo;tx_ifft];
% Conversion paralelo a serie
tx_out_s=reshape(tx_out_conCP,1,length(tx_out_conCP));

```

```

% Vector con símbolos totales
total_simb=[total_simb tx out s];
end

```

d) Cálculo de PSD

```

fs=1;          % Frecuencia de muestreo
W=4096;       % Tamaño de las frecuencias normalizadas
[Pxx,W]=pwelch(total_simb,hamming((length(total_simb)/200)),[],W,fs);
plot(linspace(-
fs/2,fs/2,length(Pxx)),10*log10(fftshift(Pxx/max(Pxx))), 'Color','r');
ylabel('PSD [dB]','FontSize',10)
xlabel('Frecuencia [Hz]','FontSize',10)
grid on
hold on

```

ANEXO C: FUNCIÓN DETECTOR ID

En este anexo se detalla el código del detector ID.

```

% En esta función se busca obtener el vector Sv, el cual contiene
% los símbolos recuperados

% Sv= Vector de símbolos recuperados luego de v iteraciones
% Sv_a= Vector de símbolos estimados luego de v-1 iteraciones

function [Sv] = ID(R,v,parte_ID,m)

```

a) Detector para modulación 4-QAM

```

if m==2          % 4-QAM
    Sv_a=R;      % Vector a la salida de FFT
    for mm=1:1:v % Bucle para iteraciones del detector
        Sv=R+parte_ID*Sv_a; % Proceso de iteración ID
        d=1-mm/v; % Intervalo umbral
        Svr=real(Sv); % Parte real del vector Sv
        Svi=imag(Sv); % Parte imaginaria del vector Sv

        % Mapeo de símbolos
        Sv_a(Svr>=d & Svi>=d)=+1+1i;
        Sv_a(Svr>=d & Svi<=-d)=+1-1i;
        Sv_a(Svr<=-d & Svi>=d)=-1+1i;
        Sv_a(Svr<=-d & Svi<=-d)=-1-1i;

    end

```

b) Detector para modulación 16-QAM


```

elseif m==4 % 16-QAM
    Sv_a=R; % Vector a la salida de FFT
    for mm=1:1:v % Bucle para iteraciones del detector
        Sv=R+parte_ID*Sv_a; % Proceso de iteración ID
        d=1-mm/v; % Intervalo umbral
        Svr=real(Sv); % Parte real del vector Sv
        Svi=imag(Sv); % Parte imaginaria del vector Sv

        % Mapeo de símbolos

        Sv_a(Svr>=2+d & Svi>=2+d)=+3+3i;
        Sv_a(Svr<=-2-d & Svi>=2+d)=-3+3i;
        Sv_a(Svr<=-2-d & Svi<=-2-d)=-3-3i;
        Sv_a(Svr>=2+d & Svi<=-2-d)=+3-3i;

        Sv_a(Svr>=2+d & (Svi>=d & Svi<=2-d))=+3+1i;
        Sv_a((Svr>=d & Svr<=2-d) & Svi>=2+d)=+1+3i;
        Sv_a((Svr<=-d & Svr>=-2+d) & Svi>=2+d)=-1+3i;
        Sv_a(Svr<=-2-d & (Svi>=d & Svi<=2-d))=-3+1i;
        Sv_a(Svr<=-2-d & (Svi<=-d & Svi>=-2+d))=-3-1i;
        Sv_a((Svr<=-d & Svr>=-2+d) & Svi<=-2-d)=-1-3i;
        Sv_a((Svr>=d & Svr<=2-d) & Svi<=-2-d)=+1-3i;
        Sv_a(Svr>=2+d & (Svi<=-d & Svi>=-2+d))=+3-1i;

        Sv_a((Svr>=d & Svr<=2-d) & (Svi>=d & Svi<=2-d))=+1+1i;
        Sv_a((Svr>=d & Svr<=2-d) & (Svi<=-d & Svi>=-2+d))=+1-1i;
        Sv_a((Svr<=-d & Svr>=-2+d) & (Svi>=d & Svi<=2-d))=-1+1i;
        Sv_a((Svr<=-d & Svr>=-2+d) & (Svi<=-d & Svi>=-2+d))=-1-1i;

    end

```

c) Detector para modulación 64-QAM

```

elseif m==6 % 64-QAM

    Sv_a=R; % Vector a la salida de FFT
    for mm=1:1:v % Bucle para iteraciones del detector
        Sv=R+parte_ID*Sv_a; % Proceso de iteración ID
        d=1-mm/v; % Intervalo umbral
        Svr=real(Sv); % Parte real del vector Sv
        Svi=imag(Sv); % Parte imaginaria del vector Sv

        % Mapeo de símbolos

        Sv_a(Svr>=6+d & Svi>=6+d)=+7+7i;
        Sv_a(Svr<=-6-d & Svi>=6+d)=-7+7i;
        Sv_a(Svr<=-6-d & Svi<=-6-d)=-7-7i;
        Sv_a(Svr>=6+d & Svi<=-6-d)=+7-7i;

        Sv_a((Svr>=d & Svr<=2-d) & Svi>=6+d)=+1+7i;
        Sv_a((Svr>=2+d & Svr<=4-d) & Svi>=6+d)=+3+7i;

```

```

Sv_a((Svr>=4+d & Svr<=6-d) & Svi>=6+d)=+5+7i;
Sv_a((Svr>=6+d & (Svi>=4+d & Svi<=6-d))=+7+5i;
Sv_a((Svr>=6+d & (Svi>=2+d & Svi<=4-d))=+7+3i;
Sv_a((Svr>=6+d & (Svi>=d & Svi<=2-d))=+7+1i;
Sv_a((Svr<=-d & Svr>=-2+d) & Svi>=6+d)=-1+7i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & Svi>=6+d)=-3+7i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & Svi>=6+d)=-5+7i;
Sv_a((Svr<=-6-d & (Svi>=4+d & Svi<=6-d))=-7+5i;
Sv_a((Svr<=-6-d & (Svi>=2+d & Svi<=4-d))=-7+3i;
Sv_a((Svr<=-6-d & (Svi>=d & Svi<=2-d))=-7+1i;
Sv_a((Svr<=-6-d & (Svi<=-d & Svi>=-2+d))=-7-1i;
Sv_a((Svr<=-6-d & (Svi<=-2-d & Svi>=-4+d))=-7-3i;
Sv_a((Svr<=-6-d & (Svi<=-4-d & Svi>=-6+d))=-7-5i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & Svi<=-6-d)=-5-7i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & Svi<=-6-d)=-3-7i;
Sv_a((Svr<=-d & Svr>=-2+d) & Svi<=-6-d)=-1-7i;
Sv_a((Svr>=d & Svr<=2-d) & Svi<=-6-d)=+1-7i;
Sv_a((Svr>=2+d & Svr<=4-d) & Svi<=-6-d)=+3-7i;
Sv_a((Svr>=4+d & Svr<=6-d) & Svi<=-6-d)=+5-7i;
Sv_a((Svr>=6+d & (Svi<=-4-d & Svi>=-6+d))=+7-5i;
Sv_a((Svr>=6+d & (Svi<=-2-d & Svi>=-4+d))=+7-3i;
Sv_a((Svr>=6+d & (Svi<=-d & Svi>=-2+d))=+7-1i;

Sv_a((Svr>=d & Svr<=2-d) & (Svi>=d & Svi<=2-d))=+1+1i;
Sv_a((Svr>=d & Svr<=2-d) & (Svi>=2+d & Svi<=4-d))=+1+3i;
Sv_a((Svr>=d & Svr<=2-d) & (Svi>=4+d & Svi<=6-d))=+1+5i;
Sv_a((Svr>=2+d & Svr<=4-d) & (Svi>=d & Svi<=2-d))=+3+1i;
Sv_a((Svr>=2+d & Svr<=4-d) & (Svi>=2+d & Svi<=4-d))=+3+3i;
Sv_a((Svr>=2+d & Svr<=4-d) & (Svi>=4+d & Svi<=6-d))=+3+5i;
Sv_a((Svr>=4+d & Svr<=6-d) & (Svi>=d & Svi<=2-d))=+5+1i;
Sv_a((Svr>=4+d & Svr<=6-d) & (Svi>=2+d & Svi<=4-d))=+5+3i;
Sv_a((Svr>=4+d & Svr<=6-d) & (Svi>=4+d & Svi<=6-d))=+5+5i;
Sv_a((Svr<=-d & Svr>=-2+d) & (Svi>=d & Svi<=2-d))=-1+1i;
Sv_a((Svr<=-d & Svr>=-2+d) & (Svi>=2+d & Svi<=4-d))=-1+3i;
Sv_a((Svr<=-d & Svr>=-2+d) & (Svi>=4+d & Svi<=6-d))=-1+5i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & (Svi>=d & Svi<=2-d))=-3+1i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & (Svi>=2+d & Svi<=4-d))=-3+3i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & (Svi>=4+d & Svi<=6-d))=-3+5i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & (Svi>=d & Svi<=2-d))=-5+1i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & (Svi>=2+d & Svi<=4-d))=-5+3i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & (Svi>=4+d & Svi<=6-d))=-5+5i;
Sv_a((Svr<=-d & Svr>=-2+d) & (Svi<=-d & Svi>=-2+d))=-1-1i;
Sv_a((Svr<=-d & Svr>=-2+d) & (Svi<=-2-d & Svi>=-4+d))=-1-3i;
Sv_a((Svr<=-d & Svr>=-2+d) & (Svi<=-4-d & Svi>=-6+d))=-1-5i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & (Svi<=-d & Svi>=-2+d))=-3-1i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & (Svi<=-2-d & Svi>=-4+d))=-3-3i;
Sv_a((Svr<=-2-d & Svr>=-4+d) & (Svi<=-4-d & Svi>=-6+d))=-3-5i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & (Svi<=-d & Svi>=-2+d))=-5-1i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & (Svi<=-2-d & Svi>=-4+d))=-5-3i;
Sv_a((Svr<=-4-d & Svr>=-6+d) & (Svi<=-4-d & Svi>=-6+d))=-5-5i;
Sv_a((Svr>=d & Svr<=2-d) & (Svi<=-d & Svi>=-2+d))=+1-1i;
Sv_a((Svr>=d & Svr<=2-d) & (Svi<=-2-d & Svi>=-4+d))=+1-3i;
Sv_a((Svr>=d & Svr<=2-d) & (Svi<=-4-d & Svi>=-6+d))=+1-5i;
Sv_a((Svr>=2+d & Svr<=4-d) & (Svi<=-d & Svi>=-2+d))=+3-1i;

```

```

Sv_a((Svr>=2+d & Svr<=4-d) & (Svi<=-2-d & Svi>=-4+d))=+3-3i;
Sv_a((Svr>=2+d & Svr<=4-d) & (Svi<=-4-d & Svi>=-6+d))=+3-5i;
Sv_a((Svr>=4+d & Svr<=6-d) & (Svi<=-d & Svi>=-2+d))=+5-1i;
Sv_a((Svr>=4+d & Svr<=6-d) & (Svi<=-2-d & Svi>=-4+d))=+5-3i;
Sv_a((Svr>=4+d & Svr<=6-d) & (Svi<=-4-d & Svi>=-6+d))=+5-5i;

end
end
end

```

ANEXO D: RESULTADOS DE BER vs Eb/No EN SEFDM

En este anexo se detalla el código para obtener resultados de las curvas BER vs Eb/No en SEFDM

a) Ingreso de variables

```

N=16;           % Número de subportadoras
m=2;           % Agrupación de bits según modulación
2(4QAM)/4(16QAM)/6(64QAM)
v=40;          % Número de iteraciones del detector ID
Ntrials=10;    % Número de iteraciones para cada valor de SNR
EbNo=(0:2:30); % Eb/No

```

b) Obtención de vectores BER para diferentes valores de alfa

```

% Se llama a la función SEFDM
BER1=SEFDM(0.2,N,v,m,Ntrials,EbNo);
BER2=SEFDM(0.4,N,v,m,Ntrials,EbNo);
BER3=SEFDM(0.6,N,v,m,Ntrials,EbNo);
BER4=SEFDM(0.7,N,v,m,Ntrials,EbNo);
BER5=SEFDM(0.8,N,v,m,Ntrials,EbNo);
BER6=SEFDM(0.9,N,v,m,Ntrials,EbNo);
BER7=SEFDM(0.95,N,v,m,Ntrials,EbNo);
BER8=SEFDM(1,N,v,m,Ntrials,EbNo);

```

c) Curvas BER vs Eb/No

```

i=0;

if length(BER1)~=1
    semilogy(EbNo,BER1,'+-','Color','0.6350 0.0780 0.1840');
    i=i+1;
    leyendas{i}='alfa=0.2';
end
hold on;

```

```

grid on;

if length(BER2)~=1
    semilogy(EbNo,BER2,'o-','Color','0 0.75 0.75');
    i=i+1;
    leyendas{i}='alfa=0.4';
end

if length(BER3)~=1
    semilogy(EbNo,BER3,'*-', 'Color','0.9290 0.6940 0.1250');
    i=i+1;
    leyendas{i}='alfa=0.6';
end

if length(BER4)~=1
    semilogy(EbNo,BER4,'p-', 'Color','m');
    i=i+1;
    leyendas{i}='alfa=0.7';
end

if length(BER5)~=1
    semilogy(EbNo,BER5,'s-', 'Color','0 0.5 0');
    i=i+1;
    leyendas{i}='alfa=0.8';
end

if length(BER6)~=1
    semilogy(EbNo,BER6,'h-', 'Color','0 0 1');
    i=i+1;
    leyendas{i}='alfa=0.9';
end

if length(BER7)~=1
    semilogy(EbNo,BER7,'o-', 'Color','0.4940 0.1840 0.5560');
    i=i+1;
    leyendas{i}='alfa=0.95';
end

if length(BER8)~=1
    semilogy(EbNo,BER8,'v-', 'Color','r');
    i=i+1;
    leyendas{i}='alfa=1';
end

legend(leyendas, 'location', 'southwest');
axis([0 30 10^-5 1]);
title('BER vs Eb/No en SEFDM para varios alfas');
xlabel('Eb/No [dB] ', 'FontSize',10);
ylabel('BER', 'FontSize',10);

```

ANEXO E: FUNCIÓN SEFDM

En este anexo se detalla el código de la función SEFDM

```
function [BER] = SEFDM(alfa,N,v,m,Ntrials,EbNo)

% Esta función devuelve un vector VER con un Eb/No dado
```

a) Valores iniciales

```
M=2^m; % Modulación 4-QAM/16-QAM/64-QAM
IFFT_FFT_size=round(N/alfa); % Tamaño del bloque IFFT/FFT
```

b) Determinación de una parte del proceso de iteración

```
% Matriz de subportadoras F
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*alfa*n*k/N);
    end
end

Fc=F'; % Matriz F conjugada
C=Fc*F; % Matriz C: hace referencia a la interferencia entre
subportadoras
I=eye(N); % Matriz identidad
parte_ID=I-C; % Parte del proceso de iteración
```

c) Transmisor SEFDM

```
% Controla la salida de números aleatorios
rng default;
% Generación de bits aleatorios
tx_bit_s=randi([0 1],1,N*m);
% Conversion serie a paralelo
tx_bit_p=reshape(tx_bit_s,N*m,1);
% Modulación digital
tx_modul=qammod(tx_bit_p,M,'InputType','bit');
% Definición de bloque IFFT
ceros=zeros(IFFT_FFT_size-N,1);
tx_sefdm=[tx_modul;ceros];
% IFFT
tx_ifft=ifft(tx_sefdm,IFFT_FFT_size);
% Ignorar los últimos valores
tx_out=tx_ifft(1:N);
% Conversion paralelo a serie
tx_out_s=reshape(tx_out,1,length(tx_out));
```

d) Inicialización de la BER

```
SNR=EbNo+10*log10(m);  
  
BER_M=zeros(Ntrials,length(SNR));
```

e) Canal y Receptor

```
for i=1:length(SNR)  
    for j=1:Ntrials  
        %Ruido AWGN  
        canal=awgn(tx_out_s,SNR(i),'measured','db');  
  
        %Receptor SEFDM  
  
        % Conversión serie a paralelo  
        rx_sefdm_p=reshape(canal,length(canal),1);  
        % Definición del bloque FFT  
        rx_sefdm=[rx_sefdm_p;ceros];  
        % FFT  
        rx_fft=fft(rx_sefdm,IFFT_FFT_size);  
        R=rx_fft(1:N);  
        % Detector ID  
        Sv=ID(R,v,parte_ID,m);  
        % Demodulación Digital  
        rx_bit=qamdemod(Sv,M,'OutputType','bit');  
        % Conversión paralelo a serie  
        rx_bit_s=reshape(rx_bit,1,length(rx_bit));
```

f) Cálculo de errores y BER

```
        % número de errores  
        num_error_BER=biterr(tx_bit_s,rx_bit_s);  
        % matriz BER de tamaño Ntrials X longitud del vector EbNo  
        BER_M(j,i)= num_error_BER./(N*m);  
    end  
end  
% Promedio de columnas de la matriz BER_M  
BER=mean(BER_M);  
end
```

ANEXO F: RESULTADOS DE PSD EN SEFDM

En este anexo se detalla el código para obtener resultados de PSD en SEFDM.

a) Ingreso de variables

```
N=64;           % Número de subportadoras
```

```

m=2;           % Agrupación de bits según modulación 2(4-QAM)/4(16-
QAM)/6(64-QAM)
J=2;           % Factor de sobremuestreo
total=100;     % Cantidad de símbolos OFDM
fs=1;          % Frecuencia de muestreo
W=4096;        % Tamaño de las frecuencias normalizadas

```

b) Obtención de vectores total_simb para diferentes valores de alfa

```

% Se llama a la función SEFDM_tx
total_simb1=SEFDM_tx(0.2,N,m,J,total);
total_simb2=SEFDM_tx(0.4,N,m,J,total);
total_simb3=SEFDM_tx(0.6,N,m,J,total);
total_simb4=SEFDM_tx(0.8,N,m,J,total);
total_simb6=SEFDM_tx(1,N,m,J,total);

```

c) Cálculo de PSD

```

i=0;

if length(total_simb1)~=1
    [Pxx,W]=pwelch(total_simb1,hamming((length(total_simb1))/200),[],W,fs);
    plot(linspace(-fs/2,fs/2,length(Pxx)),10*log10(fftshift(Pxx/max(Pxx))));
    i=i+1;
    leyendas{i}='alfa=0.2';
end
hold on;
grid on;

if length(total_simb2)~=1
    [Pxx,W]=pwelch(total_simb2,hamming((length(total_simb2))/200),[],W,fs);
    plot(linspace(-fs/2,fs/2,length(Pxx)),10*log10(fftshift(Pxx/max(Pxx))));
    i=i+1;
    leyendas{i}='alfa=0.4';
end

if length(total_simb3)~=1
    [Pxx,W]=pwelch(total_simb3,hamming((length(total_simb3))/200),[],W,fs);
    plot(linspace(-fs/2,fs/2,length(Pxx)),10*log10(fftshift(Pxx/max(Pxx))));
    i=i+1;
    leyendas{i}='alfa=0.6';
end

if length(total_simb4)~=1
    [Pxx,W]=pwelch(total_simb4,hamming((length(total_simb4))/200),[],W,fs);
    plot(linspace(-fs/2,fs/2,length(Pxx)),10*log10(fftshift(Pxx/max(Pxx))));
    i=i+1;
    leyendas{i}='alfa=0.8';
end

if length(total_simb6)~=1

```

```

[Pxx,W]=pwelch(total_simb6,hamming((length(total_simb6))/200),[],4096,fs);
plot(linspace(-fs/2,fs/2,length(Pxx)),10*log10(fftshift(Pxx/max(Pxx))));
i=i+1;
leyendas{i}='alfa=1';
end

legend(leyendas,'FontSize',10,'location','south');
title('PSD del simbolo SEFDM para varios alfas')
ylabel('PSD [dB] ','FontSize',10)
xlabel('Frecuencia [dB]','FontSize',10)

```

ANEXO G: FUNCIÓN SEFDM_tx

En este anexo se detalla la función SEFDM_tx

```

function [total_simb] = SEFDM_tx(alfa,N,m,J,total)
% Esta función devuelve un vector total_simb

```

a) Valores iniciales

```

M=2^m; % modulación 4-QAM/16-QAM/64-QAM
IFFT_FFT_size=round(N/alfa); % Tamaño del bloque IFFT/FFT

```

b) Transmisor SEFDM

```

% Vector que almacena los símbolos totales
total_simb=[];

for j=1:1:total

% Generación de bits aleatorios
tx_bit s=randi([0 1],1,N*m);
% Conversion serie a paralelo
tx_bit p=reshape(tx_bit s,N*m,1);
% Modulación digital
tx_modul=qammod(tx_bit_p,M,'InputType','bit');
% Definición del bloque IFFT
ceros=zeros(round((IFFT_FFT_size-N)/2),1);
tx_sefdm=[ceros;tx_modul;ceros];
tx_sefdm=ifftshift(tx_sefdm);
% Zero-Padding con J=2
relleno=zeros((length(tx_sefdm)*(J-1)),1);
tx_sefdm1=[tx_sefdm(1:length(tx_sefdm)/2);relleno;tx_sefdm((length(tx_sefdm)/2)+1:length(tx_sefdm))];
% IFFT
tx_ifft=ifft(tx_sefdm1,IFFT_FFT_size*J);
% Conversion paralelo a serie
tx_out_s=reshape(tx_ifft,1,length(tx_ifft));

```



```
% Vector con símbolos totales
total simb=[total simb tx out s];
end
end
```

ORDEN DE EMPASTADO