

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

COMPARACIÓN DE LAS FORMAS DE ONDA FBMC Y UPMC BASADAS EN OFDM A TRAVÉS DE SIMULACIONES EN MATLAB

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

LORENA ALEXANDRA CHAMBA MASABANDA

lorena.chamba@epn.edu.ec

DIRECTOR: ING. JORGE EDUARDO CARVAJAL RODRÍGUEZ, MSc.

jorge.carvajal@epn.edu.ec

Quito, enero 2019

AVAL

Certifico que el presente trabajo fue desarrollado por Lorena Alexandra Chamba Masabanda, bajo mi supervisión.

ING. JORGE EDUARDO CARVAJAL RODRÍGUEZ, MSc.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Lorena Alexandra Chamba Masabanda, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada 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 cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

LORENA ALEXANDRA CHAMBA MASABANDA

DEDICATORIA

Con mucho cariño dedico este trabajo a mis padres, Carlos y Fabiola, quienes confiaron en mí a pesar de todas las adversidades que se presentaron. Hicieron de mí una mujer decidida y responsable, dispuesta a entregar el esfuerzo necesario hasta alcanzar mis ideales. Con este proyecto culminó una etapa de mi vida que la iniciamos juntos, a pesar de la distancia ustedes siempre estuvieron pendiente y cuidando de mí. Considero que fue un sacrificio compartido y eso es lo que me motivó a culminar con éxito esta etapa de mi vida.

A mi hermana Mayra, mi ángel aquí en la tierra, sé que tus oraciones fueron escuchadas, tú más que nadie formas parte de este proyecto. Cada consejo, cada corrección, sé que lo hiciste con amor. Fuiste la persona que supo corregirme con cariño, cuando los ánimos de continuar faltaban.

Todo este trabajo ha sido gracias a ellos.

Lorena A. Chamba M.

AGRADECIMIENTO

Agradezco a Dios, por regalarme el don de la vida, por estar conmigo en cada paso que doy y por brindarme salud para poder alcanzar mis objetivos. Tengo un corazón agradecido por permitirme coincidir con las personas que fueron mi soporte y compañía durante todo el período de estudio. Porque me regala la oportunidad de disfrutar junto a las personas que sé que me aman, y a las que sé que más amo en la vida.

Un eterno agradecimiento a Carlos y Fabiola, mis padres, quienes son mi pilar en todo lo que soy, en mi formación académica y de la vida, por su incondicional apoyo a través del tiempo. Gracias por ser los principales promotores de mis sueños, por apoyarme en todo momento, por creer en mí y en mis expectativas, gracias por sus consejos, sus valores, pero más que nada por su amor. El amor recibido, la paciencia con la que cada día se preocupaban por el avance y desarrollo de esta tesis es único y digno de agradecer.

A mi universidad, a la Escuela Politécnica Nacional, porque me permitió formarme en ella, al personal docente y administrativo que formaron parte de esta etapa universitaria, por todos sus aportes, que el día de hoy se ve reflejado en la culminación de mi paso por la universidad.

A mi director de tesis, al MSc. Jorge Carvajal por su apoyo y motivación ofrecidos durante la elaboración de esta tesis, expreso mi agradecimiento a usted por cada detalle y momento dedicado para aclarar las dudas que surgieron durante la elaboración de esta tesis.

Al Ing. Alex Caza por brindarme la ayuda necesaria en la elaboración de esta tesis, los conocimientos compartidos ayudaron en el proceso y culminación de este proyecto.

A mis amigos, quienes compartieron conmigo esta etapa de formación profesional, ustedes se convirtieron en mi familia quienes me ayudaron y apoyaron, gracias por su amistad incondicional.

Lorena A. Chamba M.

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
RESUMEN	IX
ABSTRACT	X
1 INTRODUCCIÓN	1
1.1 Objetivos	2
1.2 Alcance	2
1.3 Marco Teórico	3
1.3.1 Principios de OFDM	3
1.3.1.1 DFT Discrete Fourier Transform	5
1.3.1.2 Transformada Rápida de Fourier FFT	6
1.3.2 Propiedades de OFDM	6
1.3.2.1 Partición del Canal	6
1.3.2.2 Principio de Ortogonalidad	8
1.3.2.3 Prefijo Cíclico e Intervalo de Guarda	9
1.3.3 Transmisor OFDM discreto	11
1.3.3.1 Modulador	11
1.3.3.2 Conversor Serie/Paralelo	12
1.3.3.3 IFFT (Inverse Fast Fourier Transform)	12
1.3.3.4 Prefijo Cíclico	12
1.3.4 Receptor OFDM discreto	13
1.3.4.1 Supresión de PC y sincronismo	13
1.3.4.2 Conversión serie-paralelo y bloque FFT	13
1.3.4.3 Demodulador	14
1.3.5 Ventajas de OFDM	14
1.3.6 Desventajas de OFDM	14

1.3.7	FBMC.....	15
1.3.7.1	Principios FBMC.....	15
1.3.7.2	Ortogonalidad.....	16
1.3.7.3	Filtrado.....	16
1.3.8	Transmisor FBMC.....	16
1.3.8.1	FBMC en frecuencia.....	16
1.3.9	Receptor FBMC.....	23
1.3.9.1	Demodulador FBMC.....	23
1.3.10	Ventajas FBMC.....	26
1.3.11	Desventajas FBMC.....	26
1.3.12	UFMC.....	27
1.3.12.1	Principios UFMC.....	27
1.3.13	Transmisor UFMC.....	31
1.3.13.1	Modulación M-QAM.....	32
1.3.13.2	Conversión Serie/Paralelo.....	33
1.3.13.3	División de subbandas (D).....	33
1.3.13.4	IFFT.....	33
1.3.13.5	Filtrado por Subbanda. (F).....	33
1.3.13.6	Suma de subbandas y Conversor Paralelo/Serie.....	35
1.3.14	Receptor UFMC.....	36
1.3.14.1	Convertidor Serie/ Paralelo.....	36
1.3.14.2	2N-FFT.....	36
1.3.14.3	Ecuador de Símbolos.....	36
1.3.14.4	Convertidor Paralelo/Serie.....	36
1.3.14.5	Demodulador M-QAM.....	37
1.3.15	Ventajas de UFMC.....	37
1.3.16	Desventajas de UFMC.....	37
2	METODOLOGÍA.....	38
2.1	Implementación de la clase OFDM.....	39
2.1.1	Propiedades de la clase OFDM.....	39
2.1.2	Métodos de la clase OFDM.....	40
2.1.2.1	Método ConfigParamDependientes.....	41
2.1.2.2	Método Modulacion.....	42

2.1.2.3	Método Demodulacion	44
2.2	Implementación de la clase FBMC	47
2.2.1	Propiedades de la clase FBMC	47
2.2.2	Métodos de la clase FBMC	47
2.2.2.1	Método ConfigParamDependientes	48
2.2.2.2	Método Modulacion	49
2.2.2.3	Método Demodulacion	56
2.3	Implementación de la clase UFMC	61
2.3.1	Propiedades de la clase UFMC	61
2.3.2	Métodos de la clase UFMC	62
2.3.2.1	Método ConfigParamDependientes	63
2.3.2.2	Método Modulacion	63
2.3.2.3	Método Demodulacion	66
2.4	Densidad espectral de potencia	69
2.5	BER vs SNR	72
3	RESULTADOS Y DISCUSIÓN	74
3.1	Configuración de parámetros	74
3.1.1	Parámetros configurables en OFDM	75
3.1.2	Parámetros configurables en FBMC	75
3.1.3	Parámetros configurables en UFMC	75
3.2	Escenario de Pruebas para PSD	76
3.2.1	PSD para 64 subportadoras	76
3.2.2	PSD para 72 subportadoras	79
3.2.3	PSD para 128 subportadoras	82
3.2.4	PSD para 256 Subportadoras	84
3.3	Eficiencia espectral	87
3.4	Escenarios de pruebas para BER vs SNR	88
3.4.1	BER vs SNR para 64 Subportadoras	89
3.4.2	BER vs SNR para 128 subportadoras	90
3.4.3	BER vs SNR para 256 subportadoras	91
3.4.4	BER vs SNR para 1024 subportadoras	91
4	CONCLUSIONES Y RECOMENDACIONES	93
4.1	CONCLUSIONES	93

4.2	RECOMENDACIONES.....	94
5	REFERENCIAS BIBLIOGRÁFICAS.....	95
6	ANEXOS	98
	ANEXO I	99
	ANEXO II	101
	ANEXO III	105
	ANEXO IV	108
	ANEXO V	114
	ORDEN DE EMPASTADO.....	119

RESUMEN

Las técnicas de transmisión multiportadora han sido las más adoptadas para el desarrollo de los sistemas de comunicación inalámbricos. Estos sistemas se caracterizan por su amplia gama de usos y requiere una asignación flexible de recursos tiempo-frecuencia. Hoy en día, muchos estándares de comunicaciones utilizan en los sistemas de acceso al medio la técnica de modulación OFDM (*Orthogonal Frequency Division Multiplexing*), sin embargo, debido al prefijo cíclico y bandas de guarda, este sistema es ineficiente. Por esta razón, se presentan soluciones que ayudan a superar estos inconvenientes.

En este proyecto, se estudiará las desventajas de OFDM y se mostrará dos sistemas de modulación multiportadora como soluciones más efectivas. La primera corresponde a la técnica basada en banco de filtros FBMC (*Filter Bank Multi Carrier*) y la segunda a la técnica UFMC (*Universal Filtered Multi Carrier*) que aplica un filtrado a un grupo de subportadoras para minimizar la ICI entre estas y reducir los niveles de emisiones fuera de banda. La implementación de estas técnicas requiere más complejidad en el procesado, sin embargo, garantiza que UFMC sea más eficiente cuando se transmiten ráfagas pequeñas de datos, y que FBMC sea más eficiente cuando transmiten secuencias largas.

Se realizará un análisis comparativo de FBMC y UFMC con OFDM mediante la implementación de un sistema de transmisión en el software Matlab, incluyendo transmisor, receptor y canal AWGN. El análisis se realizará en función de las densidades espectrales de potencia y BER vs SNR.

PALABRAS CLAVE: OFDM, FBMC, UFMC, PSD, BER, filtro.

ABSTRACT

Multi-carrier transmission techniques have been the most adopted for the development of wireless communication systems. These systems are characterized by their wide range of uses and require a flexible allocation of time-frequency resources. Nowadays, many communication standards use the OFDM (Orthogonal Frequency Division Multiplexing) modulation technique in the access systems to the medium, however, due to the cyclic prefix and guard bands, this system is inefficient. For this reason, solutions are presented that help overcome these drawbacks.

In this project, the disadvantages of OFDM will be studied and two multi-carrier modulation systems will be shown as more effective solutions. The first corresponds to the FBMC filter bank (Filter Bank Multi Carrier) technique and the second to the UFMC (Universal Filtered Multi Carrier) technique that applies a filtering to a group of subcarriers to minimize the ICI between them and reduce the out-of-band emission levels. The implementation of these techniques requires more complexity in processing, however, it ensures that UFMC is more efficient when transmitting small bursts of data, and that FBMC is more efficient when transmitting long sequences.

A comparative analysis of FBMC and UFMC with OFDM will be carried out through the implementation of a transmission system in the Matlab software, including transmitter, receiver and AWGN channel. The analysis will be made according to the power spectral densities and BER vs SNR.

KEYWORDS: OFDM, FBMC, UFMC, PSD, BER, filter.

1 INTRODUCCIÓN

Para el envío de información en comunicaciones se utilizan señales que deben ser procesadas con el fin de transmitir las de un lugar a otro. Comúnmente en la transmisión de señales vía inalámbrica es necesario realizar un proceso de modulación, existen dos tipos de modulaciones; por un lado están las modulaciones mono portadora o SC (*Single Carrier*) la cual usa una sola portadora para enviar la información entre emisor y receptor, por otro lado se encuentran las modulaciones multiportadora o MC (*Multi Carrier*), que dividen la información a transmitir en un número determinado de subportadoras que posteriormente se envían a través de diferentes subcanales, estas últimas presentan una mejora en el aprovechamiento del canal [1].

A diferencia de FDM (*Frequency Division Multiplexing*) que usa bandas de guarda entre portadoras, OFDM (*Orthogonal Frequency Division Multiplexing*) presenta una mejor eficiencia espectral ya que elimina las bandas de guarda entre subportadoras y trabaja con el solapamiento de canales gracias a que las subportadoras son ortogonales. OFDM, es una técnica de modulación multiportadora, adoptado ampliamente en los sistemas de acceso al medio en muchos estándares de comunicaciones. Sin embargo, esta modulación presenta ciertas desventajas como el uso de prefijo cíclico, puesto que al ser una copia de los últimos datos al inicio del símbolo OFDM, se considera una redundancia que disminuye el rendimiento de la transmisión.

Por otro lado, frente al agotamiento de los recursos físicos que están adquiriendo las redes actuales, la escasez de espectro es un problema que tienen que afrontar los sistemas inalámbricos debido a la variedad de usos que estos presentan, por esta razón se debería contar con soluciones destinadas al incremento de eficiencia espectral mediante una asignación flexible de recursos tiempo-frecuencia con el fin sobrellevar eficientemente la variedad de servicios.

En esta perspectiva, existen algunas soluciones para superar estos inconvenientes, considerando el uso de banco de filtros como una alternativa para mejorar el rendimiento de la transmisión de datos. FBMC (*Filter Bank Multi Carrier*) es un sistema de modulación multiportadora que ofrece mayor eficiencia en el ancho de banda, puesto que divide el espectro en múltiples subportadoras para aplicar un filtrado a cada una de ellas con el fin de reducir las emisiones fuera de banda OoB (*Out of Band*) que se presenta en OFDM, de este modo, el problema de interferencia entre subportadoras disminuye en comparación con OFDM.

Asímismo, considerando que debido al filtrado individual de subportadoras se presenta una respuesta impulsiva larga, se podría pensar en filtrar todas las subportadoras para reducir la respuesta impulsiva, a esta técnica se la conoce como F-OFDM. Una solución a esta última con el fin de mejorar la eficiencia en la transmisión es UFMC (*Universal Filtered Multi Carrier*), que es un sistema de modulación que realiza un filtrado no en una, ni en todas las subportadoras, sino que divide el ancho de banda en subbandas, con un determinado número de subportadoras cada una, así se obtiene una mejor respuesta impulsiva.

1.1 Objetivos

El objetivo general de este Proyecto Técnico es: Comparar mediante simulación las formas de onda, FBMC (*Filter Bank Multi Carrier*), UFMC (*Universal Filtered Multi Carrier*) con OFDM (*Orthogonal Frequency Division Multiplexing*)

Los objetivos específicos de este Proyecto Técnico son:

- a) Describir las características, parámetros y componentes principales de FBMC, UFMC y OFDM.
- b) Simular en MatLab FBMC, UFMC, OFDM en un canal AWGN.
- c) Comparar el desempeño de las formas de onda en términos de BER y eficiencia espectral.

1.2 Alcance

De manera general en la primera parte de este proyecto se introducirá una parte teórica de las características y componentes que conformarían un sistema OFDM, FBMC, UFMC, encaminado a obtener ventajas y desventajas respectivamente.

El presente proyecto va orientado a la simulación *end to end* de una sistema OFDM, FBMC y UFMC, en los cuales se implementarán la parte del transmisor y receptor, se utilizará un canal AWGN que consiste en añadir ruido blanco aditivo gaussiano a la señal transmitida.

Para las simulaciones se usará Matlab y se hará uso del paquete *Communications System Toolbox* ya que permite realizar tareas como modulaciones, calcular tasas de error y visualizar espectros con funciones ya optimizadas y de fácil manejo.

En el caso del sistema OFDM, el objetivo será modular los bits que se proporcionen de forma aleatoria. En esta parte se necesitará parámetros de entrada como número de

portadoras a crear (de datos y de guarda), cantidad de bits por símbolo (16-QAM) y la cantidad de símbolos OFDM totales a crear.

Es importante mencionar que la configuración de los parámetros que definirán el resultado de la simulación, se irá adaptando según se necesite para comparar con FBMC y UFMC. El tamaño del prefijo cíclico dependerá si la transmisión es empleada para su comparación con FBMC o UFMC, ya que tendrá el mismo tamaño que empleen las otras modulaciones (en UFMC corresponde a la longitud del filtro).

Para FBMC, primero se declarará las variables que determinarán el resultado de las simulaciones, se escogerán valores similares a los empleados en LTE con la finalidad de poder comparar con OFDM y UFMC. Se compararán las diferentes densidades espectrales de FBMC con diferentes valores de factor de solapamiento (2,3,4); para estas simulaciones se mantendrá siempre un número de portadoras de guarda mayor a 1 para obtener la eficiencia espectral necesaria y estos resultados se compararán con los resultados de la densidad espectral de potencia obtenida para OFDM.

Para UFMC el proceso es muy parecido a OFDM excepto por algunas diferencias, como el filtrado. Se declararán las variables que definirán la naturaleza de la forma de onda, esta es una característica altamente configurable y los parámetros de entrada serán elegidos de forma que se pueda comparar con estándares existentes que emplean OFDM, el tamaño del filtro estará relacionado con el de un tamaño típico de prefijo cíclico, el filtro será creado mediante los coeficientes temporales de tipo *Chebyshev*, al final se sumarán las salidas del filtro para obtener la señal completa formada por todas las bandas.

Para obtener resultados de BER vs SNR, se empleará el siguiente método, una vez obtenida la señal, la SNR del canal será conocida y se irá añadiendo cierto nivel de ruido AWGN hasta tener el ruido deseado y así poder calcular el error, el rango de simulación será desde 5 dB a 30 dB de SNR, con un paso de 0.5 dB.

Para realizar las simulaciones, el receptor captará un total de 200 símbolos multiportadora procedentes de cada modulación, una vez añadido ruido y calculado BER, se procederá a repetir la operación, para finalmente realizar un promedio.

1.3 Marco Teórico

1.3.1 Principios de OFDM

OFDM (*Orthogonal Frequency Division Multiplexing*), es una técnica de modulación multiportadora (*multicarrier*), adoptada ampliamente en muchos estándares de comunicaciones

como: LTE (*Long Term Evolution*), en los sistemas móviles de cuarta generación 4G, IEEE 802.11 para redes de área local inalámbrica WLAN (*Wireless Local Area Network*), IEEE 802.16, televisión digital terrestre en norma europea DVB-T (*Digital Video Broadcasting Terrestrial*), transmisión de datos a través de la tecnología WiMAX (*Worldwide Interoperability for Microwave Access*). OFDM surgió con el propósito de simplificar problemas relacionados con multitrayecto y alcanzar velocidades altas [1].

El principio fundamental de la modulación OFDM es la partición del canal, que consiste en dividir el ancho de banda del sistema en subcanales paralelos entre sí. Asimismo se divide el flujo de datos a transmitir en N subportadoras ortogonales. OFDM, es robusta frente a efectos de propagación como la dispersión y multitrayecto. El multitrayecto en el canal provoca que, al receptor lleguen múltiples réplicas de la señal transmitida con diferentes amplitudes y retardos, provocando un ensanchamiento temporal de la señal, y solapamientos entre símbolos adyacentes produciendo interferencia entre símbolos ISI (*Intersymbol Interference*) [1].

Si se observa una señal OFDM en el dominio del tiempo como la que se muestra en la Figura 1.1 [2] se distingue que dentro del período de la subportadora más baja f_0 , alcanzan períodos de otras subportadoras. Esto genera en la representación espectral que el máximo de cada subportadora coincida en el nulo de las demás [2].

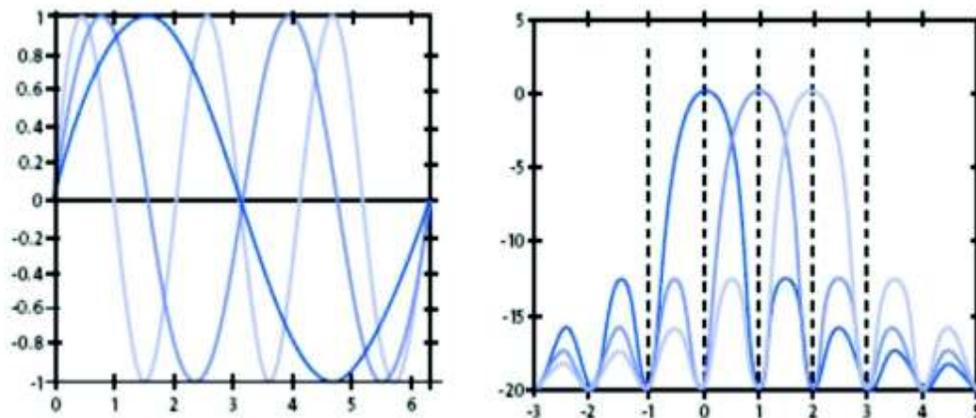


Figura 1.1. Representación temporal y frecuencial de la señal OFDM.

Inicialmente la implementación de un modulador OFDM requería de un conjunto de generadores de subportadoras sinusoidales conocido también como banco de osciladores, en el receptor se requería el mismo número de demoduladores, de ahí que, para un número alto de subportadoras el sistema se consideraba complejo e ineficiente. Debido al problema técnico que supone la generación y detección de un gran número de subportadoras en OFDM, los procesos de modulación y demodulación se realizan en tiempo discreto

mediante la IDFT (*Inverse Discrete Fourier Transform*) y la DFT (*Discrete Fourier Transform*) respectivamente [3].

1.3.1.1 DFT Discrete Fourier Transform [4]

Partiendo del concepto de la Transformada de Fourier y su inversa, se tiene que, para una señal continua $x(t)$ establecida en el dominio de tiempo, la transformada de Fourier constituye la relación con la misma señal representada en el dominio de la frecuencia.

Entonces, la transformada de Fourier y su inversa en los sistemas OFDM permite convertir señales del dominio del tiempo al dominio de la frecuencia y viceversa. Las ecuaciones que definen la Transformada de Fourier y la transformada inversa de Fourier, se presentan en la ecuación 1.1 [4] y ecuación 1.2 [4] respectivamente.

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt$$

Ecuación 1.1. Transformada de Fourier.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t} dt$$

Ecuación 1.2. Transformada Inversa de Fourier.

Se debe recordar que la relación entre la frecuencia en radianes con la frecuencia en *Hz* viene dada por $\omega = 2\pi f$.

Debido a la dificultad para generar y construir las señales en tiempo continuo por los infinitos valores con los que se debe trabajar, se implementa la transformada discreta de Fourier que requiere de un número de muestras finito $[N]$. En la ecuación 1.3 [4] y ecuación 1.4 [4] se presenta la ecuaciones que representan la transformada discreta y la transformada discreta inversa de Fourier respectivamente.

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-\frac{2\pi nk}{N}}, \quad k = 0, 1, 2, \dots, N - 1$$

Ecuación 1.3. Transformada discreta de Fourier.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{\frac{2\pi nk}{N}}, \quad n = 0, 1, 2, \dots, N - 1$$

Ecuación 1.4. Transformada inversa discreta de Fourier IDFT.

En base a las expresiones 1.3 y 1.4 se observa que el cálculo de la DFT se realiza mediante sucesivas multiplicaciones y sumas, esto genera gastos de memoria que crece conforme aumenta el valor de muestras. Considerando la realización secuencial de N multiplicaciones, por cada valor de k , el esfuerzo computacional se incrementa en función N^2 que representaría el número total de multiplicaciones.

1.3.1.2 Transformada Rápida de Fourier FFT [5]

La FFT es un algoritmo que transforma una señal compleja muestreada en tiempo, en una señal compleja muestreada en frecuencia. Este algoritmo se usa para reducir el tiempo de cálculo de la DFT, como se explicó anteriormente, para un número $[N]$ subportadoras aplicando la DFT se tendría N^2 multiplicaciones complejas. Ahora, utilizando el algoritmo de la FFT el número de operaciones se reduce a un valor de $N(\log_2 N)$. Cabe mencionar que para la utilización de este algoritmo es necesario que $[N]$ sea potencia de 2, ya que el algoritmo no permite la entrada de valores de $[N]$ que no cumplan esta condición. En la Figura 1.2 [5] se presenta un esquema en el que se muestra la IFFT.

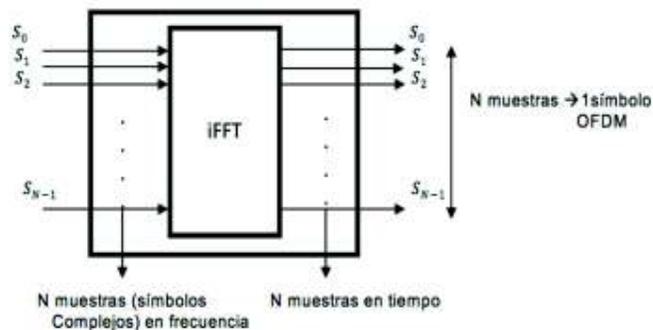


Figura 1.2. Representación de la IFFT.

1.3.2 Propiedades de OFDM.

OFDM habitualmente está formado por un conjunto de moduladores, cada modulador asociado a una frecuencia de portadora diferente, es decir, en el transmisor se combinan las salidas del modulador y se generan las señales que posteriormente serán transmitidas.

1.3.2.1 Partición del Canal

Consiste en dividir el ancho de banda del sistema $BW [Hz]$ en N subcanales paralelos , con una separación entre subportadoras Δf , como se expresa en la ecuación 1.5 [6].

$$\Delta f = \frac{BW}{N}$$

Ecuación 1.5. Separación entre subportadoras.

Donde:

Δf : separación entre subportadoras.

BW : ancho de banda del sistema.

N : número de subportadoras.

Teniendo en cuenta que las N subportadoras modulan la información por separado, entonces, cada subportadora estará asociada a una frecuencia f_k , como se muestra en la ecuación 1.6 [6].

$$f_k = f_0 + \frac{k}{T} = f_0 + \frac{BW}{N}k, \quad k = \{0, \dots, N - 1\}$$

Ecuación 1.6. Frecuencia asociada a cada subportadora.

Donde:

f_0 : frecuencia más baja.

T : período de símbolo

De esta manera, OFDM divide la información en un número determinado de subportadoras para transmitirla en diferentes subcanales centrados en diferentes frecuencias ortogonales f_k [7]. Debe señalarse que el número de subportadoras ha de asegurar que el período de símbolo de cada subportadora sea mucho mayor que el “*delay spread*¹” del canal, es decir, el ancho de banda de cada subportadora debe ser menor que el ancho de banda de coherencia del canal B_c ², esto se considera para que las subportadoras no sufran excesiva interferencia entre símbolos ISI [3].

En la ecuación 1.5, si la separación entre subportadoras Δf es menor que el ancho de banda de coherencia del canal, los desvanecimientos selectivos en frecuencia pueden considerarse prácticamente planos en la banda de cada subportadora. De esta manera, los desvanecimientos selectivos en frecuencia afecta sólo al flujo de bits que transporta cada subportadora [7] [8].

¹ Delay Spread: dispersión del retardo se interpreta como la diferencia entre el tiempo de llegada del primer componente multitrayecto y el tiempo de llegada de los últimos componentes multitrayecto [3].

² Ancho de banda de coherencia corresponde al rango de frecuencias mínima en el que la respuesta del canal puede considerarse plana [3].

1.3.2.2 Principio de Ortogonalidad

La ortogonalidad optimiza la eficiencia del espectro a través del solapamiento de las subportadoras, entonces, una de las características principales que diferencia OFDM de otros métodos de multiplexación en frecuencia es la ortogonalidad entre las subportadoras que llevan la información, es decir, se debe conseguir que la separación espectral entre subportadoras consecutivas sea siempre la misma e igual al inverso del período de símbolo, esto se presenta en la ecuación 1.7 [5].

$$T = \frac{1}{\Delta f} = \frac{N}{BW} [s]$$

Ecuación 1.7. Período de Símbolo.

De esta manera, la ortogonalidad que existe entre subportadoras permite una transmisión simultánea sin interferencia, alcanzando un mejor aprovechamiento del espectro. Igualmente, la ortogonalidad facilita ver al canal de banda ancha como un conjunto de múltiples subcanales paralelos de banda estrecha de baja velocidad, puesto que la información que tiene cada subcanal es menor en comparación a la información total que se va transmitir, es decir, el ancho de banda de cada subcanal será menor que el ancho de banda del sistema [5].

De esta forma, podemos transformar canales selectivos en frecuencia en varios subcanales con respuesta en frecuencia plana, el espectro de cada subportadora está solapado con los espectros de las subportadoras adyacentes. En las Figura 1.3 y Figura 1.4 [5] se puede observar los espectros que producirían un conjunto de canales paralelos. Se observa que es posible asignar subportadoras a una misma señal para transmitirla y que en recepción las bandas laterales se solapen sin producir interferencia entre subportadoras adyacentes debido al principio de ortogonalidad.

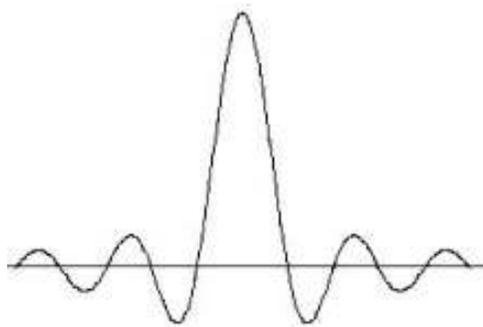


Figura 1.3. Espectro OFDM de un solo canal.

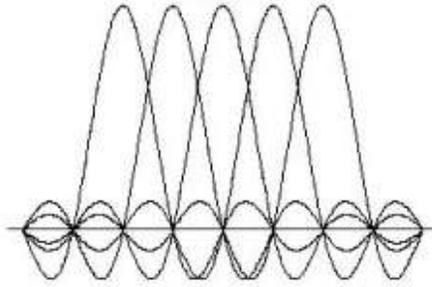


Figura 1.4. Espectro OFDM de 5 subcanales.

En la figura 1.5 [3] se muestra la diferencia existente entre los espectros de una transmisión usando una modulación con portadoras ortogonales con una transmisión en FDM.

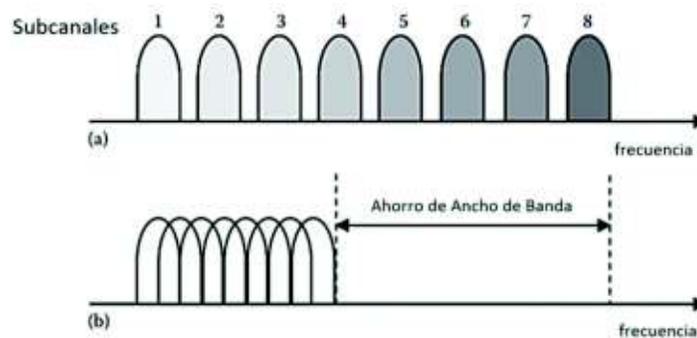


Figura 1.5. Representación de canales a) FDM b) OFDM.

1.3.2.3 Prefijo Cíclico e Intervalo de Guarda [3] [9].

Una de las mayores ventajas de OFDM es su robustez frente a los efectos del multitrayecto en el canal de propagación. Con la finalidad de prevenir la ISI (*Inter Symbol Interference*) la mayoría de aplicaciones que usan OFDM insertan una banda de guarda entre los símbolos OFDM adyacentes. En efecto, la duración de esta banda de guarda se elige considerando que sea mayor que el máximo retardo entre todas las subportadoras, consiguiendo de esta manera que las componentes retardadas de un símbolo no interfieran con el siguiente.

La banda de guarda se implementa mediante la técnica de Zero Padding, puesto que, durante el intervalo de guarda, el transmisor envía una serie de ceros (subportadoras null), alcanzando una señal más definida en el dominio frecuencial debido al aumento de muestras al espectro de la señal.

Sin embargo, mientras se controla la ISI con el intervalo de guarda, efectos como, la desalineación entre el transmisor y receptor, o la movilidad del receptor con respecto al

transmisor (*efecto Doppler*³), producirían cierta interferencia entre subportadoras ICI (*Inter Carrier Interference*), pudiendo perder la ortogonalidad entre las mismas.

En la Figura 1.6 [3] se puede observar la representación de ISI e ICI.

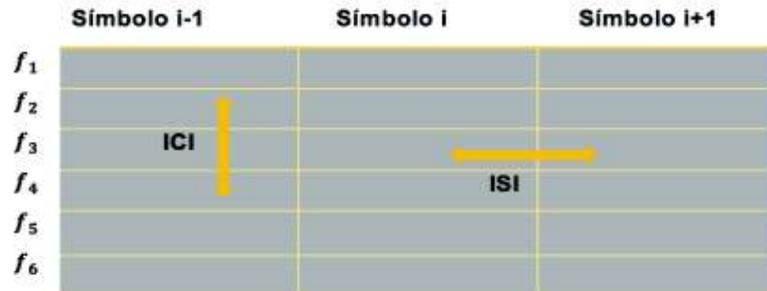


Figura 1.6. Representación básica de ISI e ICI.

Una solución para este efecto es que en el transmisor se extienda el símbolo OFDM en el intervalo de guarda, en consecuencia cualquier réplica que llegue directa o retrasada de la señal transmitida se convertirá en periódica, asegurando así la ortogonalidad de las subportadoras. Esto ocurre siempre y cuando se cumpla que el retardo sea menor al intervalo de guarda, por esta razón, en lugar de un intervalo de guarda con subportadoras null, se prefieren implementar transmisiones con extensión cíclica.

Esta extensión cíclica toma el nombre de prefijo cíclico CP (*Cyclic Prefix*) y se construye copiando la parte final del símbolo OFDM al inicio del mismo, entonces, al final se tendría un símbolo OFDM formado tanto por el prefijo cíclico como por el símbolo OFDM generado por la IDFT, de forma contraria ocurre en recepción, el prefijo cíclico se elimina antes de realizar la DFT. La duración total de símbolo OFDM se indica en la ecuación 1.8 [10]

$$T_{OFDM} = T_U + T_G$$

Ecuación 1.8. Período de símbolo final OFDM.

En la Figura 1.7 [10] se observa la representación gráfica del símbolo OFDM antes y después de insertar prefijo cíclico.

³ Movimiento relativo entre emisor y receptor, produce dispersión frecuencial, a medida que emisor-receptor se acercan la frecuencia de la señal recibida aumenta, y si se alejan la frecuencia de la señal recibida disminuye [4].

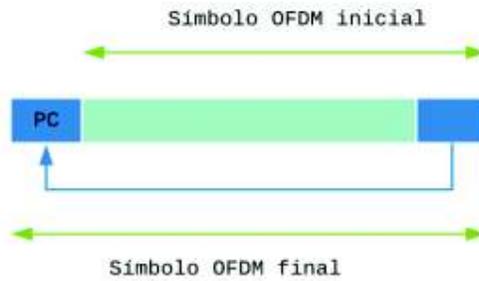


Figura 1.7. Símbolo OFDM con CP.

Aunque este prefijo es necesario cuando el canal es dispersivo en tiempo, no transporta información por lo que se reduce el uso eficiente de recursos disponibles del sistema. Sin embargo, la pérdida de eficiencia por motivo del CP puede verse reducida si se aumenta la duración del símbolo OFDM.

Cabe recalcar que no puede elegirse un tiempo de símbolo T excesivamente grande, es decir, se debe escoger la duración del CP como el mínimo valor posible que permite resistir la dispersión del canal. De este modo, debido a la utilización del CP un canal con dispersión en tiempo y por lo tanto selectivo en frecuencia (banda ancha), se convierte en un conjunto de N subcanales paralelos, no selectivos en frecuencia (banda estrecha).

1.3.3 Transmisor OFDM discreto

Un transmisor OFDM en banda base actualmente se implementa como se muestra en el diagrama de bloques de Figura 1.8 [3].

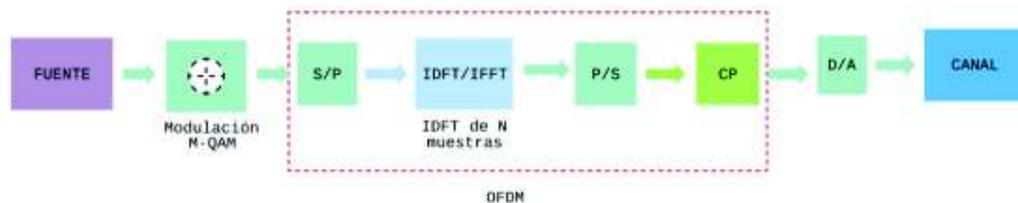


Figura 1.8. Transmisor OFDM.

Inicialmente, la secuencia de datos a la entrada proviene de una fuente que puede ser de diferentes tipos como texto, multimedia, entre otros. Considerando que los N datos a transmitir son $x_k, k = 0, 1, 2, \dots, N - 1$, implica que cada x_k esté asociado a una frecuencia portadora f_k .

1.3.3.1 Modulador

Estos datos están representados mediante un número complejo que proviene de una

constelación que puede ser QPSK (*Quadrature Phase Shift Keying*), 16-QAM (*Quadrature Amplitud Modulation*), 64-QAM entre otras, estas modulaciones se escogen dependiendo de las aplicaciones para las que se use y la velocidad que se desee alcanzar.

1.3.3.2 Conversor Serie/Paralelo

La cadena de números complejos ingresa a un convertidor serie a paralelo S/P con el fin de ubicar el conjunto de N símbolos complejos en paralelo y cada uno de estos símbolos corresponde a una de las subportadoras. Los símbolos a la salida de este convertidor representan los componentes discretos en frecuencia del transmisor OFDM.

1.3.3.3 IFFT (Inverse Fast Fourier Transform)

Para convertir los componentes frecuenciales discretos en muestras temporales se aplica la inversa de la DFT a los N símbolos, implementada a través del algoritmo IFFT, como resultado se obtiene una secuencia de muestras en paralelo de longitud N , cada muestra contiene la información de las subportadoras que entraron.

1.3.3.4 Prefijo Cíclico

Después que las muestras temporales resultantes ingresan a un convertidor paralelo a serie, se copia las últimas muestras al inicio del símbolo, prefijo cíclico CP, y se forma el símbolo OFDM [4]. Esto se realiza para recuperar el símbolo completo en recepción, puesto que, al repetir la parte final (más afectada por interferencias) al principio, el receptor conocería el inicio y el final replicado y resultaría más fácil recuperar el símbolo original [6].

El tamaño del prefijo cíclico depende de la aplicación que se use, siendo algunos valores típicos de $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ de la duración del símbolo. Algunos estándares que usan prefijo cíclico en la Tabla 1.1 [2] [11] se indican algunos valores típicos.

Tabla 1.1. Duración del prefijo cíclico para algunos estándares.

Estándar	N_{FFT}	N_{CP}	N_{FFT}/N_{CP}
IEEE 802.11 a/g	64	16	$\frac{1}{4}$
IEEE 802.11n/ac	128	32,16	$\frac{1}{4}, \frac{1}{8}$
DVB-T	2048,4096	64,128,256,512	$\frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}$

Los símbolos pasan por un convertidor digital-analógico D/A, de este modo, se obtiene la señal OFDM en banda base. Habitualmente la señal antes de enviar al canal discreto en tiempo se traslada a una frecuencia de portadora dada a la que se transmitirá.

Finalmente la señal que será enviada a la salida del transmisor está expuesta a efectos del canal, en la que su propagación se ve afectada por fenómenos como multirrayecto, difracción.

1.3.4 Receptor OFDM discreto

La señal que llega a recepción debe ser tratada con el proceso inverso. El bloque más importante en recepción corresponde a la DFT implementado a través de la FFT, pues permite recuperar las componentes frecuenciales situadas en las mismas frecuencias ortogonales que fueron utilizadas en transmisión. En la Figura 1.9 [3] se presenta el diagrama de bloques de un receptor OFDM.



Figura 1.9. Receptor OFDM.

Una vez que la señal transmitida ha pasado a través del canal se ve afectada por ruido, lo primero que se hace es trasladar nuevamente la señal a banda base, mediante un demodulador seguido de un filtro paso bajo LPF (*Low Pass Filter*) con el fin de eliminar las componentes fuera de la banda de paso [3].

1.3.4.1 Supresión de PC y sincronismo

La señal pasará por el convertidor analógico-digital A/D, y pasa al siguiente bloque donde se realiza un proceso de sincronización que consiste en recuperar el sincronismo del sistema en tiempo y frecuencia. De la misma manera, en este bloque se elimina el prefijo cíclico.

1.3.4.2 Conversión serie-paralelo y bloque FFT

Después de la supresión del prefijo cíclico se obtiene las N muestras temporales, estas muestras temporales ingresan a un convertidor serie a paralelo y a su vez se las muestrea en frecuencia mediante el bloque DFT utilizando el algoritmo FFT. La salida del bloque FFT

pasa por el convertidor paralelo-serie y finalmente esta señal en serie pasa a través de un demodulador para recuperar la señal original.

1.3.4.3 Demodulador

Al bloque del demodulador ingresan los flujos de símbolos en serie para finalmente recuperar los datos originales dependiendo del modulador que se haya usado en transmisión.

A continuación se presenta los puntos fuertes y débiles de un sistema OFDM.

1.3.5 Ventajas de OFDM

Entre las primeras ventajas se puede citar, la simplicidad de construcción del sistema, debido a que ningún elemento es costoso computacionalmente, es decir, tanto la IFFT y FFT no requieren elementos computacionales excesivos y se los puede adquirir en un procesador pequeño.

Asimismo, gracias a la ortogonalidad que existe entre las subportadoras se elimina la interferencia entre símbolos. Esta ortogonalidad hace que los valores de frecuencia de cada subportadora en los que se envía la señal coincida con los ceros de las demás.

En entornos con multitrayecto o desvanecimiento se utiliza el CP, que copia la parte final del símbolo OFDM al inicio del mismo, nótese, que al expandir el símbolo en tiempo, logra disminuir la probabilidad que se altere un símbolo por desvanecimiento.

La flexibilidad es otra ventaja que conviene mencionar, ya que OFDM permite tratar de forma independiente las diferentes subportadoras, esto significa que cada subportadora puede llevar distinto número de bits. Asimismo se puede distinguir las subportadoras que resisten mejor los efectos del canal.

En cuanto a eficiencia espectral, se conoce que, debido al multitrayecto los entornos son selectivos en frecuencia, entonces, al modular las subportadoras en OFDM se consigue una respuesta en frecuencia casi plana, de esta manera se prescinde de los problemas que presentan los entornos selectivos en frecuencia y en algunos casos se puede evitar ecualizaciones.

1.3.6 Desventajas de OFDM

El efecto Doppler que se presenta en el canal afecta la ortogonalidad entre subportadoras, aumentando así la BER y, consecuentemente, disminuye el desempeño del sistema.

Si se tiene un número de subportadoras muy grande, los sistemas OFDM presentan PAPR (*Peak to Average power Ratio*), debido a que los amplificadores de potencia provocarán

distorsión dentro y fuera de banda. En otras palabras, si se presenta una relación alta entre la potencia pico y la potencia media de la señal que se transmite se genera desbalances en potencia entre cada subportadora, como consecuencia la señal transmitida se degrada.

Los sistemas OFDM presentan errores de sincronización, a causa de que algunas subportadoras no oscilan perfectamente ocasionando la pérdida de sincronismo en el sistema.

Las subportadoras presentan lóbulos laterales grandes, que son el resultado de la respuesta impulsiva rectangular en el dominio del tiempo, esto produce filtraciones de potencia hacia las bandas de otros usuarios, que causan interferencias indeseables.

Se debe trabajar con señalización y sincronización frecuencial, puesto que la primera facilita la decisión para conocer el inicio de cada símbolo OFDM y la segunda es indispensable para reducir la ICI.

OFDM utiliza un filtrado con respuesta impulsiva rectangular, esto genera una incorrecta localización de la señal en tiempo y en frecuencia, en otras palabras, son sensibles a desajustes en la temporización y el desplazamiento en frecuencia.

En estos sistemas es necesario tener varios tipos de redundancia para facilitar la recepción de tramas al receptor, lo que impide el envío de bits útiles o de información reduciendo la tasa binaria. Asimismo, se reduce la eficiencia espectral, debido a la presencia de bandas de guarda o prefijo cíclico.

1.3.7 FBMC

A pesar de ser OFDM una modulación multiportadora eficiente, se han creado soluciones que mejoran la eficiencia espectral de esta modulación. De esta manera, FBMC (*Filter Bank Multi Carrier*) se ha creado en base a OFDM, considerando que una de las soluciones frente al problema de interferencia entre subportadoras, es la de filtrar a cada una de estas por separado. Por esta razón, en esta sección se describe las características principales de la técnica FBMC [12].

1.3.7.1 Principios FBMC

De manera similar que en OFDM, FBMC divide el ancho de banda en una cantidad de subportadoras, al realizar un filtrado a cada una disminuye las componentes fuera de banda de cada subportadora, por lo que se puede prescindir de las bandas de guarda o prefijo cíclico. Del mismo modo FBMC mejora su respuesta a desplazamientos temporales y de frecuencia [13].

En FBMC la reducción de la radiación fuera de banda implica una mejor utilización del espectro, que equivale a una mayor eficiencia espectral. En FBMC cada subportadora es filtrada individualmente mediante un banco de filtros es decir, FBMC se diferencia de OFDM por el tipo de filtro que usa [14].

FBMC elimina el prefijo cíclico que se usa en OFDM para combatir la ISI, al eliminar el prefijo cíclico se mantiene el tamaño del símbolo multiportadora y se evita símbolos largos.

1.3.7.2 Ortogonalidad

Mientras en OFDM se debe garantizar la ortogonalidad para todas las subportadoras, en FBMC se requiere únicamente ortogonalidad entre subportadoras adyacentes. A fin de aprovechar al máximo el ancho de banda del canal, las subportadoras deben adaptarse a la modulación de amplitud de cuadratura O-QAM (*Offset Quadrature Amplitud Modulation*). En efecto, combinar un banco de filtros con modulación O-QAM permite optimizar el filtrado de las subportadoras para que los lóbulos laterales del filtro prototipo se atenúen, alcanzando una interferencia entre símbolos y entre subportadoras menor que en OFDM, sin necesidad de utilizar del prefijo cíclico [15].

1.3.7.3 Filtrado

Un símbolo FBMC se obtiene mediante el uso de bancos de filtros, pudiendo realizarse de dos maneras. La primera, realiza un procesamiento en frecuencia mediante la dispersión de frecuencia FS (*Frequency Spreading*) [16]. La segunda, realiza un procesamiento en tiempo, mediante una Red Polifase PPN-FFT (*Poly Phase Network-FFT*) [17]. Cabe señalar que aplicando cualquiera de los dos métodos se obtiene una respuesta similar, la diferencia entre cada uno de los métodos es la ubicación de la etapa de filtrado respecto a la IDFT.

1.3.8 Transmisor FBMC

A continuación se describirán las modificaciones que se realizan en un sistema OFDM para obtener un sistema FBMC, (en la Figura 1.8 se presenta el esquema que se implementa en OFDM).

1.3.8.1 FBMC en frecuencia

En un inicio, la implementación de un sistema FBMC estaba basado en una etapa de síntesis y otra de análisis. Estas etapas síntesis-análisis están formadas por filtros ubicados en transmisión y en recepción. En el transmisor los filtros descomponen la señal en diferentes componentes para cada subportadora, mientras que en el receptor, los filtros se usan para reconstruir la señal original partiendo de las componentes que se recibe de cada

subportadora. De ahí que, en transmisión el banco filtros reciben el nombre de síntesis, mientras que en recepción se denominan banco de filtros de análisis [18].

A continuación se explica el procedimiento para generar una señal FBMC, siguiendo el diagrama de bloques de la Figura 1.10 [17]. Este diagrama presenta el esquema para generar un símbolo FBMC a través de un procesamiento en el dominio de la frecuencia.

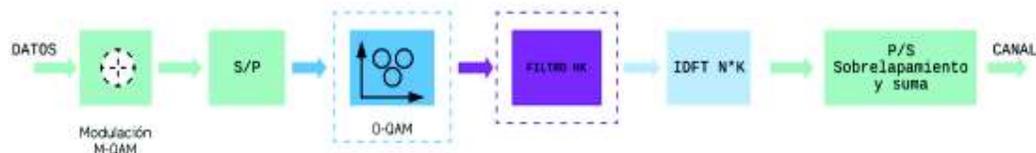


Figura 1.10. Transmisor FBMC mediante el procesamiento de frecuencia.

Partiendo del esquema presentado en la Figura 1.8, se observa en la Figura 1.10 que en un transmisor FBMC en frecuencia, se incorpora dos bloques consecutivos antes de la IDFT, correspondientes al de modulación O-QAM y al de Filtro HK respectivamente. Tomando esta consideración se procede a explicar la función que realiza cada bloque de la figura 1.10.

Modulación M-QAM

Siguiendo el mismo procedimiento que se usó en OFDM, los bits que provienen de una fuente de datos ingresan al primer bloque para ser modulados, la salida de este bloque será un número complejo de una constelación QPSK, 16-QAM, 64-QAM cuya elección depende de las aplicaciones para las que se use esta técnica.

Convertidor Serie/Paralelo

La salida del bloque anterior corresponde a una cadena de símbolos complejos en serie, en este bloque se convierte esta cadena en un conjunto de N símbolos complejos en paralelo, para que esta columna de símbolos complejos ingresen al bloque de O-QAM.

Modulación O-QAM

En FBMC puede usarse cualquier tipo de modulación siempre y cuando los subcanales estén separados, es decir, si solo se envía información en subcanales pares o impares; mientras se cumpla esta condición no existirá superposición [19]. Sin embargo, este proceso no es óptimo porque no se utilizarían eficientemente todos los recursos. Ahora bien, para utilizar todos los subcanales se necesita una modulación específica que ayude a sobrellevar la superposición entre subportadoras adyacentes .

Asimismo, manteniendo la idea de filtrado, ocurre que, al filtrar al pulso de transmisión producirá ensanchamiento del pulso en el dominio del tiempo, estos pulsos ensanchados se solapan perdiendo la ortogonalidad temporal entre ellos, ocasionando así interferencia entre símbolos (ISI) [20]. En consecuencia, antes de realizar el filtrado por subportadora es muy importante que los símbolos complejos ingresen al bloque O-QAM.

Es así que, en FBMC con el propósito de mantener la ortogonalidad entre subportadoras adyacentes tanto en el dominio del tiempo, como en el dominio de la frecuencia y de maximizar la eficiencia de transmisión, se usa la modulación QAM desplazada O-QAM. El procedimiento que realiza O-QAM para evitar la interferencia es el siguiente: se descomponen los símbolos complejos en parte real y en parte imaginaria desfasándolos la mitad de un período de símbolo entre ellos [21]. Así, la parte real modula subportadoras pares y la parte imaginaria modula subportadoras impares. En efecto, al tener un símbolo real y otro imaginario aunque los símbolos se solapen se mantendrá la ortogonalidad.

Nótese que al separar los símbolos en parte real y parte imaginaria, para transmitir N símbolos se requerirá de $2N$ subportadoras.

En la Figura 1.11 [22] se observa cómo es la ubicación de varias subportadoras contiguas.

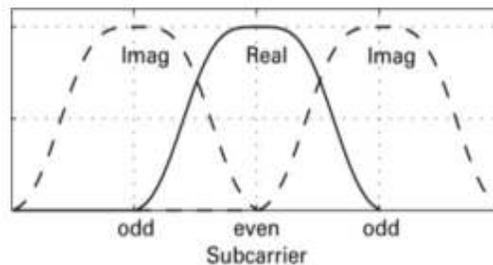


Figura 1.11. Subportadoras OQAM, parte real e imaginaria.

Usando la técnica de modulación O-QAM en conjunto con el banco de filtros se llega a prescindir del prefijo cíclico que se usa en OFDM, pues no es necesario tomar la última parte del símbolo y copiarla al inicio del mismo.

Filtrado

Teniendo en cuenta que en OFDM la DFT genera un espectro que tiene la forma de una función $\text{sinc}(x)$, como la que se observa en la Figura 1.12 [23] que proviene de un filtro tipo FIR (*Finite Impulse Response*) de ventana rectangular, se observa que el espectro presenta emisiones fuera de banda. Cabe señalar que, un filtro FIR se puede definir por coeficientes en el dominio del tiempo o por coeficientes en el dominio de la frecuencia. De

modo que estos conjuntos de coeficientes se relacionan mediante la transformada discreta de Fourier (DFT) [24].

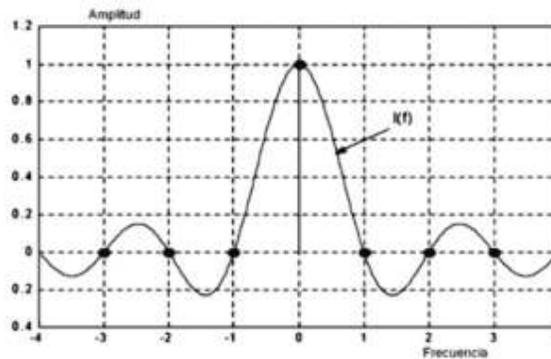


Figura 1.12. Espectro de una portadora en OFDM.

Para comenzar con la etapa del filtrado, se diseña un filtro prototipo que corresponde al primer filtro en el banco, es decir, el filtro asociado con la subportadora de frecuencia cero. Se denomina filtro prototipo porque los otros filtros se obtienen mediante desplazamientos en frecuencia. Con el fin de reducir las emisiones fuera de banda, es necesario aumentar el número de coeficientes del filtro en el dominio del tiempo y, de manera equivalente, en el dominio de la frecuencia, con el fin de tener un mejor control de la respuesta del filtro [25].

Filtro Prototipo

El filtro prototipo se caracteriza por el factor de solapamiento K , generalmente es un número entero y es la relación entre la duración de respuesta de impulso del filtro y el período de símbolo multiportadora T_s [26]. Asimismo, el factor de solapamiento en el dominio del tiempo representa el número de símbolos multiportadora que se superponen, mientras que, en el dominio de la frecuencia representa el número de coeficientes de frecuencia que se introducen entre los coeficientes del filtro FFT. Por otro lado, el factor de solapamiento es un factor clave para el uso óptimo del ancho de banda y para la supresión deseada de la banda lateral puesto que, a medida que el valor de K incrementa, mayor es la atenuación fuera de banda [27].

Bajo las consideraciones que se han planteado hasta este punto, se diseña el filtro prototipo bajo el principio de Nyquist⁴. En efecto, para facilitar el diseño del filtro se obtiene los coeficientes en el dominio de la frecuencia, en la que la respuesta del filtro contiene $2K-1$

⁴ La respuesta impulsional del filtro de transmisión debe cruzar por cero, en todos los múltiplos enteros del período de símbolo, esto, en el dominio de la frecuencia genera simetría respecto a la frecuencia de corte, es decir, la mitad del período de símbolo.

pulsos. Generalmente, se usa el filtro prototipo conocido con el nombre de *PHYDYAS* [28] cuyos coeficientes de frecuencia de medio filtro para valores de $K = 2,3,$ y 4 , se presenta en la tabla 1.2.

Tabla 1.2. Coeficientes en frecuencia del filtro prototipo.

K	H_0	H_1	H_2	H_3	σ^2 (dB)
2	1	$\sqrt{2}/2$	-	-	-35
3	1	0.911438	0.411438	-	-44
4	1	0.971960	$\sqrt{2}/2$	0.235147	-65

En la figura 1.13 [29] se muestra la respuesta en frecuencia continua del filtro prototipo a partir de los coeficientes en frecuencia de la tabla 1.2 para $K = 4$.

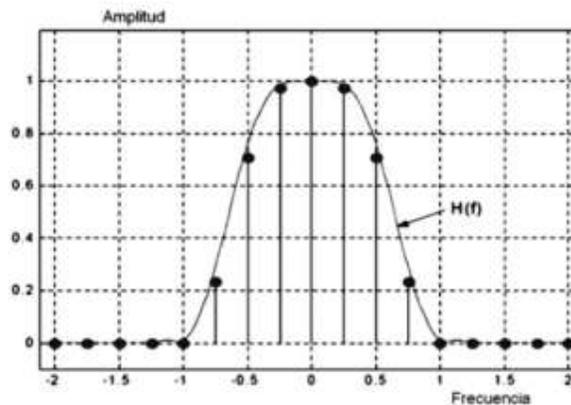


Figura 1.13. Respuesta en frecuencia del filtro prototipo.

La respuesta al impulso $h(t)$ del filtro se obtiene mediante la transformada de Fourier inversa de la respuesta en frecuencia del pulso, y queda expresada mediante la ecuación 1.9 [24]

$$h(t) = 1 + 2 \sum_{k=1}^{K-1} H_k \cos\left(2\pi \frac{kt}{KT}\right)$$

Ecuación 1.9. Respuesta al impulso.

Con el fin de comparar la respuesta al impulso del filtro prototipo para diferentes valores de K , se presenta la figura 1.14 [28] en la que se observa que el rizado fuera de banda presente en OFDM desaparece.

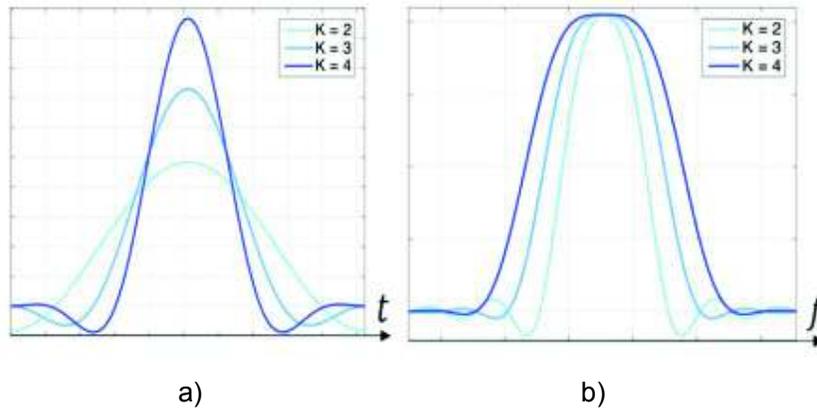


Figura 1.14. a) Respuesta Impulsiva $h(t)$, b) Respuesta frecuencial $H(f)$ del filtro prototipo FBMC.

Considerando que el procesamiento se hace en el dominio de la frecuencia este bloque se ubica antes del bloque de la IDFT ($N * K$). Una observación importante que se debe señalar es que los subcanales con índice par o índice impar no se superponen, puesto que en frecuencia un subcanal particular se solapa únicamente con sus subcanales adyacentes.

En este bloque se genera el símbolo FBMC multiportadora, para ello primero, se incrementa la frecuencia de muestreo inicial por el factor de solapamiento K , de la siguiente manera: los símbolos complejos que llegan del bloque O-QAM se separan añadiendo entre ellos $K - 1$ ceros consecutivos, de modo que , los símbolos O-QAM queden separados entre ellos un total de $K - 1$ muestras nulas [17].

Para filtrar cada subportadora con el filtro prototipo, se reparte la información de un símbolo en las subportadoras adyacentes, este procedimiento se conoce como *Frequency Spreading*. Para realizar la dispersión en frecuencia el filtro se aplica como se muestra en la Figura 1.15 [24].

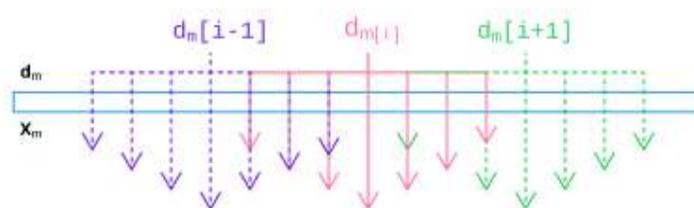


Figura 1.15. Dispersión en frecuencia.

Un elemento importante que diferencia este método del método de red polifase PPN es la longitud del filtro prototipo. En este método la longitud del filtro está expresada por $L = 2K - 1$, siendo K el factor de solapamiento que se explicó anteriormente [27].

Finalmente, después de la multiplicación por los coeficientes del filtro, la IFFT es alimentada por un valor de $2K - 1$ entradas. Lo que se hizo fue, distribuir la información de los datos en varias entradas IFFT.

IDFT ($K*N$) [25] [29]

Como en un banco de filtros con un factor de solapamiento K , un dato se modula con $2K - 1$ subportadoras, implica aumentar la frecuencia de muestreo en el dominio de la frecuencia, así, el tamaño de la IFFT deberá tener un valor de KN , donde N es el tamaño original de la IFFT. Por lo tanto, para realizar el filtrado en transmisión se usa una IFFT extendida para generar todas las muestras necesarias, esto significa que para la modulación es necesario extender el tamaño de la IFFT y la FFT.

Cabe señalar que mientras crece el valor de K , el sistema presenta mayor atenuación fuera de banda y en cierto modo incrementa la complejidad del sistema .

Sobrelapamiento y Suma

El último bloque del transmisor que genera la señal en banda base, establece el solape temporal entre símbolos, esto significa que, dependiendo del factor de solapamiento, un total de K símbolos se solaparían en tiempo. Se tiene entonces, que para cada conjunto de datos de entrada, la salida del IFFT es un bloque de KN muestras, entonces K salidas IFFT consecutivas se sobrelapan en el dominio del tiempo. Por lo tanto la salida del banco de filtros se obtiene mediante el sobrelapamiento y suma de los símbolos. Para esto, se superponen símbolos multiportadora consecutivos y suman las partes superpuestas con $N/2$ muestras. En el receptor la ventana de FFT se desliza por $N/2$ en lugar de N muestras que se tiene a la salida del bloque IFFT extendida [21] [25].

Aplicando el método expuesto anteriormente y usando los filtros escogidos, se presenta una interferencia entre subcanales adyacentes, es decir, el subcanal $k - \text{ésimo}$ interfiere únicamente con los $k - 1$ y $k + 1$ contiguos, y no con los demás, lo que significa que las subcanales pares o impares no interfieren entre sí.

A modo de resumen en la Figura 1.16 [24] se presenta gráficamente el proceso de modulación de FBMC en frecuencia.

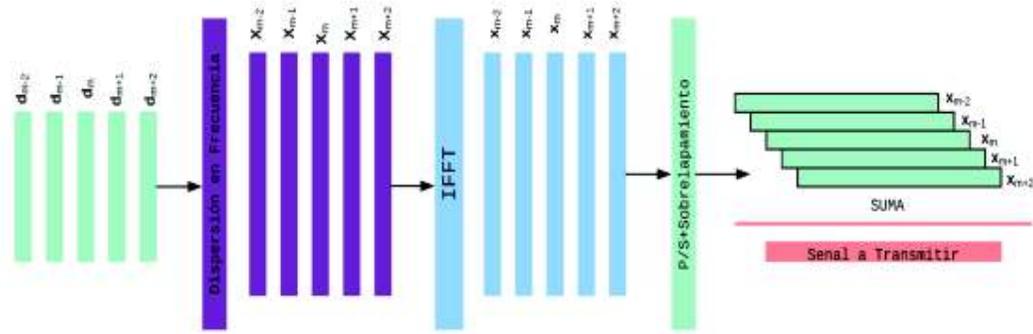


Figura 1.16. Transmisor FBMC en el dominio de la frecuencia.

1.3.9 Receptor FBMC

1.3.9.1 Demodulador FBMC

La implementación del demodulador es muy similar a la del modulador, siguiendo el orden de los bloques que se presentan en la Figura 1.17 [28]. A continuación se explica su funcionamiento.

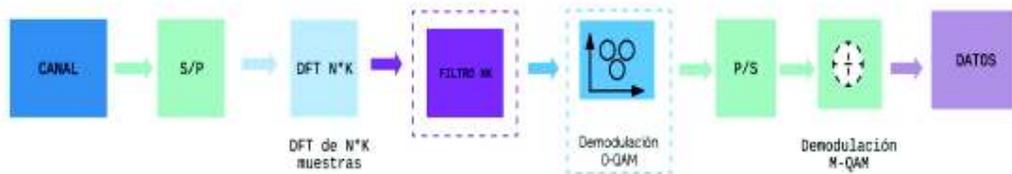


Figura 1.17. Diagrama de bloques del receptor FBMC.

Observando la figura 1.17, se plantea que en recepción se realiza el proceso inverso que en transmisión. Debido a que se emplea el mismo banco de filtros que en transmisión pero en sentido contrario, es decir, el objetivo es recuperar en una frecuencia lo que en transmisión se repartió en varias frecuencias.

Convertidor Serie Paralelo y DFT (N*K)

Las muestras temporales ingresan al bloque serie-paralelo. Una vez que se obtiene la señal sincronizada, se selecciona el primer símbolo cuya longitud sería KN muestras y se realiza la DFT mediante el algoritmo FFT del mismo tamaño.

Filtrado H_k

Ahora bien, comúnmente entre el bloque DFT KN y el filtro H_k existe una etapa de estimación y ecualización del canal, para realizar este proceso intermedio, en transmisión

tuvo que haberse guardado algunos símbolos piloto. Después de este proceso se procede a filtrar en recepción. Cabe recalcar que el filtro se aplica a la señal de salida del ecualizador en caso que se realice este proceso. Entonces, la forma en que se recupera los símbolos que se han transmitido es, aplicando el método de diezmo⁵ de factor K con la finalidad de eliminar símbolos intermedios que no contienen información [17].

Demodulador O-QAM

En este bloque con el fin de obtener los símbolos O-QAM que se han transmitido, y reducir la interferencia entre símbolos, se escoge los símbolos alternadamente, es decir, reales o imaginarios dependiendo del índice del símbolo multiportadora que se recibe.

Convertidor Paralelo Serie y Demodulación M-QAM

Los símbolos demodulados ingresan al bloque paralelo-serie para que estos símbolo entren al demodulador M-QAM dependiendo del modulador que se utilizó en transmisión.

Como se explicó anteriormente, existen dos formas de implementar FBMC, a continuación se explica el sistema FBMC mediante Red Polifase.

Transmisor FBMC en el dominio del tiempo

Posterior al sistema de FBMC implementado mediante banco de filtros de síntesis- análisis, se plantea un esquema basado en una estructura polifásica o PPN que realiza un procesamiento en el dominio del tiempo. Este esquema se propuso porque el incremento de la IDFT $K * N$ inicialmente presentaba una alta complejidad computacional. En la Figura 1.18 se presenta el esquema para un Transmisor FBMC en el dominio del tiempo [17].

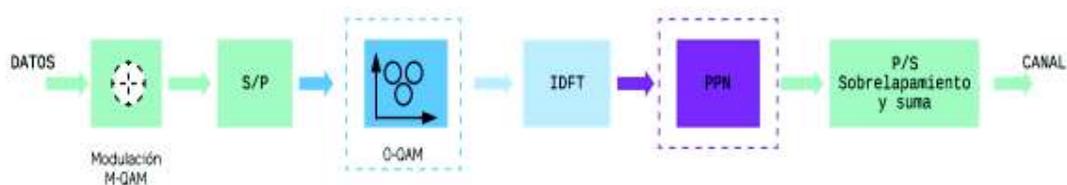


Figura 1.18. Diagrama de bloques del Transmisor FBMC en el dominio del tiempo.

Debido a que el procesamiento desde el bloque de Modulación M-QAM hasta la modulación O-QAM es similar al que se revisó en la sección del transmisor FBMC en el dominio de la frecuencia, se procede a describir este método a partir del bloque de IDFT.

⁵ Diezmo: Únicamente se toma cada muestra $N - \text{ésima}$ de la señal, con el propósito de reducir o en el mejor de los casos eliminar las redundancias en las señales.

Partiendo del esquema presentado en la Figura 1.8 se observa que se incorpora dos bloques en ese diagrama, de esta manera se obtiene el esquema de la modulación FBMC mediante el método de red polifase. Como se muestra en la Figura 1.18, en este esquema se incorpora un bloque antes de la IDFT que corresponde al bloque de modulador O-QAM, que simplemente separa los símbolos en parte real e imaginaria. La red polifásica PPN se ubica después de la IDFT, debido a que el procesamiento que se realiza en este método se realiza en el dominio del tiempo [26] [29].

Mediante este procedimiento se puede integrar el procesamiento IFFT generando las respuestas impulsivas de los filtros en la red polifase (PPN-FFT). De modo que, en este modelo se mantiene el tamaño de la IFFT añadiendo filtros digitales [17].

Como se puede observar en la figura 1.18, se elimina el bloque de CP de la figura 1.8, puesto que FBMC garantiza ortogonalidad entre subcanales mediante el bloque de modulación O-QAM. Asimismo, el diseño correcto de filtro prototipo proporciona resistencia frente a la ISI.

Este método trabaja con la salida de la IDFT, es decir, las N muestras se retrasan un valor equivalente $K - 1$ veces, posterior a este paso se realiza el filtrado usando el mismo filtro prototipo basado en el teorema de Nyquist descrito anteriormente. A diferencia del método de IDFT extendida, este filtro tendrá una longitud $L = KN$ [17].

Después de haber replicado la señal de salida de la IFFT K veces, se procede a multiplicar elemento por elemento por el filtro prototipo que se describió anteriormente, filtro de Phydys. Para realizar la suma de estas salidas filtradas, se retrasa $\frac{N}{2}$ para conseguir el efecto de la estructura PPN [26].

Para entender de mejor manera el procesamiento que realiza este método, se presenta la Figura 1.19 [24] en la que se detalla el procesamiento FBMC en el dominio del tiempo.

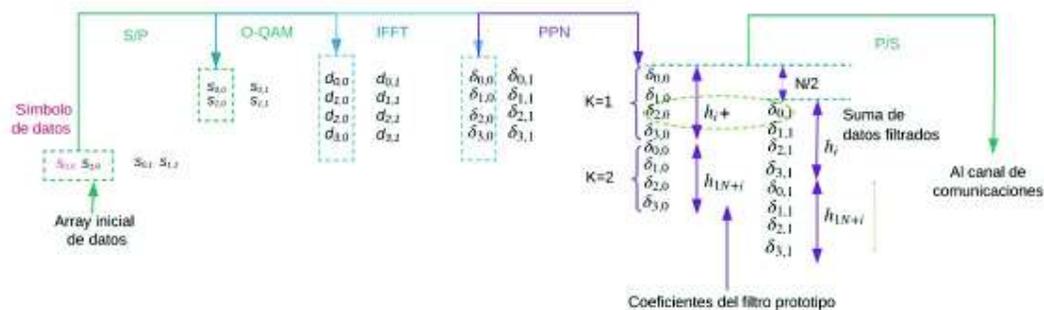


Figura 1.19. Procesado FBMC para $N = 4$.

Por su parte en recepción se realizará el proceso inverso al que se realiza en transmisión, a continuación en la Figura 1.20 se presenta el diagrama de bloques de un receptor FBMC usando la red polisafe.



Figura 1.20. Diagrama de bloques del receptor FBMC con red polisafe.

Una vez concluida la explicación de las características principales de esta técnica, se procede a describir los puntos fuertes y débiles de la misma.

1.3.10 Ventajas FBMC

Con FBMC se obtiene numerosos beneficios respecto al actual sistema multiportadora OFDM, el más evidente es la selectividad en frecuencia. Por un lado, en OFDM los grandes lóbulos laterales de cada filtro se solapan con el resto de subcanales, en los que se ha dividido el espectro OFDM, mientras que en FBMC se divide el canal de transmisión en un conjunto de subcanales en los que cada canal, únicamente, solapa con los canales contiguos.

Debido a que los sistemas FBMC utilizan un banco de filtros que permiten una correcta localización de las subportadoras tanto en el dominio del tiempo como en el de la frecuencia, estos sistemas presentan una mejor contención espectral de las señales.

De acuerdo a las características que presenta OFDM se puede presentar como ventaja que FBMC permite alcanzar una interferencia entre símbolos y entre subportadoras menor.

Por otro lado, otra ventaja que presenta esta modulación, es la posibilidad de diseñar subcanales con la atenuación que se desee en la banda de rechazo, esto genera un incremento en el rendimiento del sistema puesto que se eliminan las subportadoras destinadas a bandas de guarda.

1.3.11 Desventajas FBMC

Esta modulación presenta más complejidad que en OFDM, puesto que se incorpora un procesamiento adicional a la señal.

Como desventaja de estos sistemas se tiene que la utilización del banco de filtros genera un solapamiento en sus formas de onda, de manera que en el transmisor existen bloques que dependan del factor de solapamiento.

Puesto que el filtrado que presenta esta modulación es de banda estrecha, genera un respuesta impulsiva larga en el tiempo, esto provoca que FBMC no sea idónea para comunicaciones con pequeñas cantidades de datos.

1.3.12 UFMC

UFMC (*Universal Filtered Multi-Carrier*), esta modulación combina las ventajas de ortogonalidad de OFDM y el banco de filtros de FBMC, pues, FBMC debido al filtrado por subportadora presenta una respuesta impulsiva larga. Con el fin de conseguir una respuesta impulsiva corta se podría pensar en realizar un filtrado de todas las subportadoras, esta técnica ya existe y se conoce con el nombre de OFDM filtrada (F-OFDM) [30].

1.3.12.1 Principios UFMC

División en subbandas

Manteniendo la idea de realizar un filtrado, pero no de cada subportadora (FBMC) ni de todas las subportadoras (F-OFDM), sino de un grupo de subportadoras, UFMC divide el ancho de banda total en B subbandas⁶ con k_b subportadoras cada una, después aplica un filtrado común a cada subbanda de manera independiente [22]. El proceso de filtrar a toda una subbanda y no solo a una subportadora es para obtener un ancho de banda mayor que en FBMC, por lo tanto, se obtendría una respuesta al impulso mucho menor. La idea de filtrar un grupo de subportadoras viene de que los desalineamientos en tiempo/frecuencia habitualmente ocurren entre grupos de subportadoras, como es el caso cuando se asigna un grupo de subportadoras a un usuario en UL (*Up Link*) [24].

Como se puede observar, si en UFMC el número de subportadoras coincide con el número de bandas ($B = 1$) tenemos el caso de FBMC. Por consiguiente UFMC se presenta como una generalización o “universalización” del principio de FBMC (filtrado por subportadora) y OFDM. Cabe señalar que al filtrar las subbandas de manera independiente generaría un mejor confinamiento espectral [25] [26]. En la figura 1.21 se muestra la representación de las diferentes formas de filtrar, ya sea, filtrado de banda completa, filtrado por subportadora, filtrado por un conjunto de subportadoras que realizan las modulaciones OFDM, FBMC y UFMC respectivamente.

⁶ Subbanda: Corresponde a un conjunto de subportadoras que es formado del ancho de banda total del sistema. [22]

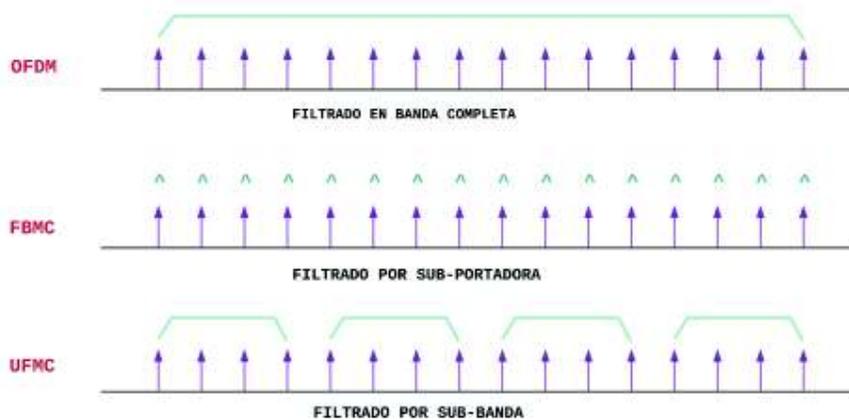


Figura 1.21. Tipos de filtro según la forma de onda.

Si bien es cierto, FBMC es más eficiente al transmitir largas cantidades de datos debido a la longitud del filtro $L = KN$, con K como el factor de solapamiento y N el número de subportadoras. Si se requiere enviar pequeñas cantidades de datos el uso de FBMC no sería recomendable. Debido a que UFMC presenta una respuesta impulsiva corta, es eficiente para aplicaciones de transmisión de datos cortas y con baja latencia [28]. Además, presenta ventajas como reducir la ICI ante un *jitter*⁷ en frecuencia por el efecto *Doppler*, que permitirían una mayor velocidad de movimiento del usuario.

Prefijo Cíclico

Por otra parte, debido a que en UFMC no se utiliza CP, en su lugar se recomienda el diseño de un filtro tal que la longitud del mismo coincida con la longitud de un CP empleado en OFDM. De esta manera, se observa que trabajar con subbandas permite reducir la longitud del filtro, ya que al aplicar el filtro a una subbanda (conjunto de subportadoras) necesita un ancho de banda mayor que el que se utiliza en FBMC, lo que genera una respuesta al impulso mucho menor. Cabe señalar que a diferencia que en FBMC en UFMC no existe solapamiento entre símbolos [27].

Como UFMC no utiliza prefijo cíclico presenta mejor eficiencia espectral en comparación con OFDM, puesto que al prescindir del mismo, utiliza todo el espectro asignado de manera eficiente. De este modo, UFMC presenta menos lóbulos laterales que OFDM, cabe recalcar que conforme los lóbulos laterales disminuyen, la interferencia en las subportadoras adyacentes también disminuyen.

En el dominio del tiempo, entre diferentes símbolos multiportadora UFMC presenta una

⁷ *Jitter* (Fluctuación de fase): Desviación de la periodicidad de una señal en relación con una señal de reloj de referencia [28]

transición suave, debido a esto entre símbolos adyacentes no hay solapamiento. Como se observa en la figura 1.22 [28] la amplitud de la envolvente de la subportadoras después de la subida permanece constante durante la duración del símbolo, después presenta una ligera transición de bajada, situación que no ocurre en OFDM [28] [29].

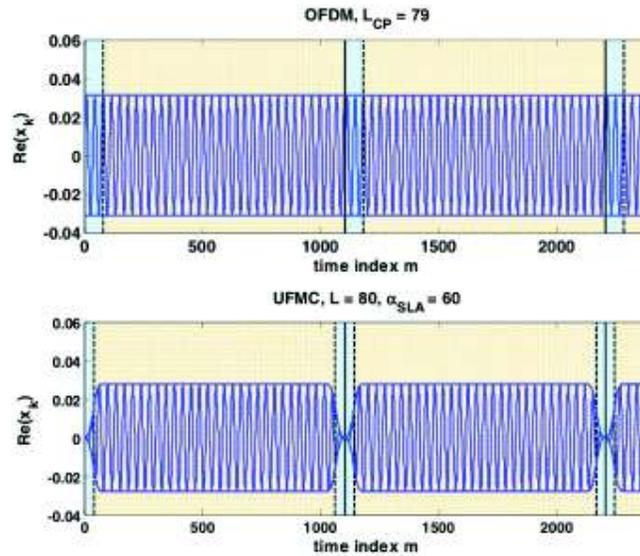


Figura 1.22. Característica de UFMC en el dominio del tiempo.

Si UFMC, se diseña con parámetros similares a los que se utiliza en LTE, es decir si en UFMC se emplea un filtro cuya longitud sea igual a $L = L_{CP-OFDM} + 1$, llega a ser más selectiva, lo que permite realizar una mejor ecualización sin necesidad de usar prefijo cíclico.

Filtrado

Para esta modulación se utiliza un filtro FIR de ventana *Dolph Chebyshev*⁸ (debido a que con este filtro presenta menores lóbulos secundarios). Este filtro cuenta con parámetros configurables como L que corresponde a la longitud del filtro, y α correspondiente a la atenuación del lóbulo secundario. La selección de estos parámetros así como su configuración son parte fundamental en el diseño del filtro, puesto que de estos valores dependerá alcanzar una caída en las colas de las bandas frecuenciales similares a las que se tiene en FBMC con el fin de conseguir una eficiencia espectral similar [30].

⁸ Ventana *Dolph Chebyshev* que minimiza el ancho del lóbulo principal [30]

En la Figura 1.23 [24] se muestra el filtro *Chebyshev* en el dominio del tiempo y en el dominio de la frecuencia, con los siguientes parámetros:

- Longitud:80
- Atenuación de los lóbulos secundarios: 60

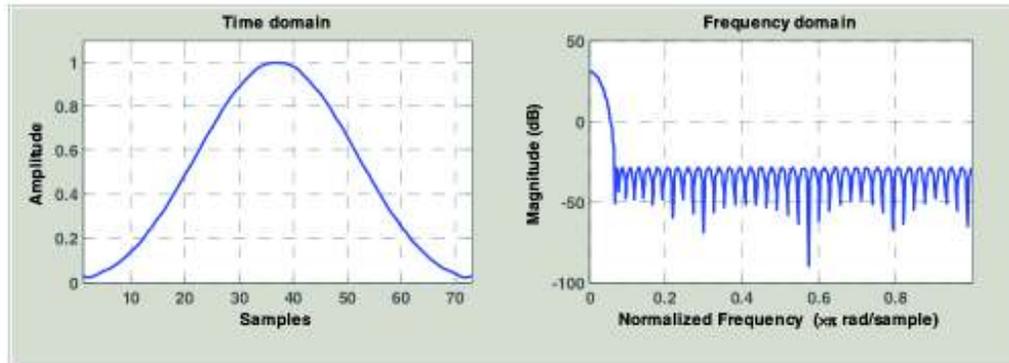


Figura 1.23. Filtro *Dolph Chebyshev* en el dominio del tiempo y de la frecuencia.

La longitud del filtro puede tomar diferentes valores, generalmente su tamaño está relacionado con el tamaño de prefijo cíclico que se utiliza en OFDM. Para α se tiene valores recomendados como $\alpha = 30, 40, 60$ y 90 . Debe señalarse que los valores de L y α delimita el número de subportadoras óptimo a utilizar [30].

A continuación en la Figura 1.24, se presenta un ejemplo del filtro diseñado con valores de $L = 74$ para distintos valores de α , en la figura se puede observar la respuesta impulsiva en cada caso [28].

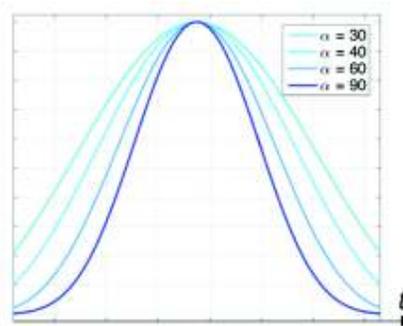


Figura 1.24. Respuesta impulsiva del filtro *Dolph Chebyshev* en función de α .

Asimismo, la forma de filtrado que realiza UFMC puede aprovecharse para multiplexar en frecuencia subbandas con subportadoras que correspondan a diferentes servicios. Los

filtros de las subbandas están desplazados en frecuencia mediante la IDFT. En la figura 1.25 se muestra el resultado de un ejemplo que ha multiplexado 6 subbandas [24].

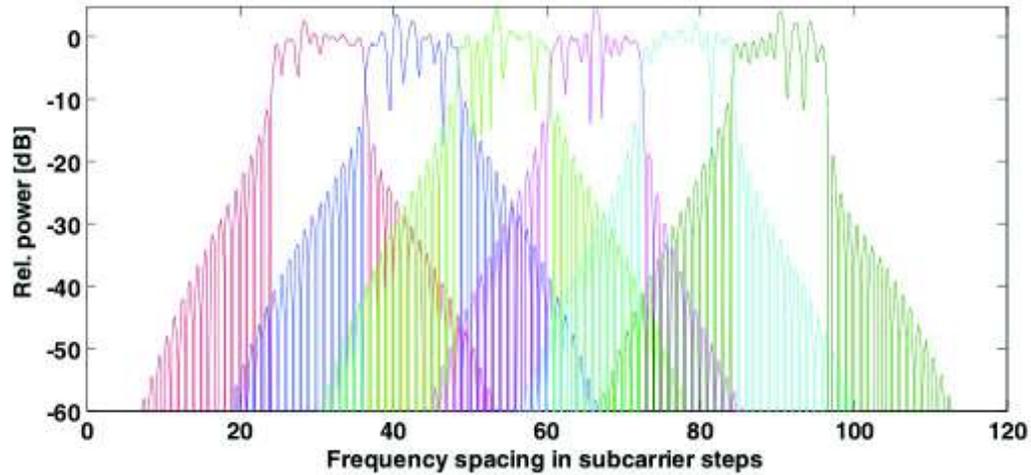


Figura 1.25. Multiplexación de 6 subbandas.

Si comparamos los espectros que se obtiene tanto para OFDM como para UPMC con una subbanda se puede observar la atenuación fuera de banda que presenta UPMC respecto a OFDM. El inconveniente de tener un lóbulo con poca atenuación se elimina con UPMC, tal como se muestra en la figura 1.26 [24].

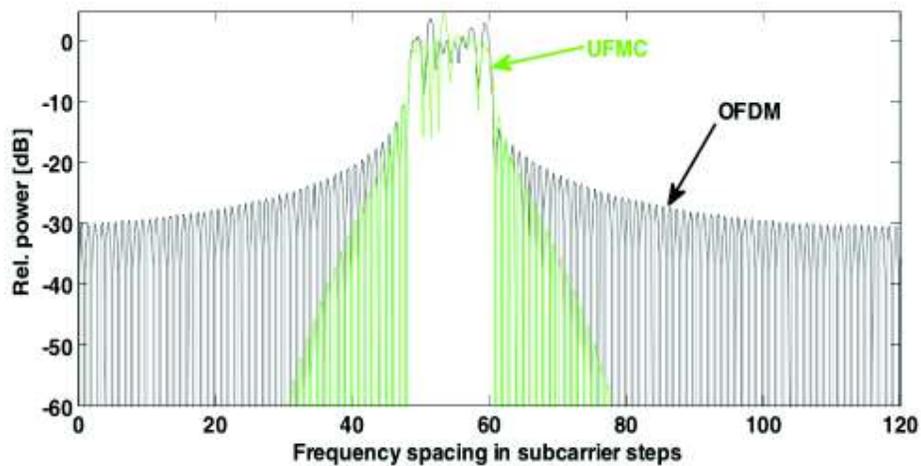


Figura 1.26. Espectro de OFDM comparado con espectro UPMC para una subbanda.

1.3.13 Transmisor UPMC

En la Figura 1.27 [30] se puede observar el diagrama de bloques del transmisor UPMC.

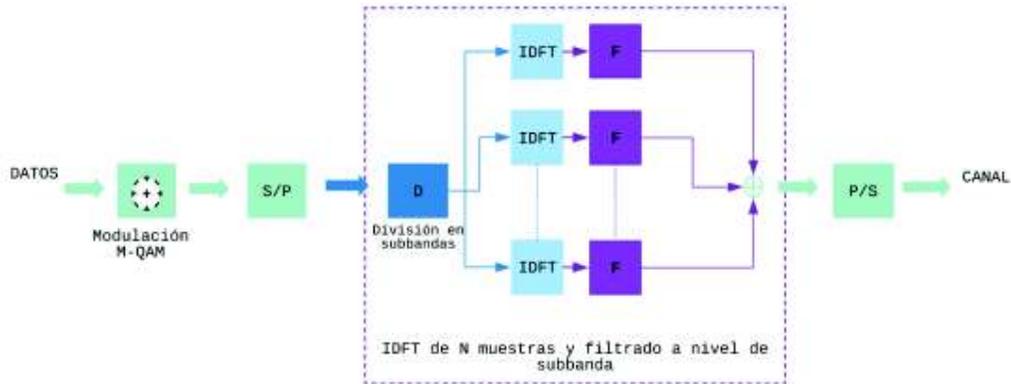


Figura 1.27. Diagrama de bloques del Transmisor UFMC.

Resumiendo, en el esquema que se presenta en la figura 1.27 [30] se muestra el procesado de la modulación de UFMC, se observa que a la IFFT se aplica al conjunto de símbolos que corresponden a una determinada subbanda. Después se tiene la superposición de subbanda luego de ser filtradas y finalmente son enviadas al canal de manera conjunta.

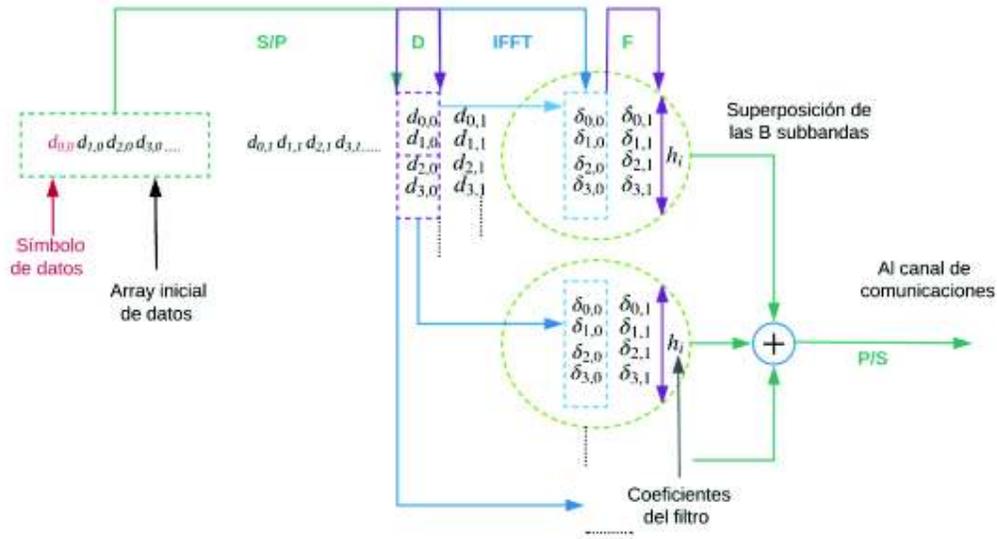


Figura 1.28. Modulación UFMC para un valor de $N = 4$.

Para entender de mejor manera se explica cada bloque de la figura 1.27

1.3.13.1 Modulación M-QAM

Para comenzar el proceso de modulación, los datos que proviene de una fuente generadora, ingresan a este bloque para ser modulados mediante una constelación M-

QAM. En otras palabras, a los bits que ingresan en este bloque se les asigna símbolos complejos mediante el modulador M-QAM.

1.3.13.2 Conversión Serie/Paralelo

Los símbolos que provienen de la constelación son números complejos, estos símbolos vienen en forma de cadena uno detrás de otro. Con el fin de situarlos en paralelo a la entrada de la IDFT estos símbolos pasan a través de un convertidor de serie a paralelo. Así, la salida de este convertidor S/P se asigna a la IFFT.

1.3.13.3 División de subbandas (D).

Puesto que B es un parámetro indispensable en el diseño de un sistema UFMC, a elección el tamaño adecuado va a depender de las circunstancias del escenario en el que se utilice el transmisor UFMC. En este sentido, se considera que mientras el sistema corresponda a un escenario de espectro fragmentado, B se podrá elegir considerando el número de subportadoras disponibles. Cabe mencionar que B puede cambiar en el tiempo siempre y cuando algunas subbandas del espectro ocasionalmente sean usadas por otros sistemas [24] [28].

Un factor importante que hay que considerar es que, si se configura igual número de subportadoras en todas las subbandas, las subportadoras mantienen ortogonalidad en el dominio complejo, de este modo, no será necesario usar la modulación O-QAM como en el caso de FBMC [30].

1.3.13.4 IFFT

Los símbolos a transmitir, ingresan al bloque de IFFT con el fin de distribuir la información en frecuencias ortogonales. Cabe señalar que este bloque convierte los símbolos complejos QAM del dominio de la frecuencia al dominio del tiempo de la misma manera que se realiza en OFDM. En este bloque es muy importante asegurarnos que a la entrada los símbolos se sitúen de forma correcta, es decir, la entrada de la IFFT debe corresponder a la subbanda correspondiente [21] [22].

1.3.13.5 Filtrado por Subbanda. (F)

Como se mencionó anteriormente, el filtrado en UFMC se realiza por subbandas de forma independiente. Para este proceso se utiliza el filtro con ventana *Dolph Chebyshev*. El filtrado se realiza en tiempo de la siguiente manera, las señales de cada subbanda ingresan al filtro prototipo y se filtran mediante el proceso de convolución. A la salida de cada filtro se genera un vector de tamaño $N + N_{filtro} - 1$, este vector se transmitirá en el dominio del

tiempo y se forma de la superposición de las componentes de las subbandas. En la figura 1.29 [24] se presenta un diagrama del filtrado por subbanda.

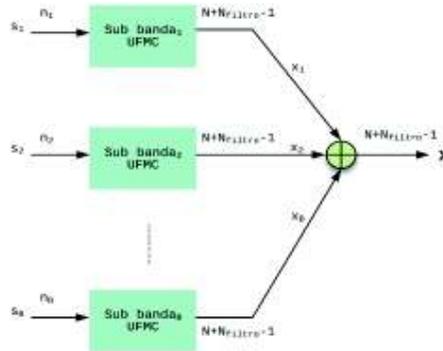


Figura 1.29. Filtrado de B subbandas.

El vector transmitido en el dominio del tiempo se lo puede representar mediante la ecuación 1.10. [24]

$$x_k = \sum_{i=1}^B F_{i,k} \cdot V_{i,k} \cdot S_{i,k}$$

Ecuación 1.10. Salida del filtro.

Donde:

B: corresponde al total de subbandas.

F_i: Matriz Toeplitz de dimensiones $(N_{IFFT} + N_f - 1) \times N$, compuesta por la respuesta al impulso, esta matriz se la utiliza para realizar la convolución.

V_i: Representa la matriz de coeficientes de la IFFT, con dimensiones $N \times n_i$, y n_i incluye las columnas de la matriz inversa de Fourier dependiendo la posición de la subbanda respectiva en el rango de la frecuencia disponible.

Entonces, los bloques de datos de entrada representados por $S_{i,k}$ se convierten en B subbandas y la i-ésima subbanda pasa a través del punto N IFFT que está representada con la matriz $V_{i,k}$. La salida de IFFT se serializará y pasará a través del filtro que se representa con la matriz $F_{i,k}$. En la Figura 1.30 [24] se indica el filtrado de una sola subbanda.

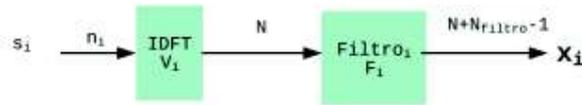


Figura 1.30. Filtrado de una Subbanda.

La ecuación 1.10 puede ser reescrita en forma matricial con el fin de ordenar eficientemente las matrices. Con estas matrices se genera una matriz IDFT de bloques diagonal, de modo que los símbolos se ordenan en una columna. Entonces, la salida del filtro descrita en forma de matriz corresponde a la ecuación 1.11 [24].

$$x = \bar{F}\bar{V}\bar{s}$$

Ecuación 1.11. Símbolo x a la salida del filtro.

Finalmente al sumar las salidas de todas las subbandas se obtiene una señal con una longitud total similar a la de OFDM tal como se muestra en la ecuación 1.12 [24]:

$$N_{tot} = N_{IFFT} + N_{filtro} - 1$$

Ecuación 1.12. Longitud total de la señal UPMC.

Donde:

N_{IFFT} : Representa el tamaño de la IFFT.

N_f : Representa el tamaño del filtro en el dominio del tiempo.

Los filtros de las subbandas son desplazados en frecuencia mediante la IDFT partiendo de la frecuencia central de cada subbanda.

1.3.13.6 Suma de subbandas y Conversor Paralelo/Serie

Una vez que se obtienen los vectores resultantes del filtrado en tiempo de todas las subbandas se suman para formar una sola señal. Los símbolos UPMC deben ingresar al bloque paralelo/serie antes de enviarse la señal al canal.

Finalmente, la descripción del proceso de modulación que realiza UPMC, lleva a verificar que este procesamiento corresponde a un paso intermedio entre OFDM y UPMC. Conviene señalar que aunque el filtrado se realice por subbandas se mantiene el espaciado entre ellas por lo tanto el tamaño de IFFT se mantiene.

1.3.14 Receptor UFMC [24] [30]

La Figura 1.31 muestra el diagrama de bloques del Receptor UFMC.



Figura 1.31. Diagrama de bloques del receptor UFMC.

1.3.14.1 Convertidor Serie/ Paralelo

Los datos recibidos en recepción después de su propagación por el canal (etapa RF), incluye distorsión causada por el ruido. Esta señal ingresa al convertidor de serie a paralelo para ubicarlos de manera correcta antes que ingresen a la FFT .

1.3.14.2 2N-FFT

Como se mencionó en la etapa de modulación, el símbolo final tendría una longitud $N_{tot} = N_{FFT} + N_{filtro} - 1$ a causa del filtro, por lo tanto, en recepción se debe contar con una FFT de tamaño $2N$ debido al procesamiento que se realizó en transmisión. Con el fin de extraer únicamente las subportadoras originales que se transmitieron se diezma por un factor 2. El filtro de recepción es tan importante como el filtro de transmisión, por lo tanto, también aplicamos el filtrado de subbandas en el receptor para suprimir la interferencia.

1.3.14.3 Ecualizador de Símbolos

Después de la conversión de FFT en el dominio de la frecuencia queda eliminar la distorsión ocasionada por el filtro de transmisión y se aplica un procesamiento de ecualización para mejorar la calidad de la señal.

1.3.14.4 Convertidor Paralelo/Serie

La salida del ecualizador ingresa en paralelo al convertidor en serie. Este convierte todos los flujos de datos paralelos en una sola secuencia para enviarlos al bloque de demodulador M-QAM.

1.3.14.5 Demodulador M-QAM

El demodulador de símbolos convierte los símbolos en bits y se recuperan los datos originales.

A continuación se presenta los puntos fuertes y débiles de la técnica UFMC.

1.3.15 Ventajas de UFMC

UFMC presenta una respuesta impulsiva menor en comparación con FBMC, por lo que se podría usar para la transmisión de pequeñas cantidades de datos.

Como en UFMC el filtrado se realiza por subbandas y es menos complejo que FBMC.

No necesita prefijo cíclico, sin embargo para mejorar la interferencia entre símbolos, puede usarlo, mejora los resultados tanto de FBMC como de OFDM.

Esta técnica presenta un espectro con lóbulos laterales menores que los de OFDM, ayuda a reducir al mínimo las emisiones fuera de banda, significa que, conforme los lóbulos laterales disminuyen, la interferencia entre subportadoras adyacentes también lo hace.

1.3.16 Desventajas de UFMC

Requiere una FFT de mayor tamaño que OFDM, por lo que la complejidad en los receptores es mayor respecto a las otras técnicas.

Debido a que se pierde parte de la ortogonalidad a causa del filtrado, presentaría problemas en aplicaciones que requieran altas velocidades.

2 METODOLOGÍA

En este apartado se describe la metodología que se usó para el desarrollo de la programación al implementar los sistemas OFDM, FBMC y UFMC. Para las simulaciones se usó el software de Matlab_R2016b, este software mediante la utilización del toolbox *Sytem Communications* permite usar algoritmos diseñados que facilita obtener gráficas de PSD (*Power Spectral Density*) y gráficas de BER (*Bit Error Rate*) vs SNR (*Signal to Noise Ratio*). Es importante presentar inicialmente un diagrama general en el que se detalla la estructura de la programación de cada modulación. A continuación, se ilustra en la figura 2.1 el diagrama de clases en el que se indica la estructura de todo el sistema, es decir, el proceso que se realizó para la elaboración de clases, funciones y scripts, y la forma en que están relacionados cada una.

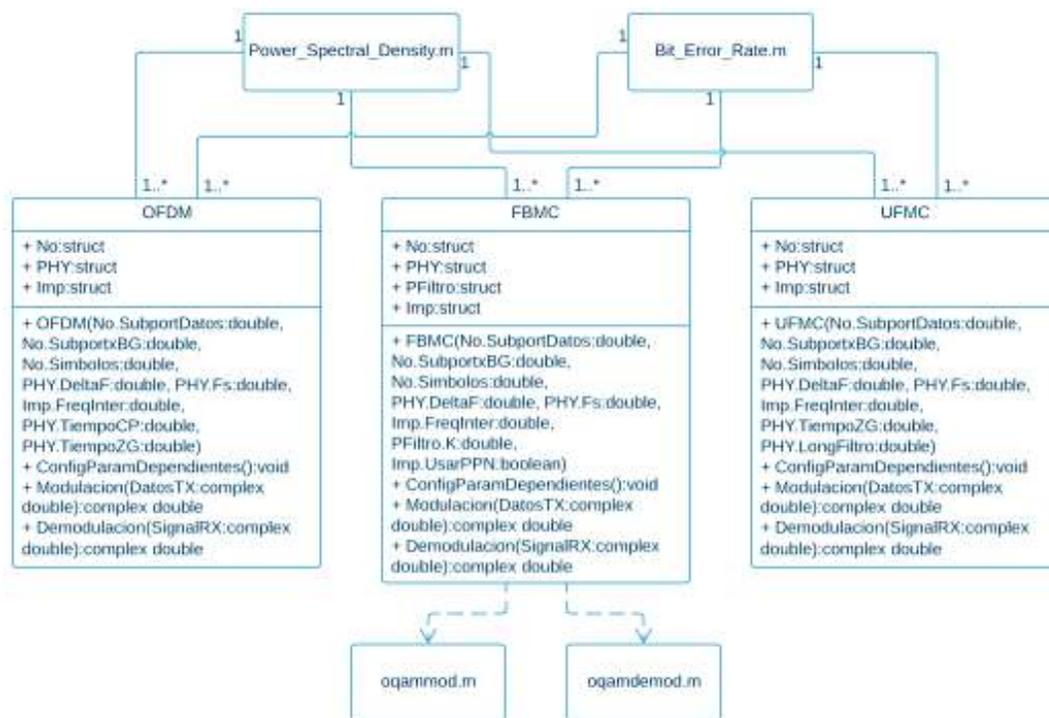


Figura 2.1 Diagrama de clases del programa implementado.

Para comparar las gráficas de Densidad Espectral de Potencia, y las curvas de BER vs SNR tanto de OFDM, FBMC y UFMC, se crean dos scripts principales que son:

- 1) `Power_Spectral_Density.m`
- 2) `Bit_Error_Rate.m`

Estos scripts se crean para facilitar la simulación de las tres modulaciones en conjunto OFDM, FBMC y UFMC, y poder compararlas. La ejecución de estos scripts depende de la

característica que se decida comparar. Su funcionamiento se detallará más adelante en los apartados 2.4 y 2.5.

Siguiendo el orden del diagrama de clases de la figura 2.1, se crearon clases individuales de MATLAB para cada técnica de modulación, estas clases son las siguientes:

- 1) OFDM (*OFDM.m*)
- 2) FBMC (*FBMC.m*)
- 3) UFMC (*UFMC.m*)

Ahora bien, dentro de cada clase se definen los atributos, mediante un bloque de propiedades que pueden ser declarados sin asignación de valores, las propiedades que tendrá cada una de las clases serán las siguientes:

- 1) No
- 2) PHY
- 3) Imp
- 4) PFiltro (exclusivo para FBMC)

Asimismo, dentro de cada clase se crean métodos que son ejecutados en los scripts principales dependiendo de la simulación que se quiera ejecutar, los métodos de cada clase son los siguientes:

- 1) ConfigParamDependientes
- 2) Modulacion
- 3) Demodulacion

Cabe señalar que al final del documento se adjunta en los ANEXOS el código de Matlab de todas las simulaciones que se realizaron, para una mejor comprensión, en cada apartado se irá comentando el código utilizado y la relación que tiene con cada bloque del sistema. A continuación se procede a describir la implementación de las clases de cada una de las modulaciones.

2.1 Implementación de la clase OFDM.

2.1.1 Propiedades de la clase OFDM

La clase *OFDM.m* permite la creación de objetos que serán utilizados posteriormente en los scripts de Densidad Espectral de Potencia y de BER vs SNR. Las propiedades y los parámetros de entrada que se almacenan en esta clase se ilustra en la Tabla 2.1.

Tabla 2.1 Propiedades de la clase `OFDM.m`

Propiedad	Descripción	Parámetros de entrada de la clase OFDM
No	Estructura que almacena valores adimensionales utilizados en las modulaciones.	<ul style="list-style-type: none"> - No.SubportDatos - No.SubportxBG - No.Simbolos
PHY	Estructura que almacena valores con significado físico.	<ul style="list-style-type: none"> - PHY.DeltaF - PHY.Fs
Imp	Estructura que almacena valores utilizados en la implementación de las modulaciones.	<ul style="list-style-type: none"> - Imp.FreqInter

2.1.2 Métodos de la clase OFDM

Antes de definir los métodos, se crea el constructor de la clase OFDM, este constructor se encarga de almacenar los parámetros que se presentan en la Tabla 2.2 dentro de las propiedades de la Tabla 2.1.

Tabla 2.2 Parámetros de entrada de la clase `OFDM.m`.

Parámetro	Descripción
No.SubportDatos	Número de subportadoras utilizadas para transportar datos.
No.SubportxBG	Número de subportadoras de cada banda de guarda.
No.Simbolos	Número de símbolos OFDM a generarse.
PHY.DeltaF	Espaciamiento entre subportadoras.
PHY.Fs	Frecuencia de muestreo.
Imp.FreqInter	Permite realizar un desplazamiento en frecuencia, solo para propósitos de visualización.
PHY.TiempoCP	Duración del prefijo cíclico en segundos.
PHY.TiempoZG	Duración de los intervalos de guarda en segundos.

En el código 2.1 se crea el constructor de la clase OFDM, en este código se observa que los parámetros descritos en la Tabla 2.2 se asignan a las propiedades de la clase OFDM.

Una vez creado el constructor de la clase OFDM, se precede a definir los métodos `ConfigParamDependientes`, `Modulacion`, `Demodulacion`, respectivamente.

```

% Constructor de la clase
function obj = OFDM(varargin)
    % Se inicializan los parámetros
    if numel(varargin) == 8
        obj.No.SubportDatos = varargin{1}; % Número de subportadoras de
datos
        obj.No.SubportxBG = varargin[15]; % Número de subportadoras por
banda de guarda
        obj.No.Simbolos = varargin[16]; % Número de símbolos OFDM
        obj.PHY.DeltaF = varargin{4}; % Espaciamiento entre
subportadoras (Hz)
        obj.PHY.Fs = varargin{5}; % Frecuencia de muestreo (Hz)
        obj.Imp.FreqInter = varargin[8]; % Desplazamiento en frecuencia
(para propósitos de visualización)
        obj.PHY.TiempoCP = varargin[17]; % Duración del prefijo
cíclico (s)
        obj.PHY.TiempoZG = varargin{8}; % Duración de los intervalos
de guarda (s)
    else
        error('El número de variables de entrada debe ser 0 (valores
predeterminados) u 8');
    end
    % Configurar parámetros dependientes
    obj.ConfigParamDependientes();
end

```

Código 2.1. Constructor de la clase OFDM.

2.1.2.1 Método ConfigParamDependientes

Como se observa en el Código 2.1, al final del constructor se llama al método `ConfigParamDependientes`, en el que se configura los parámetros dependientes de los parámetros definidos en el constructor OFDM. La descripción de estos parámetros se especifican en la Tabla 2.3. Todos estos parámetros son definidos dentro de cada clase como se indica en el ANEXO I.

Tabla 2.3 Parámetros dependientes en la clase `OFDM.m`

Parámetro	Descripción
No.TotalSubport	Número total de subportadoras, es decir, subportadoras de datos más subportadoras de cada banda de guarda.
Imp.MuestrasCP	Número de muestras del prefijo cíclico.
Imp.MuestrasZG	Número de muestras de los intervalos de guarda.
Imp.MuestrasSimb	Número de muestras de un símbolo OFDM.
Imp.NFFT	Tamaño de la IFFT.
Imp.FactorNorm	Factor de normalización empírico para que la potencia promedio de la señal sea igual a 1W.
PHY.dt	Tiempo entre muestras.
PHY.TiempoSimb	Tiempo de un símbolo OFDM.
No.TotalMuestras	Número total de muestras a transmitirse.

2.1.2.2 Método Modulación

Así mismo, dentro de la clase OFDM, se define el método `Modulación`. En este método se realiza el programa que procesa los datos en el transmisor OFDM, es decir, la conversión de serie a paralelo, modulación IFFT, inserción del prefijo cíclico. Estos datos modulados serán transmitidos al receptor OFDM a través de un canal AWGN.

El diagrama de flujo que se ilustra en la Figura 2.2 indica el proceso que sigue la información en un transmisor OFDM.

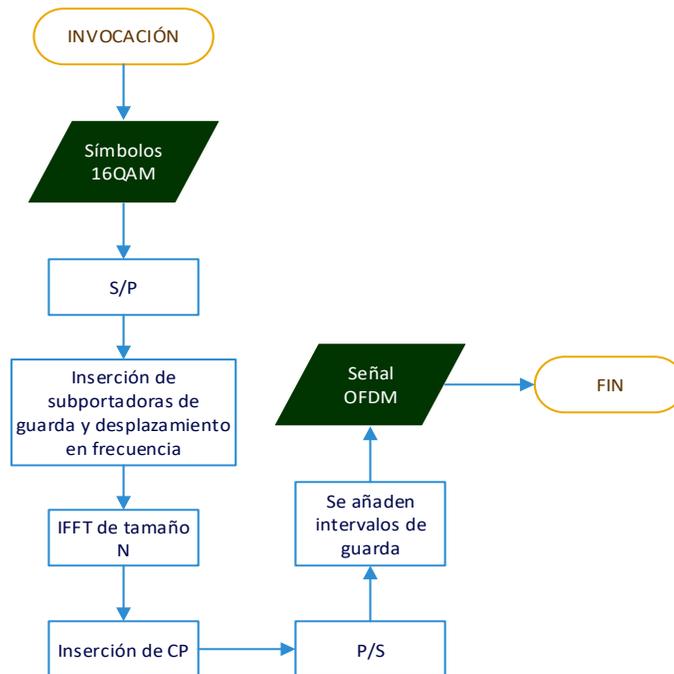


Figura 2.2 Diagrama de flujo de modulación OFDM.

Una vez definido el diagrama de flujo para la modulación OFDM mostrado en la Figura 2.2, se procede a analizar el procesamiento de la información previo al envío al receptor OFDM.

Es importante señalar que los datos que ingresan al modulador se obtienen de los scripts principales (Densidad de Potencia ó BER vs SNR), es decir, en los scripts principales se encuentra el proceso de generación de datos aleatorios y modulación mediante 16-QAM. De modo que, dentro de este método se trabaja con los símbolos 16-QAM que son generados dentro de los scripts principales. El proceso para generar los datos de entrada a este método se detalla en los apartados 2.4 y 2.5.

Parámetros de entrada y salida del método `Modulacion`

Los parámetros que utiliza el método se especifican en la Tabla 2.4. Este método trabaja con los valores almacenados en las propiedades de la clase OFDM, tales como: número de subportadoras (datos y guarda), tamaño de la IFFT, factor de normalización, desplazamiento en frecuencia, cantidad de símbolos OFDM, número de muestras del prefijo cíclico y de los intervalos de guarda, descritos en la Tablas 2.2 , y Tabla 2.3.

Tabla 2.4 Parámetros de entrada y salida del método `Modulacion`.

Parámetro	Descripción
<code>obj</code>	Objeto al que se hace referencia. Este parámetro no se ingresa debido a que MATLAB lo hace por defecto.
<code>DatosTX</code>	Parámetro de entrada, corresponde a la matriz de símbolos 16-QAM.
<code>SignalTX</code>	Parámetro de salida, corresponde al vector de símbolos OFDM.

Siguiendo el orden de cada bloque del diagrama de flujo de la Figura 2.2 se procede a explicar el procesamiento de la información en el transmisor OFDM, para generar la señal `SignalTX`.

Conversión Serie/Paralelo

Como parámetro de entrada se tiene `DatosTX`, que corresponde a la matriz de símbolos 16-QAM que son generados en los scripts principales. Estos símbolos se convierte de serie a paralelo. Después de la conversión a paralelo se obtiene una matriz resultante que tendrá un tamaño igual al número de subportadoras de datos por la cantidad de símbolos OFDM. Además, los símbolos 16-QAM se normalizan para que la señal OFDM tenga una potencia media igual a la unidad. Este proceso se realiza en el código 2.2:

```
% Se convierte de S/P y se normaliza para tener potencia = 1
Datos_S_P = reshape(DatosTX, [], obj.No.Simbolos)*obj.Imp.FactorNorm;
```

Código 2.2. Conversión serie / paralelo.

Inserción de subportadoras de guarda y desplazamiento en frecuencia

Mediante este proceso se añaden las subportadoras de guarda y se realiza un desplazamiento en frecuencia, de ser el caso, antes de la IFFT. Las subportadoras de guarda corresponden a un flujo de ceros, su valor depende del parámetro `No.SubportxBG`. Esto se hace con el fin de evitar interferencia entre canales adyacentes. Este procedimiento se realiza en el código 2.3.

```
DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.SubportDatos), :)
= Datos_S_P; % Se añaden las subportadoras de guarda
```

Código 2.3. Inserción de subportadoras de guarda.

IFFT

El siguiente paso consiste en realizar la operación de la IFFT. Para este proceso se utiliza la función *ifft()*, mediante esta función se obtiene la información en tiempo que se va a transmitir, es decir, pasamos del dominio de la frecuencia al dominio del tiempo.

Inserción CP, conversión de Paralelo/Serie

Puesto que la señal transmitida puede alterarse debido a fenómenos presente en el canal de comunicaciones, se introduce un intervalo de guarda, conocido como prefijo cíclico, cuyo tamaño será el mismo que se emplea en LTE, que corresponde a 1/14 del tiempo de símbolo OFDM. Sin embargo, este parámetro puede ser configurado dependiendo de los requerimientos del usuario. Finalmente, se convierte de paralelo a serie los símbolos OFDM generados, se añaden intervalos de guarda y se retorna la señal de salida `SignalTX` al script para su procesamiento. Este proceso se realiza en el código 2.4

```
SignalTX = [zeros(obj.Imp.MuestrasZG,1); reshape([SignalTXsinCP(end-
obj.Imp.MuestrasCP+1:end,:); SignalTXsinCP], [], 1); ...
zeros(obj.Imp.MuestrasZG,1)]; % Se añade el prefijo cíclico, se
convierte de P/S y se añaden los intervalos de guarda
```

Código 2.4. Inserción del prefijo cíclico y conversión paralelo / serie.

A continuación se procede a describir el método Demodulación.

2.1.2.3 Método Demodulacion

La demodulación OFDM se realiza mediante el método `Demodulacion`. Este método realiza el proceso del receptor OFDM, es decir, el proceso inverso al método `Modulacion`. En resumen hará lo siguiente: eliminar los intervalos de guarda, conversión serie a paralelo, eliminación prefijo cíclico, demodulación FFT, eliminación de subportadoras de guarda y desplazamiento en frecuencia. Este proceso se ilustra en el diagrama de flujo de la Figura 2.3.

Una vez definido el diagrama de flujo para la demodulación OFDM mostrado en la Figura 2.3, se procede a analizar el procesamiento de los datos en el receptor OFDM.

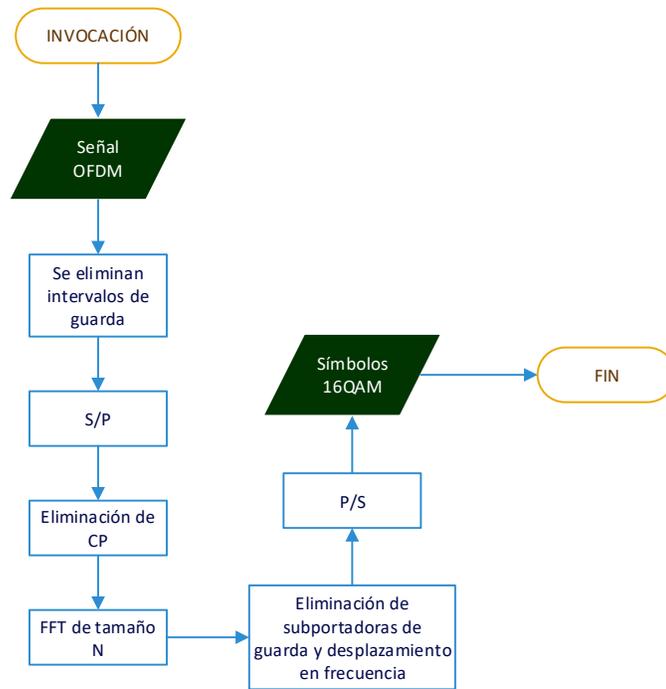


Figura 2.3 Diagrama de flujo de la demodulación OFDM.

Parámetros de entrada y salida del método `Demodulacion`

Los parámetros que utiliza el método se especifican en la Tabla 2.5. Este método trabaja con los valores almacenados en las propiedades de la clase OFDM: número de subportadoras (datos, guarda y totales), factor de normalización, desplazamiento en frecuencia, cantidad de símbolos OFDM, número de muestras del prefijo cíclico, de los intervalos de guarda y de un símbolo OFDM, descritos en la Tabla 2.2 y Tabla 2.3.

Tabla 2.5 Parámetros de entrada y salida del método `Demodulacion`.

Parámetro	Descripción
obj	Objeto al que se hace referencia. Este parámetro no se ingresa debido a que MATLAB lo hace por defecto.
SignalRX	Parámetro de entrada, corresponde al vector de símbolos OFDM.
DatosRX	Parámetro de salida, corresponde a la matriz de símbolos 16-QAM recuperados.

La señal `SignalTX` generada en el método `Modulacion` se envía al receptor OFDM a través de un canal AWGN.

Conversión serie/paralelo

El procesamiento en recepción se inicia eliminando los intervalos de guarda y convirtiendo de serie a paralelo la señal OFDM recibida `SignalRX`. Este proceso se realiza en el código 2.5.

```
% Se eliminan los intervalos de guarda y se convierte de S/P
SignalTemp = reshape(SignalRX((obj.Imp.MuestrasZG+1):(end-
obj.Imp.MuestrasZG)),obj.Imp.MuestrasSimb,obj.No.Simbolos);
```

Código 2.5. Conversión serie/paralelo.

Eliminación de CP, y FFT

Después, se elimina el prefijo cíclico y se aplica la FFT para obtener la información en frecuencia que se ha transmitido. Para realizar esta operación se emplea la función `fft()` en el código 2.6.

```
SimbolosTemp = fft(SignalTemp(obj.Imp.MuestrasCP+1:end,:)); % Se elimina
el prefijo cíclico y se realiza la FFT
```

Código 2.6. Eliminación de CP y FFT.

Eliminación de desplazamiento en frecuencia

Lo siguiente es, eliminar el desplazamiento en frecuencia y normalizar los símbolos recibidos para que su potencia media sea igual a la unidad. Este paso se realiza en el código 2.7.

```
DatosTemp =
SimbolosTemp(obj.Imp.FreqInter+(1:obj.No.TotalSubport),:)/(obj.Imp.Factor
Norm); % Se normaliza para tener potencia = 1
```

Código 2.7. Conversión serie / paralelo y FFT.

Eliminación de subportadoras de guarda, conversión paralelo/serie

Por último, se eliminan las subportadoras de guarda y se convierte de paralelo a serie los símbolos 16-QAM recibidos. De modo que, se obtiene la señal `DatosRX`, y se retorna al script para la demodulación 16-QAM y posteriormente realizar el cálculo del BER. Este paso se realiza en el código 2.8.

```
DatosRX =
reshape(DatosTemp(obj.No.SubportxBG+(1:obj.No.SubportDatos),:),[],1); %
Se convierte de P/S
```

Código 2.8. Conversión paralelo / serie, señal `DatosRX`.

Una vez que se crea la clase OFDM, se realiza el mismo procedimiento para FBMC y UFMC.

2.2 Implementación de la clase FBMC

2.2.1 Propiedades de la clase FBMC

La clase `FBMC.m` permite la creación de objetos que serán utilizados posteriormente en los scripts de Densidad Espectral de Potencia y de BER vs SNR. Las propiedades y los parámetros de entrada que se almacenan en esta clase se ilustra en la Tabla 2.6:

Tabla 2.6 Propiedades de la clase `FBMC.m`.

Propiedad	Descripción	Parámetros de entrada de la clase FBMC
No	Estructura que almacena valores adimensionales utilizados en las modulaciones.	<ul style="list-style-type: none"> - No.SubportDatos - No.SubportxBG - No.Simbolos
PHY	Estructura que almacena valores con significado físico.	<ul style="list-style-type: none"> - PHY.DeltaF - PHY.Fs
Imp	Estructura que almacena valores utilizados en la implementación de las modulaciones.	<ul style="list-style-type: none"> - Imp.FreqInter - Imp.UsarPPN
PFiltro	Esta estructura almacena valores pertenecientes al filtro prototipo.	<ul style="list-style-type: none"> - PFiltro.K

2.2.2 Métodos de la clase FBMC

Antes de definir los métodos, se crea el constructor de la clase FBMC, este constructor se encarga de almacenar los parámetros que se presentan en la Tabla 2.7 dentro de las propiedades de la Tabla 2.6

Tabla 2.7 Parámetros de entrada de la clase `FBMC.m`.

Parámetro	Descripción
No.SubportDatos	Número de subportadoras utilizadas para transportar datos.
No.SubportxBG	Número de subportadoras de cada banda de guarda.
No.Simbolos	Número de símbolos FBMC a generarse.
PHY.DeltaF	Espaciamiento entre subportadoras.
PHY.Fs	Frecuencia de muestreo.
Imp.FreqInter	Realiza un desplazamiento en frecuencia, para propósitos de visualización.
PFiltro.K	Factor de solapamiento.
Imp.UsarPPN	Valor lógico para determinar si se utiliza Red polifase o IFFT extendida.

En el código 2.9 se crea el constructor de la clase FBMC, en este código se observa que los parámetros descritos en la Tabla 2.7 se asignan a las propiedades de la clase FBMC de la Tabla 2.6.

```
% Constructor de la clase
function obj = FBMC(varargin)
    % Se inicializan los parámetros
    if numel(varargin) == 8
        obj.No.SubportDatos = varargin{1}; % Número de subportadoras de
datos
        obj.No.SubportxBG = varargin[15]; % Número de subportadoras por
banda de guarda
        obj.No.Simbolos = varargin[16]; % Número de símbolos FBMC
        obj.PHY.DeltaF = varargin{4}; % Espaciamiento entre
subportadoras (Hz)
        obj.PHY.Fs = varargin{5}; % Frecuencia de muestreo (Hz)
        obj.Imp.FreqInter = varargin[8]; % Desplazamiento en frecuencia
(para propósitos de visualización)
        obj.PFiltro.K = varargin[17]; % Factor de solapamiento
        obj.Imp.UsarPPN = varargin{8}; % Implementación de una red
polifase (PPN)

    else
        error('El número de variables de entrada debe ser 0 (valores
predeterminados) u 8');
    end
    % Configurar parámetros dependientes
    obj.ConfigParamDependientes();
end
```

Código 2.9. Constructor de la clase FBMC.

Una vez creado el constructor de la clase FBMC, se precede a definir los métodos ConfigParamDependientes, Modulacion, Demodulacion, respectivamente.

2.2.2.1 Método ConfigParamDependientes

Como se observa en el Código 2.9, al final del constructor se llama al método ConfigParamDependientes, en el que se configura los parámetros dependientes de los parámetros definidos en el constructor FBMC. La descripción de estos parámetros se especifican en la Tabla 2.8. Todos estos parámetros son definidos dentro de cada clase como se indica en el ANEXO II.

Tabla 2.8 Parámetros dependientes en la clase `FBMC.m`

Parámetro	Descripción
<code>PHY.dt</code>	Tiempo entre muestras.
<code>No.TotalSubport</code>	Número total de subportadoras, es decir, subportadoras de datos más subportadoras de cada banda de guarda.
<code>Imp.MuestrasSimb</code>	Número de muestras de un símbolo FBMC.
<code>PHY.TiempoSimb</code>	Tiempo de un símbolo FBMC.
<code>Imp.FactorSolap</code>	Factor de solapamiento.
<code>PFiltro.DominioT</code>	Respuesta al impulso del filtro prototipo.
<code>No.MuestrasPFiltro</code>	Número de muestras del filtro prototipo.
<code>No.TotalMuestras</code>	Número total de muestras a transmitirse.
<code>PFiltro.DominioF</code>	Coefficientes en frecuencia del filtro prototipo.
<code>Imp.PosdSolap</code>	Valores lógicos que determinan la posición de los símbolos FBMC durante el solapamiento.
<code>Imp.IndxdSolap</code>	Índices de los símbolos FBMC después del solapamiento.
<code>Imp.NFFT</code>	Tamaño de la IFFT.
<code>Imp.FactorNorm</code>	Factor de normalización empírico para que la potencia promedio de la señal sea igual a 1W.

2.2.2.2 Método Modulación

Así mismo, dentro de la clase `FBMC`, se define el método `Modulacion`. En este método se realiza el programa que procesa los datos en el transmisor `FBMC`. Es decir, a partir de la matriz de símbolos 16-QAM que se obtiene de los scripts principales, se realiza la modulación OQAM. En esta modulación existen dos formas de implementar `FBMC`, así que dentro del programa se deberá elegir si se desea implementar `FBMC` por IFFT extendida o Red Polifase. La diferencia entre los dos métodos es que se realiza un filtrado en el dominio de la frecuencia y el otro un filtrado en el dominio del tiempo respectivamente. Los datos modulados serán transmitidos al receptor `FBMC` a través de un canal `AWGN`. El procesamiento de la información para cada caso se ilustra el diagrama de flujo de la Figura 2.4.

Una vez definido el diagrama de flujo para la modulación `FBMC` mostrado en la Figura 2.4, se procede a analizar el procesamiento de la información previo al envío al receptor `FBMC`.

De manera similar que `OFDM`, dentro de este método se trabaja con los símbolos 16-QAM que son generados dentro de los scripts principales. El proceso para generar los datos de entrada a este método se detalla en los apartados 2.4 y 2.5.

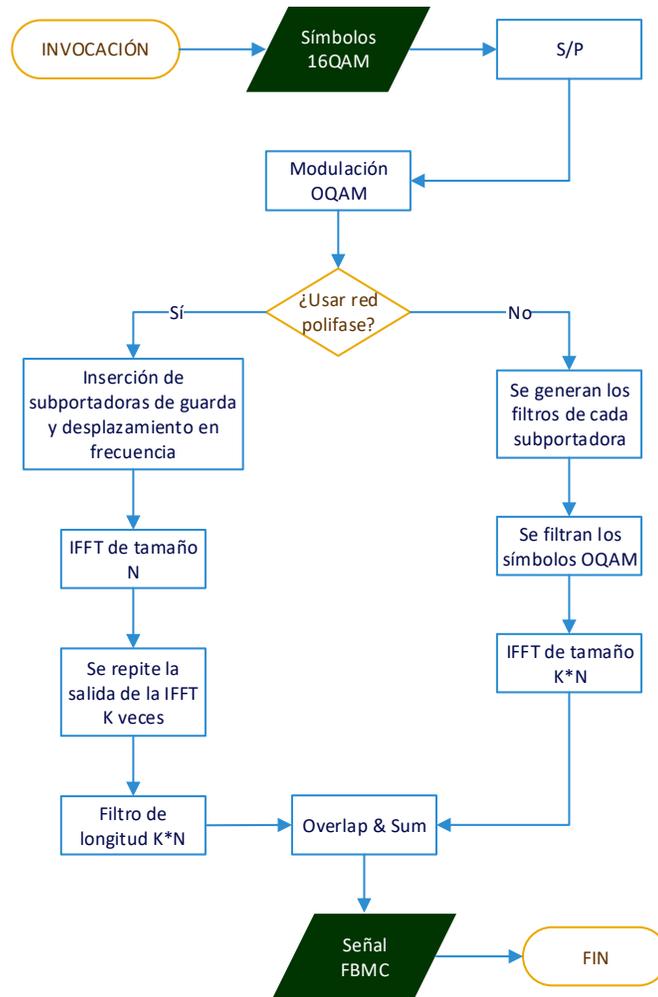


Figura 2.4 Diagrama de Flujo de la modulación FBMC.

Parámetros de entrada y salida del método Modulación

Los parámetros que utiliza este método se especifican en la Tabla 2.9. Este método trabaja con los valores almacenados en las propiedades de la clase FBMC, tales como: número de subportadoras (datos y guarda), tamaño de la IFFT, factor de normalización, desplazamiento en frecuencia, cantidad de símbolos FBMC, factor de solapamiento, número de muestras del filtro y el valor lógico para la decisión del sistema a utilizarse en la modulación. También se emplean las siguientes matrices almacenadas: filtro en el dominio del tiempo, filtro en el dominio de la frecuencia y la posición de los símbolos FBMC durante el solapamiento, descritos en la Tabla 2.7 y Tabla 2.8

Tabla 2.9 Parámetros de entrada y salida del método `Modulacion`.

Parámetro	Descripción
obj	Objeto al que se hace referencia. Este parámetro no se ingresa debido a que MATLAB lo hace por defecto.
DatosTX	Parámetro de entrada, corresponde a la matriz de símbolos 16-QAM.
SignalTX	Parámetro de salida, corresponde al vector de símbolos FBMC.

Siguiendo el orden de cada bloque del diagrama de flujo de la Figura 2.4 se procede a explicar el procesamiento de la información en el transmisor FBMC, para generar la señal `SignalTX`.

Conversión Serie/Paralelo

Como parámetro de entrada se tiene `DatosTX`, que corresponde a la matriz de símbolos 16-QAM que se generan en los scripts principales. La matriz resultante tendrá un tamaño igual a la mitad del número de subportadoras de datos por la cantidad de símbolos FBMC. Este proceso se realiza en el código 2.10.

```
Datos_S_P = reshape(DatosTX, [], obj.No.Simbolos); % Se convierte de S/P
```

Código 2.10 Conversión serie-paralelo

Modulación OQAM

Para la modulación OQAM se creó la función `oqammod`, la cual realiza la conversión de 16-QAM a OQAM. Para esta función se utiliza los parámetros de entrada y de salida que se describen en la Tabla 2.10.

Tabla 2.10 Parámetros de entrada y salida de la función `oqammod`.

Parámetro	Descripción
SimbQAM	Parámetro de entrada, corresponde a la matriz de símbolos 16-QAM.
SimbOQAM	Parámetro de salida, corresponde a la matriz de símbolos OQAM.

El proceso para esta modulación se presenta en el diagrama de flujo que se ilustra en la Figura 2.5.

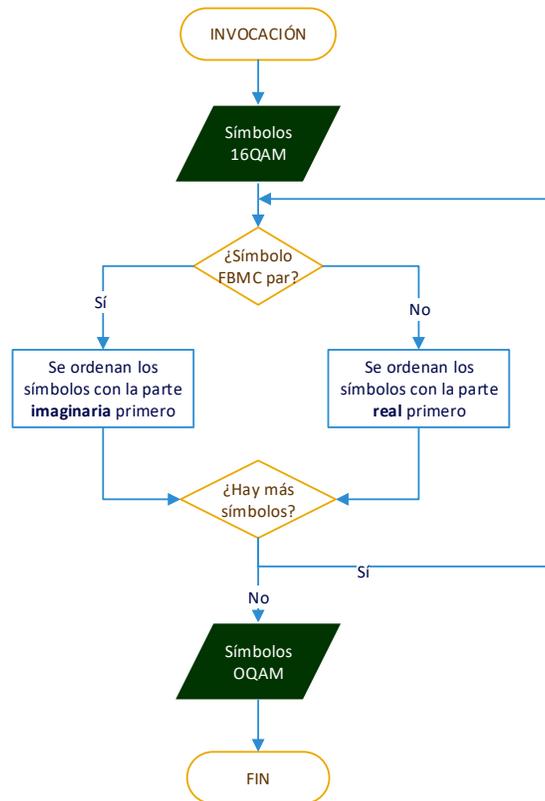


Figura 2.5 Diagrama de Flujo de la modulación OQAM.

Cabe señalar que este procedimiento es importante realizarlo, ya que garantiza ortogonalidad entre subportadoras adyacentes. A continuación se describe el funcionamiento de la función `oqammod`.

Mediante esta función se separa la parte real e imaginaria de los símbolos 16-QAM y los ordena dependiendo si el símbolo FBMC al que pertenecen es par o impar. Este proceso se detalla en el diagrama de flujo de la Figura 2.5. y se realiza en el código 2.11.

```

for i_simb = 1:K
    if rem(i_simb,2)==1 % Símbolos impares
        SimbOQAM(1:2:end,i_simb) = real(SimbQAM(:,i_simb));
        SimbOQAM(2:2:end,i_simb) = 1i*imag(SimbQAM(:,i_simb));
    else % Símbolos pares
        SimbOQAM(1:2:end,i_simb) = 1i*imag(SimbQAM(:,i_simb));
        SimbOQAM(2:2:end,i_simb) = real(SimbQAM(:,i_simb));
    end
end
end
  
```

Código 2.11. Función `oqammod`.

Como en este proceso se duplica el número de símbolos, se tiene que garantizar que la cantidad de bits generados correspondan a la mitad de los bits que se necesitan en una modulación OFDM.

Continuando en el diagrama de flujo de la Figura 2.4, los datos que se convirtieron en paralelo se modulan con la función `oqammod`, esto se realiza en el código 2.12

```
DatosOQAM = oqammod(Datos_S_P); % Se realiza la modulación OQAM
```

Código 2.12 Modulación OQAM.

En este punto, se decide la forma que se utilizará para implementar la modulación FBMC, como se mencionó anteriormente existen dos formas: IFFT extendida o Red polifase. Para elegir cuál de las dos formas anteriormente mencionadas, se hará uso de una sentencia *if* y el valor lógico `Imp.UsarPPN` de la Tabla 2.7.

IFFT Extendia

Si se elige utilizar IFFT extendida, lo siguiente es crear una matriz que contenga los filtros para cada subportadora. Estos filtros se generan a partir del filtro prototipo mediante los coeficientes de frecuencia presentados en la tabla 1.2. Los coeficientes del filtro prototipo se encuentra almacenados en las propiedades de la clase FBMC, en `PFiltro.DominioF`, como se explica a continuación.

Filtro Prototipo

El filtro prototipo a utilizarse en la simulación es el filtro PHYDYAS. Para generar los filtros para cada subportadora, dentro de la clase se implementó la función local `FiltroPrototipo_PHYDYAS`, cuyos parámetros se detallan en la Tabla 2.11.

Tabla 2.11 Parámetros de entrada y salida de la función local `FiltroPrototipo_PHYDYAS`.

Parámetro	Descripción
T0	Parámetro de entrada, corresponde a la duración de un símbolo FBMC.
dt	Parámetro de entrada, corresponde, al tiempo entre muestras.
OF	Parámetro de entrada, corresponde al factor de solapamiento.
PFiltro	Parámetro de salida, corresponde a la respuesta al impulso del filtro prototipo.

Mediante esta función se obtiene la respuesta al impulso del filtro dependiendo del factor de solapamiento K . Para ello se trabaja con los coeficientes en frecuencia del filtro prototipo de Phydias especificados en la Tabla 1.2 del capítulo 1. Para obtener la respuesta al impulso se utilizó la Ecuación 1.9.

La programación de la función local `FiltroPrototipo_PHYDYAS` se muestra en el código 2.13.

```
function PFiltro = FiltroPrototipo_PHYDYAS(T0,dt,OF)
    t_fil = -(OF*T0/2):dt:(OF*T0/2-dt);
    switch OF % Coeficientes del filtro prototipo en el dominio de la
    frecuencia de acuerdo al factor de solapamiento
        case 2
            H = sqrt(2)/2;
        case 3
            H = [0.91143783 0.41143783];
        case 4
            H = [0.97195983 sqrt(2)/2 0.23514695];
        otherwise
            error('El factor de solapamiento debe ser un entero entre 2 y
    4');
    end
    % Filtro Prototipo en el dominio del tiempo (Respuesta al impulso)
    PFiltro=
    1+2*sum(repmat(H,length(t_fil),1).*cos(2*pi*repmat(t_fil',1,length(H)).*r
    epmat(1:length(H),length(t_fil),1)/((length(H)+1)*T0)),2);
end
```

Código 2.13. Función local `FiltroPrototipo_PHYDYAS`

Con la respuesta al impulso que se obtuvo luego de aplicar la ecuación 1.9, se calcula la respuesta en frecuencia del filtro aplicando la transformada de Fourier mediante la función `fft()`. Este paso se realiza en el código 2.14.

```
PFiltroFFT =
fft(circshift(obj.PFiltro.DominioT,obj.No.MuestrasPFiltro/2)); %
Respuesta en frecuencia del filtro prototipo
PFiltroFFT(abs(PFiltroFFT)./PFiltroFFT(1)<10^-14) = 0; % Para reducir la
complejidad computacional
obj.PFiltro.DominioF = PFiltroFFT/max(PFiltroFFT); % Coeficientes en
frecuencia del filtro prototipo
```

Código 2.14. Respuesta en frecuencia del filtro PHYDYAS

Generación de filtros para cada subportadora

Cómo se está implementando FBMC mediante IFFT extendida, se debe realizar el filtrado en el dominio de la frecuencia. Para esto, se crea una matriz que contenga los filtros para cada subportadora, los cuales se generan a partir del filtro en el dominio de la frecuencia que se encuentra almacenado en las propiedades de la clase FBCM. Además del filtrado, esta matriz añade las subportadoras de guarda, realiza un desplazamiento en frecuencia y aumenta la frecuencia de muestreo. Este proceso se realiza en el código 2.15.

```

for i_1 = 1:obj.No.SubportDatos
    % Se generan los filtros para cada subportadora a
    % partir de los coeficientes del filtro prototipo
    GenPFiltros(:,i_1) = circshift(obj.PFiltro.DominioF,
obj.Imp.FactorSolap*(i_1-1+obj.Imp.FreqInter+obj.No.SubportxBG));
end

```

Código 2.15. Generación de los filtros de cada subportadora

Filtrado de símbolos OQAM

Con la matriz de filtros anterior se realiza el filtrado de los símbolos OQAM. Este proceso se realiza mediante una multiplicación matricial entre la matriz de filtros y la matriz de símbolos OQAM. Además, los símbolos resultantes se normalizan para que la señal FBMC tenga una potencia media igual a la unidad. Este proceso se realiza en el código 2.16.

```

SignalTemp =
GenPFiltros*DatosOQAM*(obj.Imp.FactorNorm*obj.Imp.FactorSolap); % Se
filtra y normaliza para tener potencia = 1

```

Código 2.16. Proceso de filtrado

IFFT de tamaño N*K y solapamiento

Continuando con el procesamiento, los símbolos filtrados ingresan a la IFFT de tamaño $K*N$. A la salida de este bloque se solapan los símbolos FBMC generados de la siguiente manera: Los símbolos FBMC se ubican con un retraso de $N/2$ muestras entre ellos. Para implementar el solapamiento se hace uso de la matriz `Imp.PosdSolap`. Este paso se realiza en el código 2.17

```

% Se realiza la IFFT extendida y se solapan los símbolos
SignalTX(obj.Imp.PosdSolap) = circshift(iffft(SignalTemp), [-
obj.No.MuestrasPFiltro/2 0]);

```

Código 2.17. IFFT y solapamiento.

Suma de símbolos solapados

Por último, se suman los símbolos solapados y se obtiene la señal FBMC, `SignalTX`. Esta señal se retorna al script para su procesamiento. Este paso se realiza en el código 2.18.

```

SignalTX = sum(SignalTX,2); % Se suman los símbolos solapados

```

Código 2.18. Obtención de la señal FBMC.

Ahora bien, si se escoge la segunda forma de implementar FBMC, es decir, mediante red Polifase, el procesamiento es diferente debido a que en éste método se realiza el filtrado en el dominio del tiempo. A continuación se procede a describir el procedimiento para la implementación de FBMC mediante red Polifase.

Red Polifase

Inserción de subportadoras de guarda y desplazamiento en frecuencia

Ahora bien, si se decide utilizar Red polifase, después de la modulación OQAM, el siguiente paso es añadir las subportadoras de guarda y realizar un desplazamiento en frecuencia. Del mismo modo, los símbolos 16-QAM se normalizan para que la señal FBMC tenga una potencia media igual a la unidad. Este paso se realiza en el código 2.19.

```
DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.SubportDatos), :)  
= DatosOQAM*obj.Imp.FactorNorm; % Se normaliza para tener potencia = 1
```

Código 2.19. Inserción de subportadoras de guarda.

IFFT de tamaño N y Red Polifase

Después, se realiza la IFFT de tamaño N, la salida de la IFFT se replica K veces y se multiplica elemento por elemento por el filtro en el dominio del tiempo de longitud K*N almacenado en las propiedades de la clase FBMC. Por último, se solapan los símbolos FBMC obtenidos con un retraso entre símbolos de N/2 muestras. Este proceso se realiza en el código 2.20.

```
% Se realiza la IFFT, se filtra y se solapan los símbolos  
SignalTX(obj.Imp.PosdSolap) = repmat(iff(DatosTemp), [obj.Imp.FactorSolap  
1]).*repmat(obj.PFiltro.DominioT, [1 obj.No.Simbolos]);
```

Código 2.20. IFFT y Red polifase.

Suma de símbolos solapados

Finalmente, se suman los símbolos y se obtiene la señal FBMC ,SignalTX, y esta se retorna al script para su procesamiento. Es importante mencionar que, este proceso es el mismo que se realiza al implementar FBMC mediante el método de IFFT extendida, como se observa en el Código 2.18.

2.2.2.3 Método Demodulación

El método `Demodulación` realiza un proceso inverso al método `Modulación`. Así mismo, dependiendo si se realizó la implementación de FBMC mediante IFFT extendida o Red Polifase, el receptor irá realizando de forma contraria el procesamiento de la señal. Este proceso se detalla en el diagrama de flujo de la Figura 2.6.

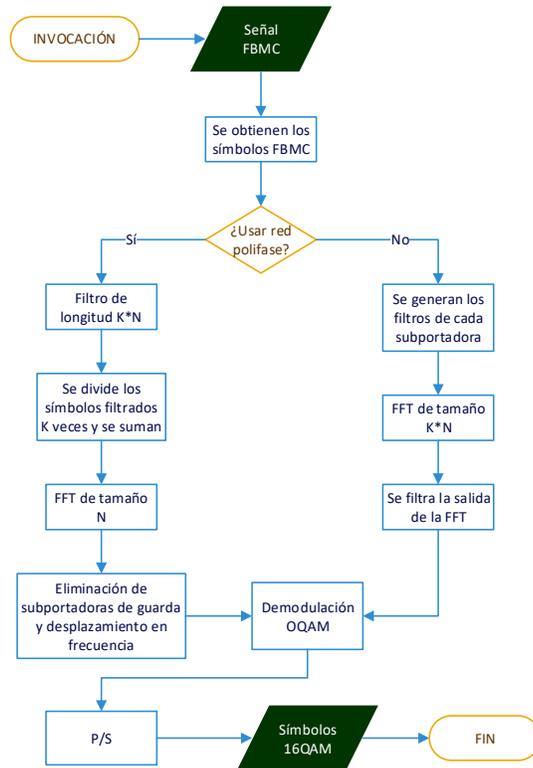


Figura 2.6 Diagrama de Flujo de la demodulación FBMC.

Los parámetros que utiliza el método se especifican en la Tabla 2.12. Este método utiliza los valores almacenados en las propiedades de la clase FBMC: número de subportadoras (datos y guarda), tamaño de la IFFT, factor de normalización, desplazamiento en frecuencia, cantidad de símbolos FBMC, factor de solapamiento, número de muestras del filtro y el valor lógico para la decisión del sistema a utilizarse en la modulación. También se emplean las siguientes matrices almacenadas: filtro en el dominio del tiempo, filtro en el dominio de la frecuencia y los índices de los símbolos FBMC después del solapamiento, descritas en la Tabla 2.7 y la Tabla 2.8.

Tabla 2.12 Parámetros de entrada y salida del método `Demodulacion`.

Parámetro	Descripción
obj	Objeto al que se hace referencia. Este parámetro no se ingresa debido a que MATLAB lo hace por defecto.
SignalRX	Parámetro de entrada, corresponde al vector de símbolos FBMC.
DatosRX	Parámetro de salida, corresponde a la matriz de símbolos 16-QAM recuperados.

Para explicar cada bloque del diagrama de flujo de la Figura 2.6, se parte de los símbolos FBMC de la señal recibida. Para esto se hace uso de la matriz `Imp.IndxdSolap`, la cual contiene los índices de los símbolos FBMC que se encuentran solapados dentro de la señal recibida. Este paso se realiza en el código 2.21.

```
Signal_SimbFBMC = SignalRX(obj.Imp.IndxdSolap); % Se obtienen los
símbolos FBMC de la señal recibida (S/P)
```

Código 2.21. Obtención de los símbolos FBMC en recepción.

Asimismo, se debe elegir el sistema a implementar, es decir, FFT extendida o Red polifase. El procesamiento en demodulación debe, necesariamente coincidir con el método que se escogió para implementar en la modulación. Para la elección se hace uso de una sentencia *if* y del valor lógico `Imp.UsarPPN`.

Con esta aclaración, se procede a explicar el procesamiento que se realiza si en modulación se escogió implementar FBMC con IFFT extendida, en el método `demodulacion` debe realizar el proceso inverso, como se explica a continuación.

IFFT Extendida

Filtros de cada subportadora y FFT de tamaño K*N

Para el sistema IFFT extendida, lo primero es crear la matriz de filtros a partir del filtro almacenado en las propiedades de la clase como se indica en el Código 2.15. Después se aplica la FFT de tamaño K*N a los símbolos FBMC. Este paso se realiza en el código 2.22

```
DatosTemp = fft(circshift(Signal_SimbFBMC,[obj.No.MuestrasPFiltro/2 0]));
% Se realiza la FFT extendida
```

Código 2.22. Aplicación de la FFT.

Filtrado de la salida de la FFT, supresión de las subportadoras de guarda

La salida de la FFT es filtrada para obtener los símbolos OQAM. Este proceso se realiza mediante una multiplicación matricial entre la transpuesta de la matriz de filtros y la salida de la FFT. Durante este proceso se eliminan las subportadoras de guarda y el desplazamiento en frecuencia. También se normalizan los símbolos OQAM obtenidos. Este paso se realiza en el código 2.23:

```
DatosOQAM =
(GenPFiltros'*DatosTemp)/(obj.Imp.FactorNorm*obj.Imp.FactorSolap^2); % Se
filtra y normaliza para tener potencia = 1
```

Código 2.23. Proceso de filtrado.

Demodulación OQAM

Para la demodulación OQAM se creó la función `oqamdemod`, la cual realizará la conversión de OQAM a 16-QAM. Este proceso es común para los dos métodos implementados, es decir, se aplica la misma demodulación OQAM, tanto en IFFT extendida como en Red Polifase. A continuación se presenta en diagrama de flujo de la función `oqamdemod` en la Figura 2.7

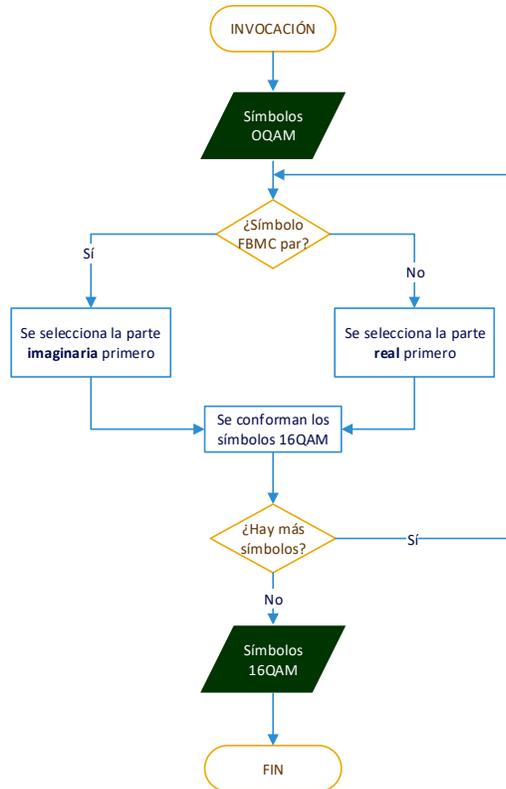


Figura 2.7 Diagrama de Flujo de la demodulación OQAM.

Así mismo esta función `oqamdemod` requiere de parámetros que están descritos en la Tabla 2.13.

Tabla 2.13 Parámetros de entrada y salida de la función `oqamdemod`.

Parámetro	Descripción
SimbOQAM	Parámetro de entrada, corresponde a la matriz de símbolos OQAM recuperados.
SimbQAM	Parámetro de salida, corresponde a la matriz de símbolos 16-QAM recuperados.

La función `oqamdemod` va eligiendo alternadamente la parte real o imaginaria de los símbolos OQAM y el orden de selección depende si el símbolo FBMC recibido es par o

impar. Por último, se conforman los símbolos 16-QAM con las partes reales e imaginarias obtenidas. Este proceso se detalla en el diagrama de flujo de la Figura 2.7. En el Código 2.24 se indica la parte principal de la función `oqamdemod`.

```

for i_simb = 1:K
    if rem(i_simb,2)==1 % Símbolos impares
        rQAM = real(SimboOQAM(1:2:end,i_simb));
        iQAM = imag(SimboOQAM(2:2:end,i_simb));
    else % Símbolos pares
        iQAM = imag(SimboOQAM(1:2:end,i_simb));
        rQAM = real(SimboOQAM(2:2:end,i_simb));
    end
    SimboQAM(:,i_simb)=complex(rQAM,iQAM);
end

```

Código 2.24. Demodulación OQAM.

Por otro lado, si en la transmisión se escogió el método FBMC mediante Red Polifase, se debe realizar el procedimiento inverso, como se describe a continuación.

RED POLIFASE

Filtrado

Para el sistema Red polifase, lo primero es realizar el filtrado de los símbolos FBMC. El filtrado se realiza mediante la multiplicación elemento por elemento de los símbolos FBMC por el filtro en el dominio del tiempo que se encuentra almacenado en `PFiltro.DominioT`.

División de símbolos filtrados K veces y suma, FFT de tamaño N

Los símbolos filtrados se dividen K veces y se suman. Después se aplica la FFT de tamaño N. A continuación se muestra en el código 2.25 la ejecución de este procesamiento.

```

SignalFiltrada = Signal_SimbFBMC.*repmat(obj.PFiltro.DominioT,[1
obj.No.Simbolos]); % Se filtra los símbolos recibidos

DatosTemp = fft(sum(reshape(SignalFiltrada,[obj.Imp.NFFT
obj.Imp.FactorSolap obj.No.Simbolos]),2)); % Se realiza la FFT

```

Código 2.25. Red polifase y FFT.

Eliminación de subportadoras de guarda y desplazamiento en frecuencia

El siguiente paso es eliminar las subportadoras de guarda y el desplazamiento en frecuencia. Además, se normalizan los símbolos OQAM obtenidos. Este proceso se realiza en el código 2.26.

```
% Se normaliza para tener potencia = 1
DatosOQAM =
DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.SubportDatos),:)/(
(obj.Imp.FactorNorm*obj.Imp.FactorSolap^2);
```

Código 2.26. Eliminación de subportadoras de guarda.

A continuación, independientemente del sistema utilizado, se realiza la demodulación OQAM obteniendo una matriz de símbolos 16-QAM de tamaño igual a la mitad del número de subportadoras de datos por la cantidad de símbolos FBMC. Este proceso se realiza en el código 2.27.

```
Datos_P_S = oqamdemod(DatosOQAM); % Se realiza la demodulación OQAM
```

Código 2.27. Demodulación OQAM.

Conversión de paralelo/serie

Por último, se convierte de paralelo a serie los símbolos 16-QAM recibidos. Estos símbolos `DatosRX` se retornan al script para la demodulación 16-QAM y posteriormente realizar el cálculo del BER. Este proceso se realiza en el código 2.28:

```
DatosRX = Datos_P_S(:); % Se convierte de P/S
```

Código 2.28. Conversión paralelo / serie.

2.3 Implementación de la clase UFMC

2.3.1 Propiedades de la clase UFMC

La clase `UFMC.m` permite la creación de objetos que serán utilizados posteriormente en los scripts de Densidad Espectral de Potencia y de BER vs SNR. Las propiedades y los parámetros de entrada que se almacenan en esta clase se ilustra en la Tabla 2.14.

Tabla 2.14 Propiedades de la clase `UFMC.m`.

Propiedad	Descripción	Parámetros de entrada de la clase UFMC
No	Estructura que almacena valores adimensionales utilizados en las modulaciones.	- No.SubportDatos - No.SubportxBG - No.Simbolos
PHY	Estructura que almacena valores con significado físico.	- PHY.DeltaF - PHY.Fs - PHY.TiempoZG - PHY.LongFiltro
Imp	Estructura que almacena valores utilizados en la implementación de las modulaciones.	- Imp.FreqInter

2.3.2 Métodos de la clase UPMC

Antes de definir los métodos, se crea el constructor de la clase UPMC, este constructor se encarga de almacenar los parámetros que se presentan en la Tabla 2.15 dentro de las propiedades de la Tabla 2.14

Tabla 2.15 Parámetros de entrada de la clase UPMC.m

Parámetro	Descripción
No.SupportDatos	Número de subportadoras utilizadas para transportar datos.
No.SupportxBG	Número de subportadoras de cada banda de guarda.
No.Simbolos	Número de símbolos UPMC a generarse.
PHY.DeltaF	Espaciamiento entre subportadoras.
PHY.Fs	Frecuencia de muestreo.
Imp.FreqInter	Permite realizar un desplazamiento en frecuencia, solo para propósitos de visualización.
PHY.TiempoZG	Duración de los intervalos de guarda en segundos.
PHY.LongFiltro	Duración del filtro en segundos.

En el código 2.29 se crea el constructor de la clase UPMC, en este código se observa que los parámetros descritos en la Tabla 2.14 se asignan a las propiedades de la clase UPMC.

```
% Constructor de la clase
function obj = UPMC(varargin)
    % Se inicializan los parámetros
    if numel(varargin) == 8
        obj.No.SupportDatos = varargin{1}; % Número de subportadoras de
datos
        obj.No.SupportxBG = varargin[15]; % Número de subportadoras por
banda de guarda
        obj.No.Simbolos = varargin[16]; % Número de símbolos UPMC
        obj.PHY.DeltaF = varargin{4}; % Espaciamiento entre
subportadoras (Hz)
        obj.PHY.Fs = varargin{5}; % Frecuencia de muestreo (Hz)
        obj.Imp.FreqInter = varargin[8]; % Desplazamiento en frecuencia
(para propósitos de visualización)
        obj.PHY.TiempoZG = varargin[17]; % Duración de los intervalos
de guarda (s)
        obj.PHY.LongFiltro = varargin{8}; % Longitud del filtro (s)

    else
        error('El número de variables de entrada debe ser 0 (valores
predeterminados) u 8');
    end
    % Configurar parámetros dependientes
    obj.ConfigParamDependientes();
end
```

Código 2.29. Constructor de la clase UPMC.

2.3.2.1 Método ConfigParamDependientes

Como se observa en el Código 2.29, al final del constructor se llama al método `ConfigParamDependientes`, en el que se configura los parámetros dependientes de los parámetros definidos en el constructor `UFMC`. La descripción de estos parámetros se especifican en la Tabla 2.16. Todos estos parámetros son definidos dentro de cada clase como se indica en el ANEXO III.

Tabla 2.16 Parámetros restantes de la clase `UFMC.m`.

Parámetro	Descripción
No.TotalSubport	Número total de subportadoras, es decir, subportadoras de datos más subportadoras de cada banda de guarda.
Imp.MuestrasZG	Número de muestras de los intervalos de guarda.
Imp.NFFT	Tamaño de la IFFT.
Imp.FactorNorm	Factor de normalización empírico para que la potencia promedio de la señal sea igual a 1W.
PHY.dt	Tiempo entre muestras.
Imp.LongFiltro	Longitud del filtro en muestras.
No.LongSubBanda	Número de subportadoras por subbanda.
No.NoSubBandas	Número de subbandas.
Imp.RespImpFiltro	Respuesta al impulso de los filtros de cada subbanda.
Imp.MuestrasSimb	Número de muestras de un símbolo UFMC.
PHY.TiempoSimb	Tiempo de un símbolo UFMC.
No.TotalMuestras	Número total de muestras a transmitirse.
Imp.FiltroEqRX	Función de transferencia del filtro de ecualización en recepción.

2.3.2.2 Método Modulacion

Para realizar la modulación UFMC se creó el método `Modulacion`. En el diagrama de flujo que se ilustra en la Figura 2.8, se observa que la IFFT se aplica al conjunto de símbolos correspondientes a una determinada subbanda. Después de ser filtradas, finalmente son enviadas al canal de manera conjunta. Estos datos modulados serán transmitidos al receptor UFMC a través de un canal AWGN.

En el diagrama de flujo que se ilustra en la Figura 2.8 se indica el proceso que sigue la información en un transmisor UFMC.

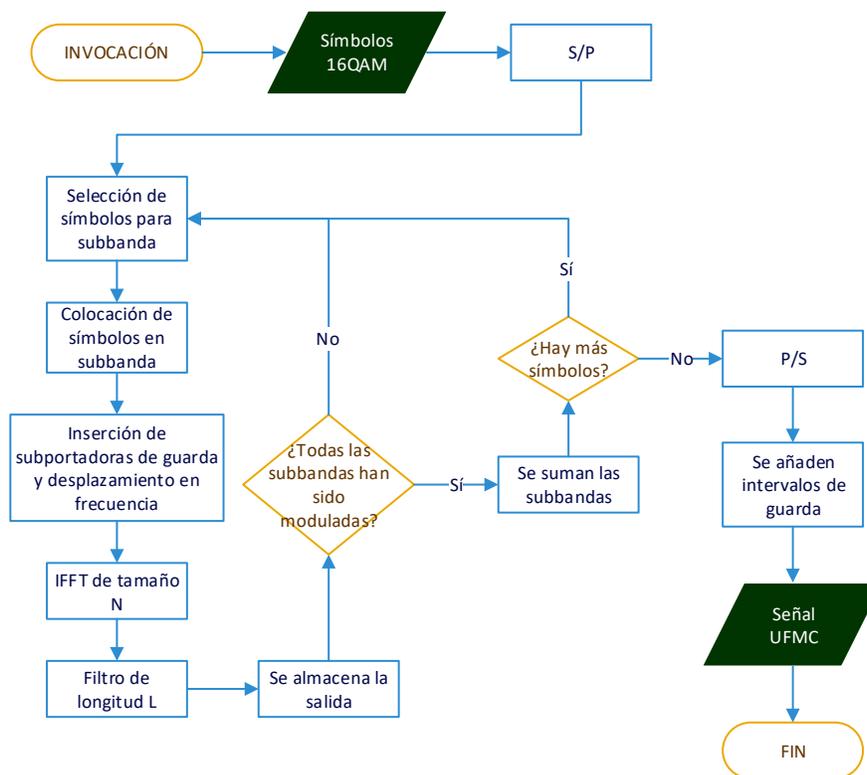


Figura 2.8 Diagrama de Flujo de la modulación UFMC.

Una vez definido el diagrama de flujo para la modulación UFMC mostrado en la Figura 2.8, se procede a analizar el procesamiento de la información previo al envío al receptor UFMC.

Es importante señalar que los datos que ingresan al modulador se obtienen de los scripts principales (Densidad de Potencia ó BER vs SNR), es decir, en los scripts principales se encuentra el proceso de generación de datos aleatorios y modulación mediante 16-QAM. De modo que, dentro de este método se trabaja con los símbolos 16-QAM que son generados dentro de los escripts principales. El proceso para generar los datos de entrada a este método se detalla los apartados 2.4 y 2.5.

Parámetros de entrada y salida del método Modulación

Este método trabaja con los valores almacenados en las propiedades de la clase UFMC, como son: número de subportadoras de guarda, tamaño de la IFFT, factor de normalización, desplazamiento en frecuencia, cantidad de símbolos UFMC, número de muestras de los intervalos de guarda, longitud del filtro, número de subbandas y longitud de las subbandas. También se emplea la matriz `Imp.RespImpFiltro`, la cual contiene los filtros de cada subbanda. Los parámetros de entrada y salida del método Modulación se indican en la tabla 2.17

Tabla 2.17 Parámetros de entrada y salida del método `Modulacion`.

Parámetro	Descripción
obj	Objeto al que se hace referencia. Este parámetro no se ingresa debido a que MATLAB lo hace por defecto.
DatosTX	Parámetro de entrada, corresponde a la matriz de símbolos 16-QAM.
SignalTX	Parámetro de salida, corresponde al vector de símbolos UPMC.

Conversión serie/paralelo

Primero se convierte de serie a paralelo los símbolos 16-QAM de entrada, que provienen de los scripts principales. La matriz resultante tendrá un tamaño igual al número de subportadoras de datos por cantidad de símbolos UPMC. Además, los símbolos 16-QAM se normalizan para que la señal UPMC tenga una potencia media igual a la unidad. Este proceso se realiza en código 2.30.

```
Datos_S_P = reshape(DatosTX, [], obj.No.Simbolos)*obj.Imp.FactorNorm; % Se  
convierte de S/P y se normaliza para tener potencia = 1
```

Código 2.30. Conversión serie-paralelo.

Selección de símbolos para subbanda

Después, se calculan los índices de los símbolos 16-QAM que corresponden a la subbanda *i*-ésima.

Colocación de símbolos en subbanda e inserción de subportadoras de guarda

Luego, con ayuda de estos índices se colocan los símbolos 16-QAM en la subbanda correspondiente teniendo en cuenta las subportadoras de guarda y el desplazamiento en frecuencia.

IFFT

Inmediatamente se realiza la IFFT de tamaño *N*. A continuación, se realiza el filtrado mediante una convolución entre la salida de la IFFT y los coeficientes del filtro en tiempo, almacenando el resultado en una matriz temporal. Una vez obtenidas todas las subbandas, se suman para obtener el símbolo UPMC.

El proceso correspondiente a los puntos descritos anteriormente, se muestra en el código 2.31.

```

for i_subband = 1:obj.No.NoSubBandas
    DatosTemp = zeros(obj.Imp.NFFT,1);
    % Se seleccionan las subportadoras de la subbanda
    SubbandIdx = (1:obj.No.LongSubBanda)+(i_subband-
1)*obj.No.LongSubBanda;
    DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+SubbandIdx) =
Datos_S_P(SubbandIdx,i_k); % Se añaden las subportadoras de guarda
    SignalTemp = ifft(DatosTemp); % Se realiza la IFFT
    SignalTX_Temp(:,i_subband) =
conv(SignalTemp,obj.Imp.RespImpFiltro(:,i_subband)); % Se filtra la
subbanda
end
SignalTX_Temp2(:,i_k) = sum(SignalTX_Temp,2); % Se suma y obtiene el
símbolo UPMC

```

Código 2.31. Modulación UPMC.

Filtro

Para el filtrado de las subbandas se utiliza un filtro (o ventana) *Dolph-Chevyshev*, con atenuación de 60 dB en los lóbulos secundarios de su respuesta en frecuencia. `FiltroTX_Tiempo` y `FiltroTX_Freq` representan el filtro base, del cual se obtienen los filtros para cada subbanda. Este proceso se realiza en el código 2.27.

```

% Filtrado de subbanda con ventana Dolph-Chebyshev
FiltroTX_Tiempo = chebwin(obj.Imp.LongFiltro,60); % Filtro base en tiempo
FiltroTX_Freq = fft(FiltroTX_Tiempo,obj.Imp.NFFT); % Filtro base en
frecuencia

```

Código 2.27. Filtro Dolph-Chevyshev.

Conversión paralelo/serie

El proceso anterior se repite hasta obtener el último símbolo UPMC. Por último, se convierte de paralelo a serie los símbolos UPMC generados, se añaden intervalos de guarda y se retorna la señal al script para su procesamiento. Este proceso se realiza en el código 2.30.

```

% Se convierte de P/S y se añaden los intervalos de guarda
SignalTX =
[zeros(obj.Imp.MuestrasZG,1);reshape(SignalTX_Temp2,[],1);zeros(obj.Imp.M
uestrasZG,1)];

```

Código 2.30. Conversión paralelo / serie

2.3.2.3 Método Demodulación

En este método la señal que se recibe en el receptor UPMC después de pasar por el canal al que se le agregó ruido AWGN, se procede a eliminar los intervalos de guarda, se le aplica la IFFT de tamaño $2N$, se convierte de serie a paralelo, se realiza diezmado de factor 2, se eliminan las subportadoras de guarda, se ecualiza, se convierte de paralelo a serie y los símbolos 16-QAM son enviados a los scripts

principales para su procesamiento. Este proceso se realiza mediante el método Demodulacion.

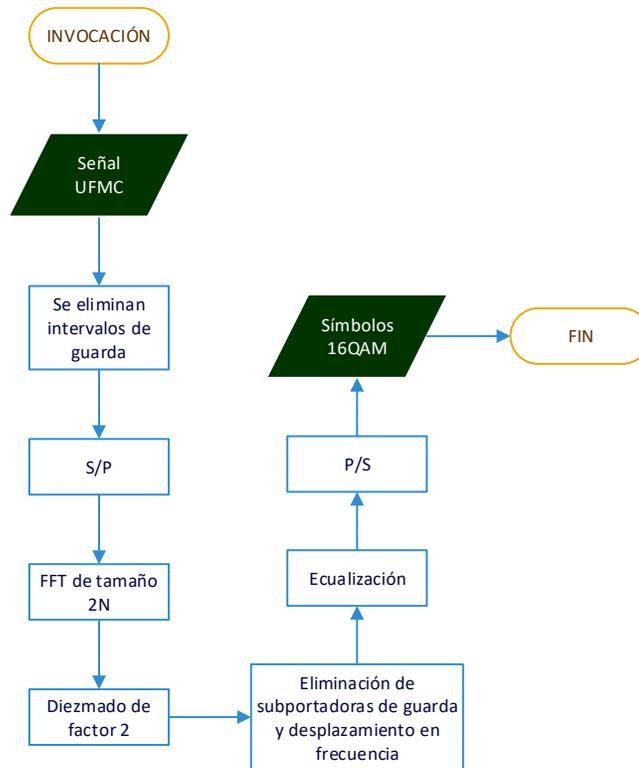


Figura 2.9 Diagrama de Flujo de la demodulación UFMC.

Parámetros de entrada y salida del método Demodulacion

Los parámetros que utiliza el método se especifican en la Tabla 2.18. Además, este método utiliza los siguientes valores almacenados en las propiedades de la clase: número de subportadoras (guarda y totales), tamaño de la IFFT, factor de normalización, desplazamiento en frecuencia, cantidad de símbolos UFMC, longitud de las subbandas, número de muestras de los intervalos de guarda y de un símbolo UFMC. También se emplea la matriz Imp.FiltroEqRX, la cual contiene los filtros de ecualización para cada subbanda.

Tabla 2.18 Parámetros de entrada y salida del método Demodulacion.

Parámetro	Descripción
Obj	Objeto al que se hace referencia. Este parámetro no se ingresa debido a que MATLAB lo hace por defecto.
SignalRX	Parámetro de entrada, corresponde al vector de símbolos UFMC.
DatosRX	Parámetro de salida, corresponde a la matriz de símbolos 16-QAM recuperados.

Eliminación de intervalos de guarda, FFT 2N, Diezmado

El primer paso es eliminar los intervalos de guarda y convertir de serie a paralelo la señal UFMC recibida. Después se realiza la FFT, pero se duplica su tamaño, es decir, 2N. A continuación, se tiene que realizar un diezmado de factor 2 de la salida de la FFT para obtener las subportadoras requeridas. Este proceso se realiza en el código 2.31.

```
SignalTemp = reshape(SignalRX((obj.Imp.MuestrasZG+1):(end-obj.Imp.MuestrasZG)),obj.Imp.MuestrasSimb,obj.No.Simbolos);
SimbTemp = fft(SignalTemp,obj.Imp.NFFT*2); % Se realiza la FFT de tamaño 2N
SimbTemp = SimbTemp(1:2:end,:); % Se realiza un diezmado de factor 2
```

Código 2.31. Conversión serie / paralelo, FFT y diezmado.

Eliminación de desplazamiento en frecuencia

Una vez realizado el diezmado, se elimina el desplazamiento en frecuencia realizado en transmisión. Además, se normalizan los símbolos 16-QAM recibidos para que su potencia media sea igual a la unidad. Este proceso se realiza en el código 2.32.

```
DatosTemp =
SimbTemp(obj.Imp.FreqInter+(1:obj.No.TotalSubport),:)/(obj.Imp.FactorNorm); % Se normaliza para tener potencia = 1
```

Código 2.32. Eliminación del desplazamiento en frecuencia.

Eliminación de las subportadoras de guarda y Ecuación

En este punto, se tienen símbolos distorsionados, razón por la cual se aplica un proceso de ecualización para compensar los efectos de los filtros, pero primero se tiene que eliminar las subportadoras de guarda. La ecualización se realiza por subbandas con el inverso de los filtros en frecuencia utilizados en transmisión. Este proceso se realiza en el código 2.33.

```
% Se equalizan los símbolos QAM recibidos
DatosEq = reshape(DatosTemp(obj.No.SubportxBG+1:end-obj.No.SubportxBG,:),obj.No.LongSubBanda,[]).*obj.Imp.FiltroEqRX;
```

Código 2.33. Ecuación de los símbolos 16-QAM recibidos

Conversión paralelo/serie

Por último, se convierte de paralelo a serie los símbolos 16-QAM ecualizados. Estos símbolos se retornan al script para su demodulación y posteriormente realizar el cálculo del BER. Este paso se realiza en el código 2.34.

```
DatosRX = DatosEq(:); % Se convierte de P/S
```

Código 2.34. Conversión paralelo-serie.

Una vez que se crearon las clases para cada sistema, se procede a describir la creación de los scripts principales, que son: Densidad Espectral de Potencia y BER vs SNR.

Una vez terminado de crear las clases OFDM, FBMC, y UFMC se crea los scripts principales, los cuales generan los bits aleatorios a transmitir y los modulan mediante 16-QAM. En estos scripts se llaman a las clases para realizar los procesos de modulación y demodulación respectivamente. En estos scripts se obtiene las gráficas de DSP y las curvas de BER vs SNR.

2.4 Densidad espectral de potencia

El script `Power_Spectral_Density.m` es utilizado para la obtención de las gráficas de DSP y eficiencia espectral de las diferentes técnicas de modulación. En el diagrama de flujo que se ilustra en la Figura 2.10 se indica el proceso para la obtención de las gráficas.

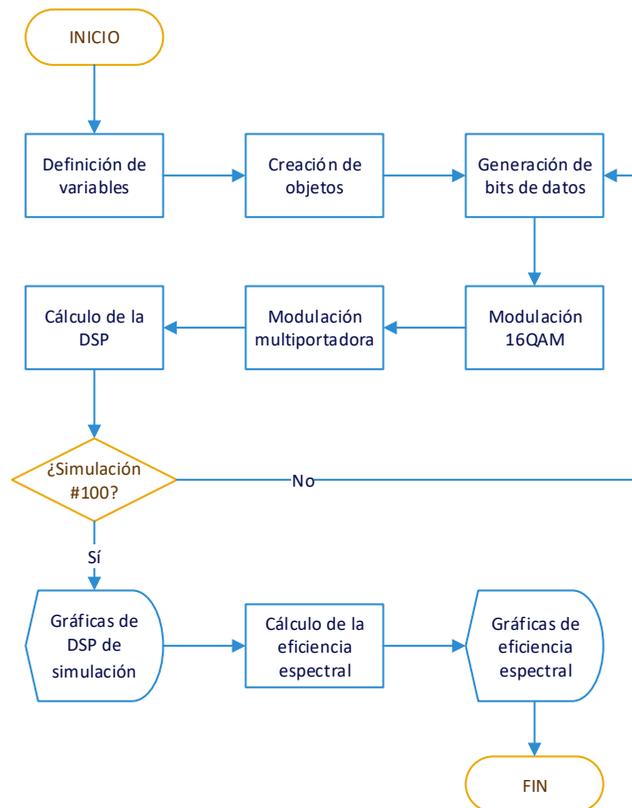


Figura 2.10 Diagrama de Flujo del script `Power_Spectral_Density.m`.

El primer paso a realizar es la definición de los parámetros para la implementación de las diferentes modulaciones. Los parámetros a definir se detallan en la Tabla 2.19.

Tabla 2.19 Parámetros de la simulación.

Parámetro	Descripción
NoSubP_Datos	Número de subportadoras utilizadas para transportar datos.
NoSubP_x_BG	Número de subportadoras de cada banda de guarda.
NoSubP_Total	Número total de subportadoras.
NoSimbolos	Número de símbolos UFMC a generarse.
No_Repeticion	Número de simulaciones.
M_QAM	Orden de la modulación QAM.
delta_f	Espaciamiento entre subportadoras.
Fs	Frecuencia de muestreo.
dt	Tiempo entre muestras.
FI	Permite realizar un desplazamiento en frecuencia, solo para propósitos de visualización.
T_IG	Duración de los intervalos de guarda en segundos.
FBMC_Factor_Solapamiento	Valores del factor de solapamiento.
FBMC_PPN	Valor lógico para determinar si se utiliza Red polifase o IFFT extendida.
OFDM_CP	Duración del prefijo cíclico en segundos.
UFMC_LongFiltro	Duración del filtro en segundos.

Después, se procede a la creación de los objetos de las diferentes clases. Para esto se llama a las clases con los parámetros de la Tabla 2.19 como parámetros de entrada. En el caso de FBMC, se crea un objeto por cada factor de solapamiento. En el código 2.35 se observa la creación de un objeto de la clase FBMC.

```

% FBMC con K = 3
FBMC3 = FBMC(...
    NoSubP_Datos, ... % Número de subportadoras de datos
    NoSubP_x_BG, ... % Número de subportadoras por banda de guarda
    NoSimbolos, ... % Número de símbolos FBMC
    delta_f, ... % Espaciamiento entre subportadoras (Hz)
    Fs, ... % Frecuencia de muestreo (Hz)
    FI, ... % Desplazamiento en frecuencia (para propósitos de
visualización)
    FBMC_Factor_Solapamiento(2), ... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN... % Implementación de una red polifase (PPN)
);
    
```

Código 2.35. Objeto de la clase FBMC.

El siguiente paso es la generación de los bits de datos aleatorios para cada técnica de modulación. A continuación, se realiza la modulación 16-QAM de los bits de datos. Los símbolos 16-QAM tendrán una potencia media igual a la unidad. En el código 2.36 se observa un ejemplo para la modulación FBMC.

```
Data_FBMC = randi([0 1],NoSubP_Datos/2*NoSimbolos*log2(M_QAM),1);
x_FBMC =
qammod(Data_FBMC,M_QAM,'InputType','bit','UnitAveragePower',true);
```

Código 2.36. Ejemplo para la modulación FBMC.

Posteriormente, se realiza la modulación multiportadora de los símbolos 16-QAM. Este proceso se realiza mediante el método `Modulacion` que poseen los objetos. Luego, se calcula la DSP de la señal multiportadora obtenida y se añade a la DSP acumulada. En el código 2.37 se observa un ejemplo para la modulación OFDM.

```
s_OFDM = OFDM.Modulacion(x_OFDM);
PSD_OFDM_Simulacion =
PSD_OFDM_Simulacion+(abs(fft(s_OFDM)).^2)/OFDM.No.TotalMuestras;
```

Código 2.37. Ejemplo para la modulación OFDM.

La generación de bits, modulación 16-QAM, modulación multiportadora y cálculo de DSP se repite el número de veces especificado en el parámetro `No_Repeticion`. Una vez realizadas todas las simulaciones, se obtiene un promedio y se normaliza a 0 dB para, finalmente, graficar la DSP de cada modulación. En el código 2.38 se observa un ejemplo para la modulación UFMC.

```
PSD_UFMC_Simulacion = PSD_UFMC_Simulacion/No_Repeticion;
PSD_UFMC_Simulacion = PSD_UFMC_Simulacion/max(PSD_UFMC_Simulacion);
plot(f_UFMC/delta_f-FI-
ceil(NoSubP_Total/2),10*log10(PSD_UFMC_Simulacion),'Color',ColorUFMC);
```

Código 2.38. Ejemplo para la modulación UFMC.

A continuación, se calcula la eficiencia espectral. Para esto se empieza definiendo la cantidad de símbolos multiportadora por ráfaga. Después, se obtiene la duración en tiempo por ráfaga. Acto seguido, se obtiene la eficiencia espectral de cada modulación en función de la duración de ráfaga. Por último, se grafica la eficiencia espectral de cada modulación. En el código 2.39 se observa un ejemplo para la modulación FBMC.

```
t_FBMC4=FBMC4.Imp.FactorSolap/delta_f+0.5/delta_f*(Nsimb_FBMC-1);
ef_FBMC4=(NoSubP_Datos/2*log2(M_QAM)*Nsimb_FBMC./t_FBMC4)/BW;
plot(t_FBMC4/1e-3,ef_FBMC4,'color',ColorFBMC4);
```

Código 2.39. Ejemplo para la modulación FBMC

A continuación se describe el script correspondiente al BER vs SNR.

2.5 BER vs SNR

El script `Bit_Error_Rate.m` es utilizado para obtener de las gráficas de BER vs SNR de las diferentes técnicas de modulación. En la figura 2.13 se indica el proceso para la obtención de las gráficas.

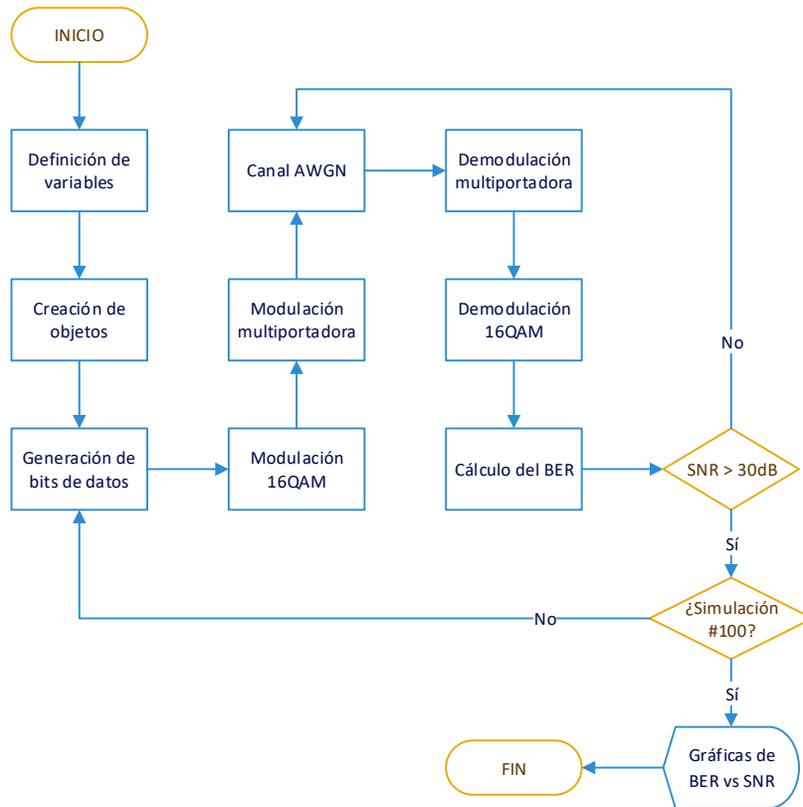


Figura 2.11 Diagrama de Flujo del script `Bit_Error_Rate.m`.

El primer paso a realizar es la definición de los parámetros para la implementación de las diferentes modulaciones. Los parámetros a definir son los mismos que se detallan en la Tabla 2.17. Además, se define el parámetro `SNR_dB`, el cual contiene los valores de SNR para los cuales se calcula el BER.

Después, se procede a la creación de los objetos de las diferentes clases. Para esto se llama a las clases con los parámetros de la Tabla 2.17 como parámetros de entrada. En el caso de FBMC, se crea un objeto por cada factor de solapamiento. En el código 2.35 se observa la creación de un objeto de la clase `FBMC`.

El siguiente paso es la generación de los bits de datos aleatorios para cada técnica de modulación. A continuación, se realiza la modulación 16-QAM de los bits de datos. Los

símbolos 16-QAM tendrán una potencia media igual a la unidad. En el código 2.36 se observa un ejemplo para la modulación FBMC.

Posteriormente, se realiza la modulación multiportadora de los símbolos 16-QAM. Este proceso se realiza mediante el método `Modulacion` que poseen los objetos. En el código 2.39 se observa un ejemplo para la modulación OFDM.

```
TX_OFDM = OFDM.Modulacion(x_OFDM);
```

Código 2.39. Ejemplo para la modulación OFDM.

A continuación, se procede a calcular el BER para cada valor de SNR almacenado en el parámetro `SNR_dB`. Para esto, a la señal multiportadora se le añade ruido AWGN con un valor de SNR. Después, se realiza la demodulación multiportadora mediante el método `Demodulacion` que posee cada objeto. Acto seguido, se realiza la demodulación 16-QAM para obtener los bits recibidos. Finalmente, se calcula el BER para el valor de SNR utilizado y se almacena el valor obtenido en una matriz. En el código 2.40 se observa un ejemplo para la modulación UPMC.

```
RX_UPMC = awgn(TX_UPMC,snrdB,'measured');  
y_UPMC = UPMC.Demodulacion(RX_UPMC);  
DataRX_UPMC =  
qamdemod(y_UPMC,M_QAM,'UnitAveragePower',true,'OutputType','bit');  
[~,BER_UPMC(i_SNR,i_rep)] = biterr(Data_UPMC,DataRX_UPMC);
```

Código 3.40. Ejemplo para la modulación UPMC.

Los pasos desde la generación de bits hasta el cálculo del BER se repite el número de veces especificado en el parámetro `No_Repeticion`. Una vez realizadas todas las simulaciones, se obtiene un promedio para, finalmente, graficar el BER vs SNR de cada modulación. En el código 2.41 se observa un ejemplo para la modulación FBMC.

```
semilogy(SNR_dB,mean(BER_FBMC2,2),'+-','color',ColorFBMC2);
```

Código 2.41. Ejemplo para la modulación UPMC.

3 RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados que se obtuvieron al realizar las simulaciones de las modulaciones OFDM, FBMC y UFMC. Es importante mencionar que, en un sistema de comunicación digital una de las características que se utiliza para reconocer si un sistema de comunicación es eficiente o no, es la tasa de error BER, por esta razón, se presentarán las gráficas de BER vs SNR, pues, permitirán realizar un análisis de los resultados que se obtuvo con el fin de comparar las mejoras que presenta FBMC y UFMC respecto a OFDM.

De la misma manera, para observar y analizar la densidad espectral de potencia para cada una de las modulaciones, se obtuvo las gráficas de la PSD de las diferentes modulaciones, con el fin de comparar las mejoras que existe con respecto a OFDM.

Es importante presentar las consideraciones que se tomaron en cuenta para realizar las simulaciones, a continuación se lista las siguientes:

- ❖ Se utilizó un esquema en banda base.
- ❖ Se consideró que el emisor y el receptor están perfectamente sincronizados.
- ❖ No se consideró interferencia entre canales.
- ❖ Se utilizó un canal ideal AWGN.
- ❖ No se consideró la existencia de interferencias externas.

Presentadas las consideraciones que se tuvieron al implementar los sistemas de modulación multiportadora, a continuación se procede a describir la configuración de los parámetros para las simulaciones de densidad espectral de potencia y BER vs SNR, respectivamente.

3.1 Configuración de parámetros

Para el análisis de las gráficas de densidad espectral de potencia PSD, se obtuvo el espectro para: 1, 10, 200 símbolos de cada una de las técnicas de modulación implementadas, OFDM, FBMC y UFMC. Para obtener estas gráficas se ha empleado un sistema sobremuestreado con el fin que las emisiones fuera de banda sean más visibles para su respectiva comparación. En la simulación se configuró parámetros para cada sistema, estos parámetros se han ido modificando con el fin de encontrar una mejor contención espectral. Cabe señalar que los parámetros que se configuraron para OFDM, son valores que corresponden a parámetros LTE, razón por la cual, al configurar los parámetros en FBMC y UFMC se escogieron valores similares, para realizar la respectiva comparación.

A continuación se presenta los parámetros para cada sistema de modulación.

3.1.1 Parámetros configurables en OFDM

Para OFDM los parámetros configurables fueron los siguientes:

Tabla 3.1 Parámetros para la simulación OFDM.

Parámetro	OFDM
Número de subportadoras de datos	64, 72, 128, 256 subportadoras.
Bandas de guarda	11x2 subportadoras.
Duración del prefijo cíclico	(1/14)Ts.
Modulación	16-QAM.
Número de símbolos	1, 10 y 200 símbolos multiportadora.

3.1.2 Parámetros configurables en FBMC

Para FBMC los parámetros configurables fueron los siguientes:

Tabla 3.2 Parámetros para la simulación FBMC.

Parámetro	FBMC
Número de subportadoras de datos	64, 72, 128, 256 subportadoras.
Bandas de guarda	11x2 subportadoras.
Factor de solapamiento K	2, 3, 4.
Modulación	16-QAM, OQAM.
Número de símbolos	1, 10 y 200 símbolos multiportadora.

3.1.3 Parámetros configurables en UFMC

Para UFMC los parámetros configurables fueron los siguientes:

Tabla 3.3 Parámetros para la simulación UFMC.

Parámetro	UFMC
Número de subportadoras de datos	64, 72, 128, 256 subportadoras.
Bandas de guarda	11x2 subportadoras.
Factor de subbandas	Depende del número de subportadoras.
Modulación	16-QAM.
Atenuación de filtro α	60 [dB].
Número de símbolos	1, 10 y 200 símbolos multiportadora.

Una vez configurados los parámetros para cada modulación multiportadora, se procede a describir los escenarios en los que se realizaron las simulaciones. Para cada escenario se presentaran los resultados que se obtuvieron en tablas, estos resultados permitirán comparar cauntitativamente la PSD de las modulaciones OFDM, FBMC y UFMC.

3.2 Escenario de Pruebas para PSD

En la simulación, cada sistema transmite bits aleatorios sobre el canal que se le agrega ruido blanco gaussiano. Para el análisis de las gráficas de Densidad Espectral de Potencia PSD, en el sistema FBMC, se comparará las densidades espectrales para los diferentes valores de ($K = 2,3,4$). Como el objetivo es comparar entre los tres sistemas de modulación, las gráficas de la PSD para cada sistema se presenta en la misma figura, con el fin de observar la diferencia que existe entre ellas.

Cabe señalar que los resultados se irán presentando conforme va variando el número de subportadoras (64, 72, 128, 256). Asimismo, en cada caso, se tendrá 3 escenarios, puesto que el número de símbolos va cambiando (1, 10, 200). A continuación se presentan los resultados y las gráficas que se obtuvieron al simular el script correspondiente a `Power_Spectral_Density`.

3.2.1 PSD para 64 subportadoras

En el primer escenario, para determinar la PSD se ha hecho las pruebas para 64 subportadoras. Dentro de este escenario, se observa cómo varía la PSD mientras el número de símbolos multiportadora aumenta, razón por la cual, se ha determinado 3 variantes (1,10, 200 símbolos). El sistema UFMC, en este escenario forma un total de 8 subbandas con 8 subportadoras cada una, cabe recalcar, que se está formando la subbanda con un mismo número de subportadoras respectivamente. Con los parámetros establecidos para este escenario, finalmente después de cada simulación, se recogieron los datos en la Tabla 3.4, estos valores corresponden a los cortes de la gráfica PSD en el eje Y y están expresados en [dB].

Tabla 3.4 PSD en [dB] para 64 subportadoras.

	OFDM [dB]	FBMC [dB]			UFMC [dB]
		K=2	K=3	K=4	
1 SÍMBOLO	-40	-322	-322	-322	-100
10 SIMBOLOS	-40	-50	-143	-160	-100
200 SÍMBOLOS	-40	-48	-142	-160	-100

La PSD simulada de OFDM, FBMC y UPMC, para 64 subportadoras, 22 bandas de guarda y 1 símbolo multiportadora se presenta en la Figura 3.1.

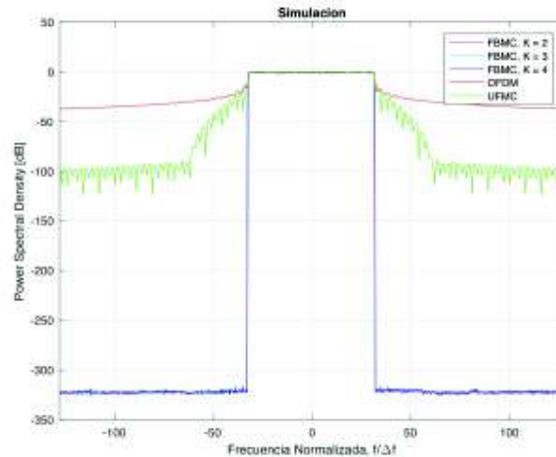


Figura 3.1 PSD para 1 símbolo multiportadora.

Al observar las gráficas de la PSD en la Figura 3.1, para el sistema FBMC se tiene que la emisión fuera de banda disminuye hasta un valor aproximado de -322 dB. Para esta modulación se distingue que, cuando FBMC envía pequeñas cantidades de datos, las emisiones fuera de banda disminuyen, pero, no distinguen el factor de solapamiento, es por eso que su uso para estos escenarios no es recomendable, ya que, el filtrado por subportadora implica un nivel de complejidad mayor que UPMC y OFDM, por lo tanto no es recomendable. Asimismo, la Figura 3.1 muestra que UPMC presenta un valor aproximado de -100 dB, mientras que OFDM en la misma frecuencia presenta un valor aproximado de -40 dB. Entonces, se comprueba que tanto UPMC, y FBMC presentan mejoras respecto a OFDM.

En la Figura 3.2 se presenta la gráfica de la PSD al aumentar el número de símbolos multiportadora a 10, la PSD simulada de OFDM, FBMC y UPMC, para 64 subportadoras, 22 bandas de guarda.

Para el sistema FBMC se tiene que las emisiones fuera de banda difieren conforme el factor de solapamiento ($K = 2,3,4$), esto es, que dependiendo el valor de K disminuye hasta un valor aproximado de -322 dB. Es importante recalcar que, para $K = 2$ disminuye hasta un valor de -50 dB, como se observa, este valor se aproxima mucho al valor que presenta OFDM en el mismo escenario, esto afirma que para K menor, los resultados que se obtienen son similares a los de OFDM, sin embargo, presenta mejoras. Por otro lado, UPMC se mantiene en el valor aproximado de -100 dB, mientras que OFDM en la misma frecuencia permanece aproximadamente -40 dB.

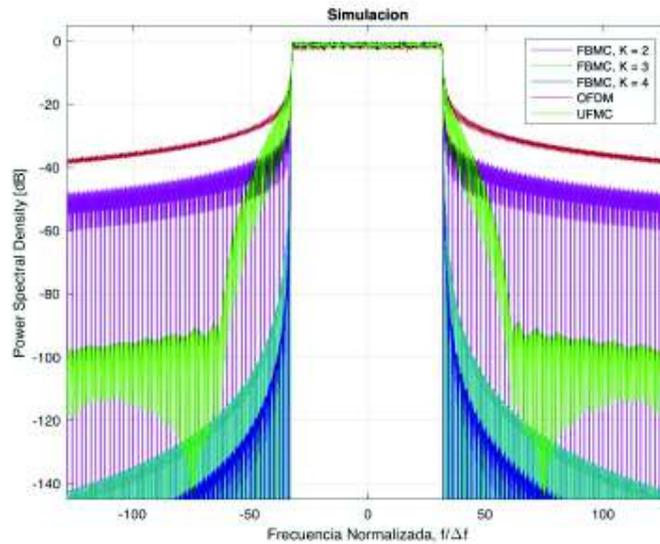


Figura 3.2 PSD para 10 símbolos multiportadora.

Finalmente, se hace la prueba para un total de 200 símbolos multiportadora, la PSD simulada de OFDM, FBMC y UPMC, para 64 subportadoras, 22 bandas de guarda se ilustra en la Figura 3.3.

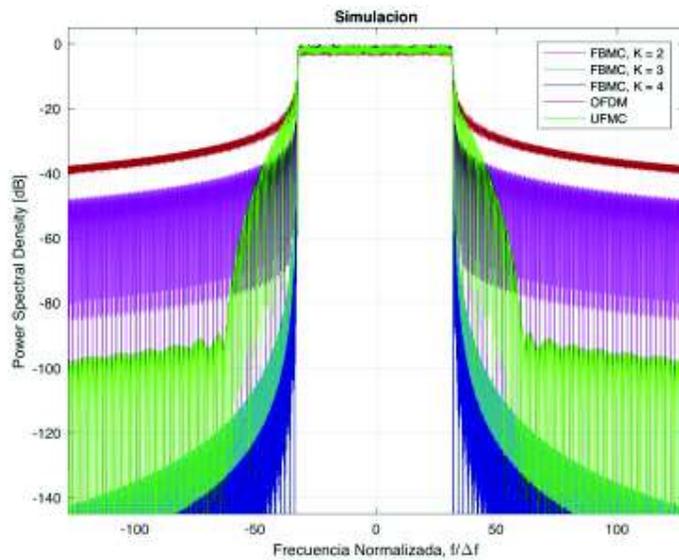


Figura 3.3 PSD para 200 símbolos multiportadora.

En la Figura 3.3, se puede apreciar en una misma gráfica las densidades espectrales de potencia (PSD) de las tres modulaciones OFDM, FBMC ($K = 2, 3, 4$) y UPMC.

En la modulación FBMC para $K = 2$, $K = 3$, $K = 4$, donde K indica el número de símbolos que se solapan. Se puede observar como la modulación FBMC para $K = 4$ presenta emisiones fuera de banda mucho menores que las modulaciones OFDM y UPMC. Esto, demuestra que a mayor valor de K , más selectivo es el banco de filtros, por lo tanto mayor contención de espectro existe. Al observar las gráficas de la PSD en FBMC la emisión fuera de banda disminuye hasta un valor de -160 dB aproximadamente. El sistema UPMC se mantiene en un valor aproximado de -100 dB. Hay que destacar que a mayor número de símbolos en UPMC en la gráfica de la PSD se nota el filtrado por subbandas, ya que se puede apreciar en la parte superior de la gráfica, se han formado 8 picos. Mientras que OFDM en la misma frecuencia disminuye hasta un valor de -40 dB.

A continuación se presentan los resultados que se han obtenido para un total de 72 subportadoras.

3.2.2 PSD para 72 subportadoras

En el segundo escenario, para determinar la PSD se ha hecho las pruebas para 72 subportadoras. Dentro de este escenario, se va a ir observando cómo varía la PSD mientras el número de símbolos multiprotadora aumenta, asimismo, se va a simular para 1, 10, 200 símbolos. UPMC, para este escenario formará un total de 6 sub bandas con 12 subportadoras cada una. Es importante notar que para este número de sub portadoras FBMC puede trabajar con parámetros LTE, es decir, agrupar en cada subbanda 12 subportadoras. Al final de cada simulación, se recogieron los datos en la Tabla 3.5, estos valores corresponden a los cortes de la gráfica PSD en el eje Y y están expresados en [dB].

Tabla 3.5 PSD en [dB] para 72 subportadoras.

	OFDM [dB]	FBMC [dB]			UPMC [dB]
		K=2	K=3	K=4	
1 SÍMBOLO	-40	-320	-320	-320	-100
10 SIMBOLOS	-40	-50	-145	-160	-100
200 SÍMBOLOS	-40	-50	-145	-160	-100

De la misma manera, para OFDM, FBMC ($K = 2,3,4$) y UPMC, usando 72 subportadoras de datos, 22 bandas de guarda, 1 símbolo multiprotadora, se obtuvo la densidad espectral de potencia que se ilustra en la Figura 3.4.

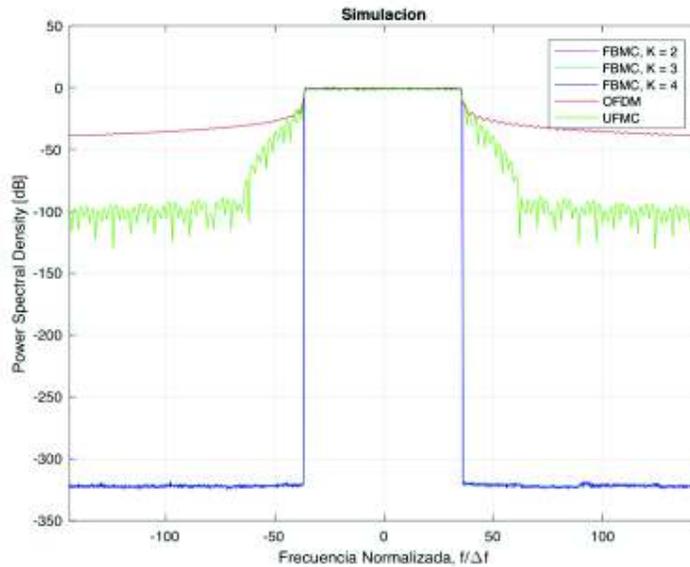


Figura 3.4 PSD para 1 símbolo multiportadora.

Como se observa en la Figura 3.4, cuando se trabaja con un número de subportadoras bajo y se genera 1 símbolo multiportadora, comparando la contención espectral de las 3 modulaciones, la que menos contención espectral presenta es la modulación OFDM puesto que presenta aproximadamente -40 dB de atenuación. Por otro lado, para el caso de FBMC, para los 3 factores de solapamiento ($K=1$, $K=2$ y $K=3$) presenta aproximadamente -320 dB de atenuación, es decir, para cantidades pequeñas de datos realizar una modulación con FBMC no es la mejor opción. Mientras tanto, para UFMC, el caso mejora puesto que comparado con OFDM presenta aproximadamente una atenuación de -100 dB. Por otro lado, para OFDM, FBMC ($K = 2,3,4$) y UFMC, usando 72 subportadoras de datos, 22 bandas de guarda, 10 símbolos multiportadora, se obtuvo la densidad espectral de potencia simulada que se muestra en la Figura 3.5.

Se distingue que el sistema que mayor contención espectral presenta es FBMC con factor de solapamiento $K = 4$, ya que disminuye hasta un valor aproximado de -160 dB. Ahora, comparando la contención espectral de las 3 modulaciones, la que menos contención espectral presenta es la modulación OFDM puesto que refleja un valor aproximado de -40 dB de atenuación. Por otro lado, para el sistema UFMC, el caso mejora puesto que comparado con OFDM presenta aproximadamente una atenuación de -100 dB.

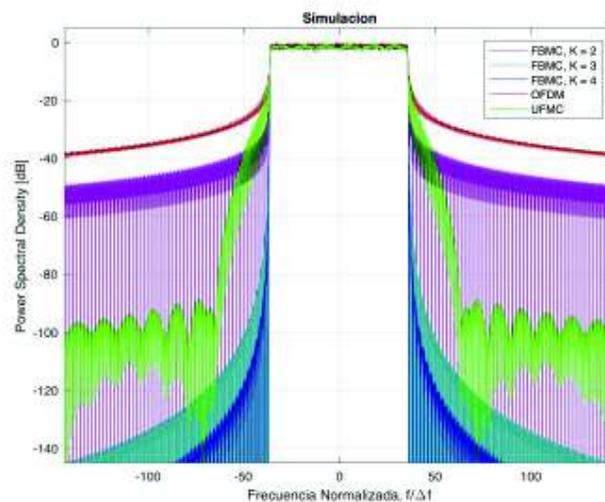


Figura 3.5 PSD para 10 símbolos multiportadora.

A continuación se presenta en la Figura 3.6, las gráficas de la PSD, para OFDM, FBMC ($K = 2,3,4$) y UFMC, usando 72 subportadoras de datos, 22 bandas de guarda y 200 símbolos multiportadora.

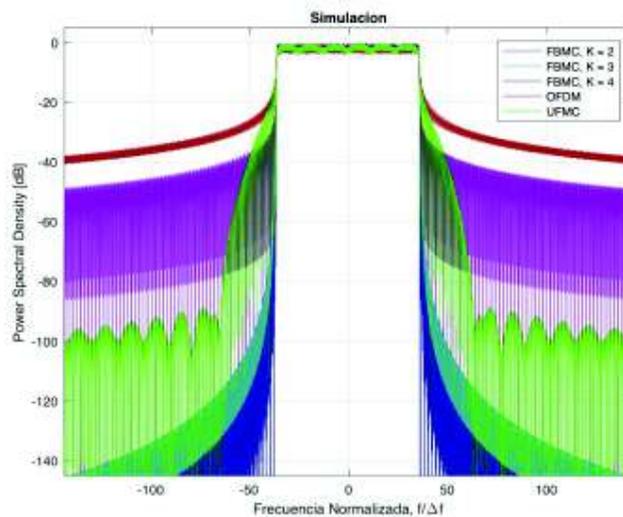


Figura 3.6 PSD para 200 símbolos multiportadora.

En la Figura 3.6 es fácil distinguir la diferencia entre las tres modulaciones. La diferencia es más visible en la PSD de UFMC (color verde), en esta, se observa el resultado de filtrar un grupo de subportadoras, ya que, para un total de 72 subportadoras, usando parámetros de LTE, se formaron 6 subbandas de 12 subportadoras cada una. Este efecto se observa en el máximo del espectro, puesto que presenta 6 lóbulos. Por lo tanto, la modulación UFMC también presenta una reducción de las emisiones fuera de banda ya que presenta

un valor aproximado de -100 dB, este valor es mejor que OFDM que presenta un valor aproximado de -40 dB. Se observa entonces que, al utilizar un filtrado por subbanda la contención espectral por subbanda es mucho mayor que la que presenta OFDM. Por otro lado, el sistema FBMC, sigue mostrando mejor contención espectral con factor de solapamiento 4, pues este presenta un valor de -160 dB aproximadamente.

3.2.3 PSD para 128 subportadoras

En el tercer escenario, para determinar la PSD se ha hecho las pruebas para 128 subportadoras. Dentro de este escenario, de igual manera que el los otros escenarios, se pretende observar la forma en que varía la PSD mientras el número de símbolos aumenta de 1, 10, 200 símbolos multiportadora. UFMC, para este escenario formará un total de 16 subbandas con 8 subportadoras cada una. Al final de cada simulación, se recogieron los datos en la Tabla 3.6, estos valores corresponden a los cortes de la gráfica PSD en el eje Y y están expresados en [dB].

Tabla 3.6 PSD en [dB] 128 subportadoras.

	OFDM [dB]	FBMC [dB]			UFMC [dB]
		K=2	K=3	K=4	
1 SÍMBOLO	-40	-320	-320	-320	-100
10 SIMBOLOS	-41	-50	-158	-175	-100
200 SÍMBOLOS	-41	-50	-158	-175	-100

Para OFDM, FBMC ($K = 2,3,4$) y UFMC, usando 128 subportadoras de datos, 22 bandas de guarda, 1 símbolo multiportadora, se obtuvo la densidad espectral de potencia que se ilustra en la Figura 3.7.

Como se observa en la Figura 3.7, al generar 1 solo símbolo multiportadora, comparando la contención espectral de las 3 modulaciones, la que menos contención espectral presenta es la modulación OFDM puesto que presenta aproximadamente -40 dB de atenuación. Asimismo, el sistema FBMC, para los 3 factores de solapamiento ($K=1$, $K=2$ y $K=3$) presenta aproximadamente -320 dB de atenuación, es decir, se mantiene en los valores que se obtuvo para un menor número de subportadoras. De igual forma, el sistema UFMC presenta aproximadamente una atenuación de -100 dB.

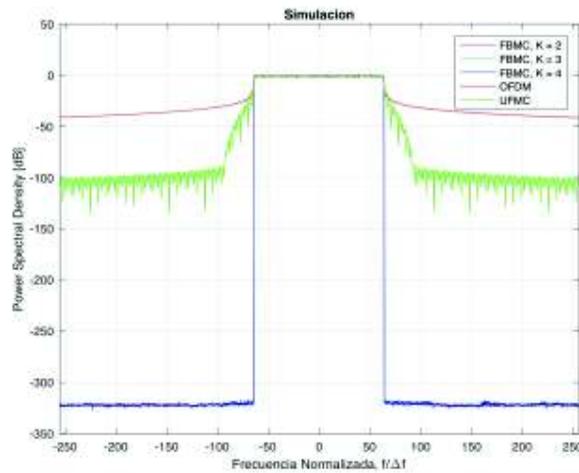


Figura 3.7 PSD para 1 símbolo multiportadora.

Ahora, para OFDM, FBMC ($K = 2,3,4$) y UPMC, usando 128 subportadoras de datos, 22 bandas de guarda, 10 símbolos multiportadora, se obtuvo la densidad espectral de potencia simulada que se muestra en la Figura 3.8.

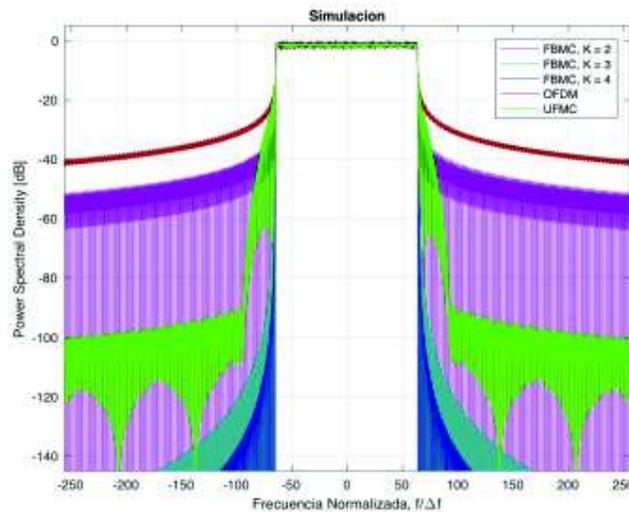


Figura 3.8 PSD para 10 símbolos multiportadora.

Al observar las gráficas que se ilustran en la figura 3.8, se distingue que el sistema que mayor contención espectral presenta es FBMC con factor de solapamiento $K = 4$, ya que disminuye hasta un valor aproximado de -160 dB. Estos valores no varían para el caso de 72 subportadoras. De la misma manera que para el sistema OFDM puesto que refleja un valor aproximado de -40 dB de atenuación. Por otro lado para el sistema UPMC, se mantiene en un valor de atenuación aproximadamente de -100 dB.

Finalmente, en este escenario, simulando OFDM, FBMC ($K = 2,3,4$) y UFMC, para 128 subportadoras de datos, 22 bandas de guarda, 200 símbolos multiportadora, se obtuvo la densidad espectral de potencia simulada que se muestra en la Figura 3.9.

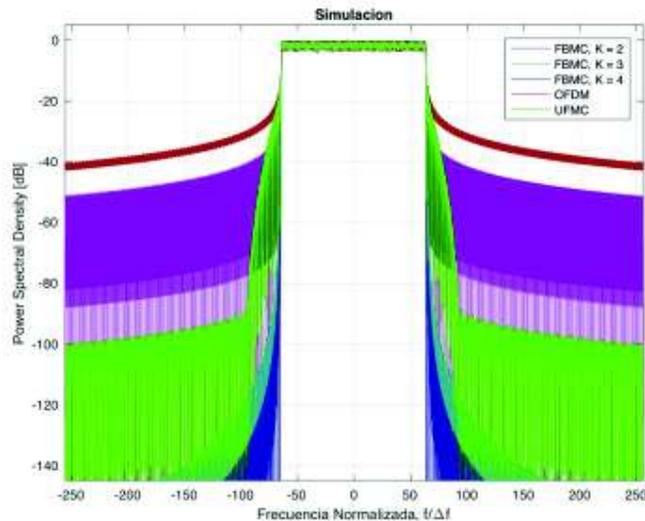


Figura 3.9 PSD para 200 símbolos multiportadora.

En la Figura 3.9 se observa que cuando el número de subportadoras aumenta, y para un número alto de símbolos se genera un espectro en el que se puede apreciar de mejor forma la contención del espectro, ya que es más fácil distinguir la diferencia entre las tres modulaciones. En este caso, para el sistema UFMC se crean las subbandas y se puede distinguir en las caídas que presenta el espectro, este llega a un valor aproximado de -100 dB, este valor es mejor que OFDM que presenta un valor aproximado de -41 dB. Por otro lado, el sistema FBMC, sigue mostrando mejor contención espectral con factor de solapamiento 4, pues este presenta un valor de -175 dB aproximadamente.

3.2.4 PSD para 256 Subportadoras

En el cuarto escenario, para determinar la PSD se ha hecho las pruebas para 256 subportadoras. Dentro de este escenario, se observa cómo varía la PSD mientras el número de símbolos multiportadora aumenta, 1,10, 200 símbolos. UFMC, para este escenario formará un total de 32 subbandas con 8 subportadoras cada una. Al final de cada simulación, se recogen los datos en la Tabla 3.7, estos valores corresponden a los cortes de la gráfica PSD en el eje Y y están expresados en [dB].

Tabla 3.7 PSD en [dB] 256 subportadoras.

	OFDM	FBMC			UFMC
		K=2	K=3	K=4	
1 SÍMBOLO	-42	-320	-320	-320	-102
10 SIMBOLOS	-42	-52	-173	-190	-102
200 SÍMBOLOS	-43	-52	-173	-190	-102

Para OFDM, FBMC ($K = 2,3,4$) y UFMC, usando 256 subportadoras de datos, 22 bandas de guarda, 1 símbolo multiportadora, se obtuvo la densidad espectral de potencia simulada que se muestra en la Figura 3.10.

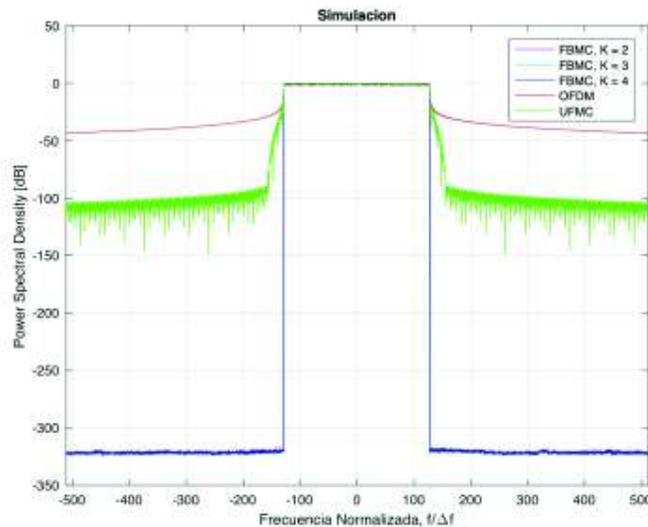


Figura 3.10 PSD para 1 símbolo multiportadora.

Además, se observa en la Figura 3.10, la diferencia que existe si aumenta el número de subportadora, pues aunque los valores en FBMC se conservan igual que a los otros casos, en donde se trabaja con 1 símbolo multiportadora, el sistema UFMC alcanza un valor aproximado de -102 dB. Asimismo el sistema OFDM mejora al alcanzar un valor de -42 dB pero sigue presentando una menor contención espectral.

Ahora, para OFDM, FBMC ($K = 2,3,4$) y UFMC, usando 256 subportadoras de datos, 22 bandas de guarda, 10 símbolos multiportadora, se obtuvo la densidad espectral de potencia simulada que se muestra en la Figura 3.11.

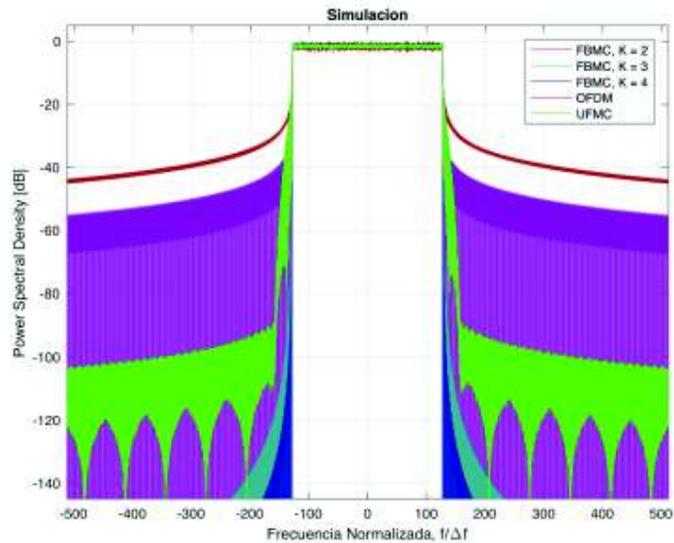


Figura 3.11 PSD para 10 símbolos multiportadora.

En la Figura 3.11, se observa el efecto de trabajar con un número alto de subportadoras y de símbolos multiportadora, pues presenta una mejor resolución del espectro, destacando FBMC y UFMC que presentan mayor contención espectral respecto a OFDM. Alcanzando valores aproximados de -190 dB y -102 dB, respectivamente.

Asimismo, para OFDM, FBMC ($K = 2,3,4$) y UFMC, usando 256 subportadoras de datos, 22 bandas de guarda, 200 símbolos multiportadora, se obtuvo la densidad espectral de potencia simulada que se muestra en la Figura 3.12.

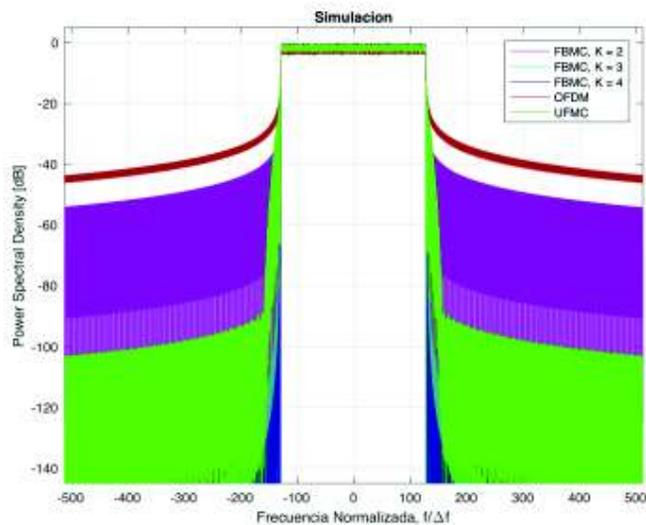


Figura 3.12 PSD para 200 símbolos multiportadora.

Al observar las gráficas de la PSD en FBMC la emisión fuera de banda disminuye hasta un valor de -190 dB. OFDM en la misma frecuencia disminuye un valor de aproximado de -43 dB. UFMC se mantiene en un valor de -102 dB aproximadamente.

Terminado todos los escenarios de pruebas para obtener la gráficas de la PSD, se procede a realizar las simulaciones para obtener las curvas de BER vs SNR.

3.3 Eficiencia espectral

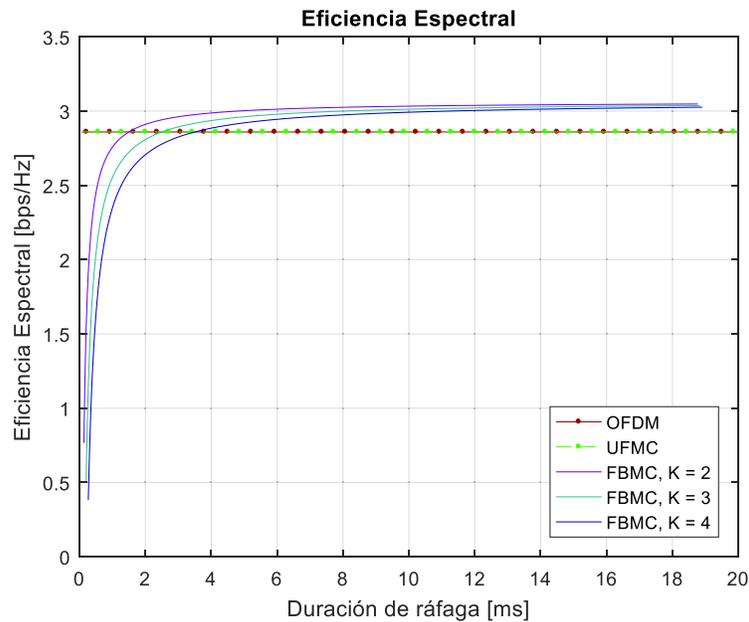


Figura 3.13 Eficiencia espectral de OFDM, FBMC y UFMC.

En la figura 3.13 se puede observar que la eficiencia espectral de UFMC es la misma que OFDM debido a que la longitud del filtro utilizado coincide con el tamaño de CP de OFDM. Para mejorar la eficiencia espectral de UFMC se debe disminuir la longitud del filtro utilizado.

También, se puede observar que la eficiencia espectral de OFDM y UFMC no depende de la duración de ráfaga como es el caso de FBMC.

En el caso de FBMC, se puede observar que su eficiencia espectral es dependiente de la duración de ráfaga. Para ráfagas de muy pequeña duración, se observa que la eficiencia espectral es peor en comparación con OFDM y UFMC. Por el contrario, se observa que para ráfagas de gran duración la eficiencia espectral es mejor que OFDM y UFMC.

3.4 Escenarios de pruebas para BER vs SNR

En esta sección se analizarán las diferentes curvas de BER vs SNR que se obtuvieron durante la simulación. Los parámetros que se configuraron fueron los mismos que se usaron para la simulaciones de la PSD, en este caso solo se trabajó con 200 símbolos multiportadora.

Para el análisis, se considera que la señal fue transmitida a través de un canal AWGN utilizando diferentes valores de SNR. El rango de SNR utilizado fue de 5 dB a 30 dB, con un intervalo de 0.5 dB. Para obtener las curvas de BER se generaron 200 símbolos multiportadora de cada modulación OFDM, FBMC y UPMC. Estos símbolos multiportadora pasaron por el canal y se calculó el BER para los distintos valores de SNR. Este proceso se repitió un total de 100 veces, y finalmente se obtuvo un valor promedio de BER.

Los resultados de la simulación se obtuvieron a partir de los parámetros de simulación descritos en las Tablas 3.1, 3.2 y 3.3. El resultado que se obtuvo para un total de 72 subportadoras es la gráfica que se ilustra en la figura 3.14.

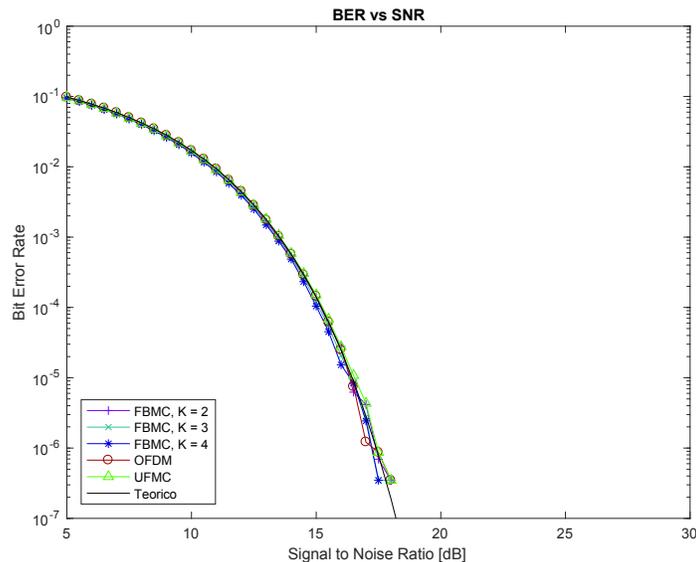


Figura 3.14. BER vs SNR para las modulaciones OFDM, UPMC Y FBMC para 72 subportadoras.

Como se puede observar, los valores de BER que se obtuvieron para cada modulación son similares entre sí. Para poder apreciar las diferencias entre las curvas, se realiza un zoom de la gráfica, como se muestra en la figura 3.15.

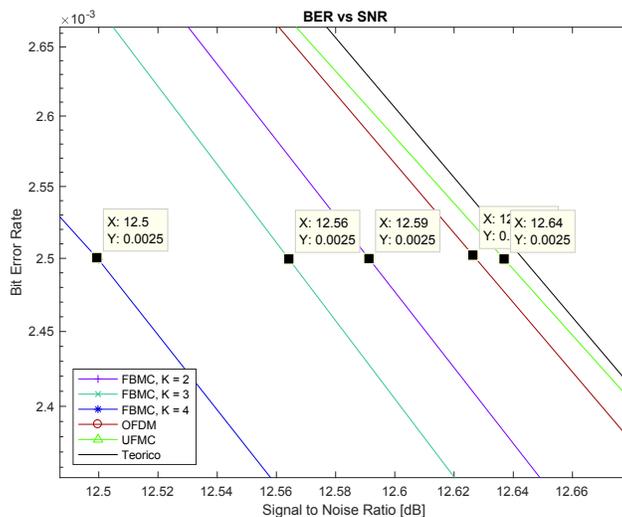


Figura 3.15. Zoom de la gráfica BER vs SNR.

En la Figura 3.15 se observa que para un BER de 2.5×10^{-3} , FBMC y UFMC requieren menor valor de SNR que OFDM, así: FBMC con factor de solapamiento $k = 4$ requiere un valor de SNR de 12.5 [dB], FBMC con factor de solapamiento $k = 3$ requiere un valor de SNR de 12.56, FBMC con factor de solapamiento $k = 2$ requiere un valor de SNR de 12.59 [dB]. Asimismo, UFMC requiere un valor de 12.64 [dB], similar al de OFDM.

En el caso de FBMC, para un valor de $K = 4$ se obtiene un BER menor, puesto que presenta menor contención espectral. Para $K = 2$ la relación de BER crece, esto significa que para este valor de K el sistema presentaría más cantidad de errores, por lo que no sería recomendable su uso para el envío de mucha información. La solución sería enviar un menor número de bits cuando se trabaje con valores menores de SNR.

De la misma manera al comparar las curvas del BER de OFDM y UFMC, muestran que al enviar datos utilizando un transmisor con UFMC y otro con OFDM, los resultados son mejores en UFMC gracias a las características de los filtros *Dolph Chebyshev*, con valores mayores de SNR en recepción va mejorando la probabilidad de error.

Se presenta a continuación las gráficas de BER vs SNR para 64, 128, 256 y 1024 subportadoras.

3.4.1 BER vs SNR para 64 subportadoras

En la Figura 3.16 se muestra la gráfica de BER vs SNR, para un total de 64 subportadoras.

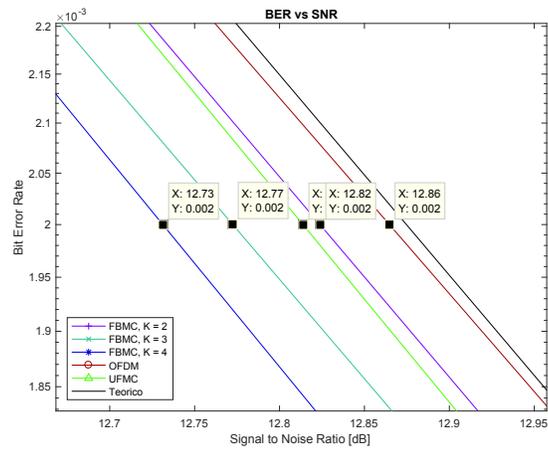


Figura 3.16. BER vs SNR para las modulaciones OFDM, UPMC Y FBMC para 64 subportadoras.

Para estos resultados que se observa en la figura 3.16 se ha empleado un total de 64 subportadoras, con 11 bandas de guarda a cada lado. Para OFDM se mantiene el prefijo cíclico igual a $(\frac{1}{14})T_s$. Las modulaciones que se emplearon para OFDM y UPMC fue de 16-QAM, mientras que para FBMC se empleó 16-QAM y OQAM. Para poder comparar se ha generado la misma cantidad de símbolos multiportadora.

3.4.2 BER vs SNR para 128 subportadoras

En la figura 3.17 se muestra la gráfica de BER vs SNR, para un total de 128 subportadoras.

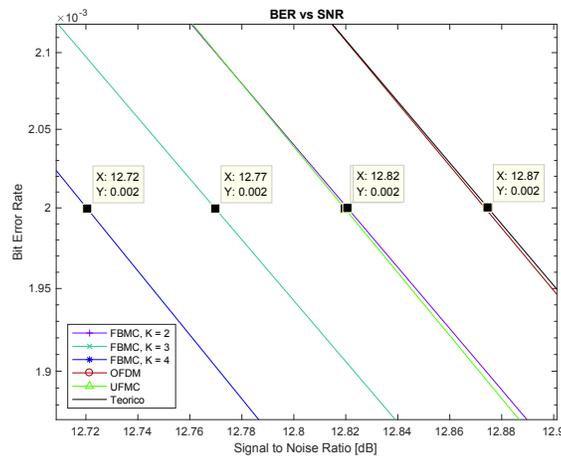


Figura 3.17. BER vs SNR para las modulaciones OFDM, UPMC Y FBMC para 128 subportadoras.

Al observar la figura 3.17 se puede apreciar una mejora en la curva de BER vs SNR, Las curvas de UFMC y FBMC con $K = 2$, toman valores similares. El sistema que mejor valor de BER obtiene es FBMC con $k = 4$.

3.4.3 BER vs SNR para 256 subportadoras

En la figura 3.18 se muestra la gráfica de BER vs SNR, para un total de 256 subportadoras.

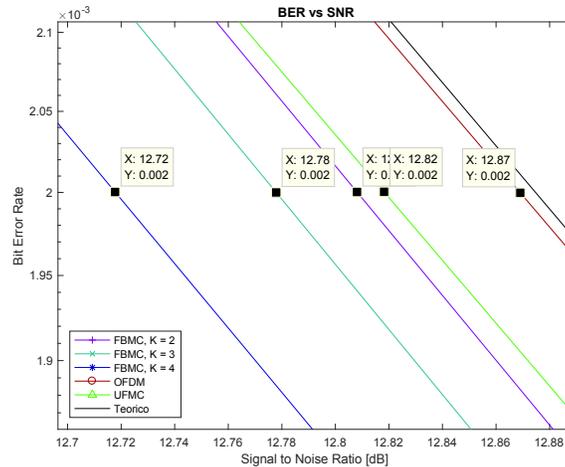


Figura 3.18. BER vs SNR para las modulaciones OFDM, UFMC Y FBMC para 256 subportadoras.

En la Figura 3.18 se observa que para este escenario, los valores de BER para UFMC y para FBMC con $K = 2$, se asemejan, lo que quiere decir que para un número de subportadoras alto, con niveles de SNR entre los 10 dB FBMC y UFMC presentan el mismo rendimiento. Asimismo, el sistema que mejor valor de BER obtiene es FBMC con $k = 4$.

3.4.4 BER vs SNR para 1024 subportadoras

En la figura 3.19 se muestra la gráfica de BER vs SNR, para un total de 128 subportadoras. Observando las Figuras de BER vs SNR, se aprecia las tasas de errores obtenidos son similares para diferentes niveles de ruido en el canal. Pero, en el caso de UFMC y FBMC, la cantidad de errores es menor, distinguiendo el caso del sistema de FBMC con $K = 4$, lo que representa una mejora en la tasa de errores en FBMC y UFMC respecto a OFDM.

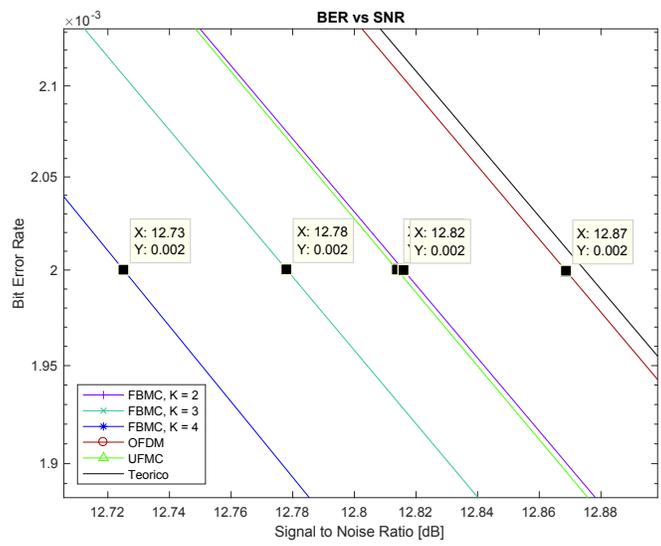


Figura 3.19. BER vs SNR para las modulaciones OFDM, UPMC Y FBMC para 1024 subportadoras.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- ❖ En este Trabajo se estudió y comparó las modulaciones multiportadora FBMC y UFMC, basadas en OFDM. Así, la comparación se ha realizado mediante simulaciones que han permitido confirmar las mejoras que presentan respecto a OFDM.
- ❖ El estudio se realizó para verificar que, tanto FBMC como UFMC mejoran la contención espectral respecto a OFDM, aunque las dos primeras requieren mas recursos de procesado, estas técnicas disminuye las emisiones fuera de banda.
- ❖ La ortogonalidad que requiere OFDM para todas las subportadoras, se reduce en FBMC a la ortogonalidad únicamente de subportadoras adyacentes sin la necesidad del prefijo cíclico gracias a la modulación OQAM, es decir, aunque dos subportadoras contiguas se solapen, al transmitir la parte real de un símbolo en una subportadora y la imaginaria en la siguiente, estas subportadoras no interfieren.
- ❖ En FBMC el diseño del filtro prototipo, es la clave principal para mejorar su rendimiento, pues, de esto depende que el sistema presente emisiones fuera de banda bajas. Este filtro es diseñado con el fin de obtener mayor contención espectral de potencia posible y depende del factor de solapamiento K , que, como se confirmó en los resultados de las simulaciones, para un valor de $K = 2$, las emisiones fuera de banda presentan un nivel de rechazo espectral similar a las de OFDM. Por el contrario, mientras el valor de K aumenta mayor contención espectral presenta.
- ❖ Un sistema FBMC con factor de solapamiento $K = 4$ es el que mayor contención espectral presenta, sin embargo, al realizar el filtrado en banda estrecha, provoca una respuesta impulsiva larga en el tiempo. Por lo tanto, debido a la longitud de sus filtros no es recomendable para la transmisión de ráfagas cortas de datos.
- ❖ En un sistema FBMC, a través del uso de un banco de filtros se separa las subportadoras, reduciendo la necesidad de recurrir a métodos que utiliza OFDM para evitar interferencia. De este modo, la contención espectral que se obtiene mediante esta técnica evita la distorsión de señales en bandas adyacentes.
- ❖ Como las técnicas FBMC y UFMC necesitan una banda de guarda inferior, permite trabajar en bandas más estrechas. Esto representa un ahorro, puesto que se optimiza el espectro electromagnético.

- ❖ Por otro lado, la técnica UFMC, gracias a las características de los filtros Dolph-chebyshev, presenta mejoras respecto a OFDM, ya que estos filtros son parametrizables tanto en su longitud como en la forma de atenuación de los lóbulos laterales. La elección correcta de los valores de estos parámetros reduce la interferencia entre subportadoras adyacentes, generando un espectro similar al de FBMC y de este modo una eficiencia espectral similar. Bajo esta perspectiva UFMC, también disminuye considerablemente las emisiones fuera de banda respecto a OFDM.
- ❖ En UFMC, al aplicar un filtrado a un grupo de subportadoras se protege la interferencia entre las mismas, de esta forma aumenta la eficiencia espectral respecto de OFDM ya que no es necesario utilizar el CP. Además, el filtrado por subbandas supone la reducción de la longitud de los filtros en comparación con FBMC. Por lo tanto, UFMC por su mejor respuesta impulsiva se adapta mejor para la transmisión de ráfagas cortas de datos.

4.2 RECOMENDACIONES

- ❖ Se recomienda utilizar un computador de características mínimas de 8 GB en RAM, un procesador 2,6 GHz, Core i5 de quinta generación, puesto que, al realizar simulaciones con valores de 1024 subportadoras y 200 símbolos requiere una carga computacional alta.
- ❖ Para las simulaciones se recomienda usar el software Matlab versión 16 o superior, debido a que las versiones anteriores no soporta los parámetros de entrada de las funciones que se utilizaron del toolbox *System Communication*.
- ❖ Se recomienda utilizar parámetros similares para los tres sistemas al simular con el fin de realizar una comparación objetiva.
- ❖ Si bien en este trabajo no se considera interferencias externas, se podría ampliar este trabajo con algoritmos para estimación del canal.
- ❖ Asimismo, se podría utilizar dispositivos RF conectados a un ordenador con la herramienta de Matlab en transmisión y recepción, con el fin de obtener medidas reales sobre el canal, y así realizar un estudio del rendimiento de estos sistemas.
- ❖ Se podría estudiar los efectos que tendría el movimiento físico del receptor, para ver la respuesta que tendrían los sistemas ante el efecto Doppler.
- ❖ Para disminuir la degradación de la tasa de bits errados, se puede emplear códigos de corrección de errores en cada sistema.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] P. Shelswell, "The COFDM Modulation System," *The Heart of Digital Audio Broadcasting, BBC Research and Development Report, BBC RD*, vol. 8, 1996.
- [2] R. Ziemer and W. H. Tranter, *Principles of communications: system modulation and noise*. John Wiley & Sons, 2006.
- [3] A. G. Armada, M. Calvo, and L. de Haro, "Diseño del subsistema de sincronismo para un sistema de comunicaciones basado en OFDM," *Dpto. SSR. ETSI Telecomunicación, UPM Ciudad Universitaria-28040 MADRID*, 1998.
- [4] D. Zurdo Martín, "Estudio de prestaciones de la transmisión OFDM para su utilización en entrenamiento supervisado," 2010.
- [5] C. J. C. González, *Fundamentos del análisis de Fourier*. Departamento de Enxeñaría Eléctrica, Escola Técnica Superior de Enxeñeiros Industriais, Universidade de Vigo, 2003.
- [6] O. Edfors, M. Sandell, J. v. d. Beek, D. Landström, and F. Sjöberg, "An introduction to orthogonal frequency-division multiplexing," ed: Luleå tekniska universitet, 1997.
- [7] W. Y. Zou and Y. Wu, "COFDM: An overview," *IEEE transactions on broadcasting*, vol. 41, no. 1, pp. 1-8, 1995.
- [8] L. JIMENEZ CANO, "Modulación multiportadora OFDM," *Revistas Udistrital*, vol. 6, pp. 30-34, nov 2000.
- [9] F. Schaich and T. Wild, "Waveform contenders for 5G—OFDM vs. FBMC vs. UFMFC," in *Communications, Control and Signal Processing (ISCCSP), 2014 6th International Symposium on*, 2014, pp. 457-460: IEEE.
- [10] D. Matiae, "OFDM as a possible modulation technique for multimedia applications in the range of mm waves," *Introduction to OFDM*, vol. 1, pp. 10-30, 1998.
- [11] L. Hazy, "Initial channel estimation and frame synchronization of OFDM systems for frequency-selective channels," 1997.
- [12] F. Schaich, T. Wild, and Y. Chen, "Waveform contenders for 5G-suitability for short packet and low latency transmissions," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, 2014, pp. 1-5: IEEE.
- [13] C. A. Ramírez Behaine, "Modulación por multiportadoras tipo OFDM, las bases de la nueva generación de transmisión de información," *Revista Ingenierías Universidad de Medellín*, vol. 5, pp. 75-83, 2006.

- [14] C. O. D. I. DE TELECOMUNICACIÓN, "Modulación y multiacceso en 5G," 2018.
- [15] C. Larrazábal and P. Ramón, "Determinación de los requerimientos en cuanto a método de acceso y cobertura para la implementación de sistemas 5G/Determination of the access method and coverage requirements for the implementation of 5G systems," *Revista Estudiantil URU*, no. 2, pp. 41-49, 2016.
- [16] E. O. Martínez and L. B. Agudelo, "FBMC, una opción para un uso óptimo del espectro electromagnético," *Revista en telecomunicaciones e informática*, vol. 3, no. 5, pp. 39-53, 2013.
- [17] M. Bellanger *et al.*, "FBMC physical layer: a primer," *PHYDYAS, January*, vol. 25, no. 4, pp. 7-10, 2010 [En línea]
Available: http://www.ict-phydyas.org/teamspace/internal-folder/FBMC-Primer_06-2010.pdf [Último acceso: 25 de junio de 2018].
- [18] O. Martínez and E. Augusto, "Propuesta para el uso de FBMC en el enlace de subida de un sistema de comunicación inalámbrico multipunto a punto," 2015 [En línea]
Available: <https://repository.upb.edu.co/handle/20.500.11912/3013>
[Último acceso: 12 de julio de 2018].
- [19] S. Lu, D. Qu, and Y. He, "Sliding window tone reservation technique for the peak-to-average power ratio reduction of FBMC-OQAM signals," *IEEE Wireless Communications Letters*, vol. 1, no. 4, pp. 268-271, 2012.
- [20] D. G. MARTÍN, "GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN," 2017 [En línea]
Available: <https://core.ac.uk/download/pdf/94522102.pdf> [Último acceso: 05 de agosto de 2018].
- [21] H. V. G. Navarro and C. A. R. López, "ESTUDIO DE NUEVAS TECNOLOGIAS DE ACCESO DE RADIO PARA REDES MOVILES FUTURAS (5G)."
- [22] D. Naranjo, D. Buenaño, and I. T. Mejía, "Evolución de la tecnología móvil. Camino a 5G," *Revista Contribuciones a las Ciencias Sociales*, 2016.
- [23] E. G. Gutiérrez, "Introducción al filtrado digital," *Catalunya: Escola Superior de Musica de Catalunya, Departamento de Sonologia*, 2009.
- [24] H. Moliné Borrueal, "Tecnología 5G y formas de onda de acceso al medio," 2016.
- [25] R. G. P. Tamayo, "5G: Una nueva generación para una nueva sociedad," *HAMUT'AY*, vol. 1, no. 1, pp. 28-32, 2017.

- [26] H. Tullberg *et al.*, "The METIS 5G system concept: Meeting the 5G requirements," in *IEEE Communications Magazine*, 2016, vol. 54, no. 12, pp. 132-139: Institute of Electrical and Electronics Engineers (IEEE).
- [27] N. Van der Neut, "PAPR reduction in FBMC systems using an FBMC optimized ACE method," University of Pretoria, 2015 [En línea] Available: <https://repository.up.ac.za/handle/2263/56119> [Último acceso: 25 de septiembre de 2018].
- [28] Á. Viorreta Gómez, "Toward to design of 5G optical communications," Universitat Politècnica de Catalunya, 2017.
- [29] R. Nissel, S. Schwarz, and M. Rupp, "Filter bank multicarrier modulation schemes for future mobile communications," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 8, pp. 1768-1782, 2017.
- [30] V. Vakilian, T. Wild, F. Schaich, S. ten Brink, and J.-F. Frigon, "Universal-filtered multi-carrier technique for wireless systems beyond LTE," in *Globecom Workshops (GC Wkshps)*, 2013 *IEEE*, 2013, pp. 223-228: IEEE.

6 ANEXOS

ANEXO I. Código de Implementación de la clase `OFDM.m`

ANEXO II. Código de Implementación de la clase `FBMC.m`

ANEXO III. Código de Implementación de la clase `UFMC.m`

ANEXO IV. Código de Implementación del script `Power_Spectral_Density.m`

ANEXO V. Código de Implementación del script `Bit_Error_Rate.m`

ANEXO I

```

classdef OFDM < handle

    properties (SetAccess = private)
        No           % para almacenar parámetros adimensionales
        PHY          % para almacenar parámetros físicos
        Imp          % para almacenar parámetros de implementación
    end

    methods
        % Constructor de la clase
        function obj = OFDM(varargin)
            % Se inicializan los parámetros
            if numel(varargin) == 8
                obj.No.SubportDatos = varargin{1}; % Número de
subportadoras de datos
                obj.No.SubportxBG = varargin{2}; % Número de
subportadoras por banda de guarda
                obj.No.Simbolos = varargin{3}; % Número de símbolos
OFDM
                obj.PHY.DeltaF = varargin{4}; % Espaciamiento entre
subportadoras (Hz)
                obj.PHY.Fs = varargin{5}; % Frecuencia de muestreo
(Hz)
                obj.Imp.FreqInter = varargin{6}; % Desplazamiento en
frecuencia (para propósitos de visualización)
                obj.PHY.TiempoCP = varargin{7}; % Duración del prefijo
cíclico (s)
                obj.PHY.TiempoZG = varargin{8}; % Duración de los
intervalos de guarda (s)
            elseif numel(varargin) == 0
                % Valores predeterminados de LTE
                obj.No.SubportDatos = 72;
                obj.No.SubportxBG = 11;
                obj.No.Simbolos = 14;
                obj.PHY.DeltaF = 15e3;
                obj.PHY.Fs = 15e3*72*2;
                obj.Imp.FreqInter = 0;
                obj.PHY.TiempoCP = 1/(14*15e3);
                obj.PHY.TiempoZG = 0;
            else
                error('El número de variables de entrada debe ser 0
(valores predeterminados) u 8');
            end
            % Configurar parámetros dependientes
            obj.ConfigParamDependientes();
        end

        function ConfigParamDependientes(obj)
            obj.No.TotalSubport =
obj.No.SubportDatos+2*obj.No.SubportxBG; % Número total de subportadoras
            obj.Imp.MuestrasCP = round(obj.PHY.TiempoCP*obj.PHY.Fs); %
Número de muestras del prefijo cíclico
            obj.Imp.MuestrasZG = round(obj.PHY.TiempoZG*obj.PHY.Fs); %
Número de muestras de la banda de guarda
            obj.Imp.MuestrasSimb =
round(obj.PHY.Fs/obj.PHY.DeltaF)+obj.Imp.MuestrasCP; % Número de muestras
por símbolo OFDM
        end
    end
end

```

```

        obj.Imp.NFFT = round(obj.PHY.Fs/obj.PHY.DeltaF); % Tamaño de
la FFT
        obj.Imp.FactorNorm =
sqrt(obj.Imp.NFFT^2/obj.No.SubportDatos); % Factor de normalización para
que la potencia = 1
        obj.PHY.dt = 1/obj.PHY.Fs; % Tiempo entre muestras
        obj.PHY.TiempoSimb = obj.Imp.MuestrasSimb*obj.PHY.dt; % Tiempo
(s) de un símbolo OFDM
        obj.No.TotalMuestras =
(obj.No.Simbolos*obj.Imp.MuestrasSimb)+2*obj.Imp.MuestrasZG; % Número de
muestras totales
    end

    function SignalTX = Modulacion(obj,DatosTX)
        % El argumento de entrada es una matriz de tamaño
        % "# de subportadoras" por "# de símbolos OFDM"
        DatosTemp = zeros(obj.Imp.NFFT,obj.No.Simbolos);
        % Se convierte de S/P y se normaliza para tener potencia = 1
        Datos_S_P =
reshape(DatosTX,[],obj.No.Simbolos)*obj.Imp.FactorNorm;

        DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.SubportDatos),:)
= Datos_S_P; % Se añaden las subportadoras de guarda
        SignalTXsinCP = ifft(DatosTemp); % Se realiza la IFFT
        SignalTX =
[zeros(obj.Imp.MuestrasZG,1);reshape([SignalTXsinCP(end-
obj.Imp.MuestrasCP+1:end,:);SignalTXsinCP],[],1);...
zeros(obj.Imp.MuestrasZG,1)]; % Se añade el prefijo
cíclico, se convierte de P/S y se añaden los intervalos de guarda
    end

    function DatosRX = Demodulacion(obj, SignalRX)
        % Se demodula y retorna una matriz de tamaño
        % "# de subportadoras" por "# de símbolos OFDM"
        % Se eliminan los intervalos de guarda y se convierte de S/P
        SignalTemp = reshape(SignalRX((obj.Imp.MuestrasZG+1):(end-
obj.Imp.MuestrasZG)),obj.Imp.MuestrasSimb,obj.No.Simbolos);
        SimbolosTemp = fft(SignalTemp(obj.Imp.MuestrasCP+1:end,:)); %
Se elimina el prefijo cíclico y se realiza la FFT
        DatosTemp =
SimbolosTemp(obj.Imp.FreqInter+(1:obj.No.TotalSubport),:)/(obj.Imp.Factor
Norm); % Se normaliza para tener potencia = 1
        DatosRX =
reshape(DatosTemp(obj.No.SubportxBG+(1:obj.No.SubportDatos),:),[],1); %
Se convierte de P/S
    end
end
end

```

ANEXO II

```
classdef FBMC < handle

    properties (SetAccess = private)
        No           % para almacenar parámetros adimensionales
        PHY           % para almacenar parámetros físicos
        PFiltro       % para almacenar parámetros del filtro prototipo
        Imp           % para almacenar parámetros de implementación
    end

    methods
        % Constructor de la clase
        function obj = FBMC(varargin)
            % Se inicializan los parámetros
            if numel(varargin) == 8
                obj.No.SubportDatos = varargin{1}; % Número de
subportadoras de datos
                obj.No.SubportxBG = varargin{2}; % Número de
subportadoras por banda de guarda
                obj.No.Simbolos = varargin{3}; % Número de símbolos
FBMC
                obj.PHY.DeltaF = varargin{4}; % Espaciamiento entre
subportadoras (Hz)
                obj.PHY.Fs = varargin{5}; % Frecuencia de muestreo
(Hz)
                obj.Imp.FreqInter = varargin{6}; % Desplazamiento en
frecuencia (para propósitos de visualización)
                obj.PFiltro.K = varargin{7}; % Factor de solapamiento
                obj.Imp.UsarPPN = varargin{8}; % Implementación de una
red polifase (PPN)
            elseif numel(varargin) == 0
                % Valores predeterminados de LTE
                obj.No.SubportDatos = 72;
                obj.No.SubportxBG = 11;
                obj.No.Simbolos = 14;
                obj.PHY.DeltaF = 15e3;
                obj.PHY.Fs = 15e3*72*2;
                obj.Imp.FreqInter = 0;
                obj.PFiltro.K = 4;
                obj.Imp.UsarPPN = true;
            else
                error('El número de variables de entrada debe ser 0
(valores predeterminados) u 8');
            end
            % Configurar parámetros dependientes
            obj.ConfigParamDependientes();
        end

        function ConfigParamDependientes(obj)
            obj.PHY.dt = 1/obj.PHY.Fs; % Tiempo entre muestras
            obj.No.TotalSubport =
obj.No.SubportDatos+2*obj.No.SubportxBG; % Número total de subportadoras
            obj.Imp.MuestrasSimb = round(obj.PHY.Fs/(2*obj.PHY.DeltaF));
            % Número de muestras por símbolo FBMC
            obj.PHY.TiempoSimb = obj.Imp.MuestrasSimb*obj.PHY.dt; % Tiempo
(s) de un símbolo FBMC
            obj.Imp.FactorSolap = obj.PFiltro.K; % Factor de solapamiento
            % Filtro 'PHYDYAS-OQAM'
        end
    end
end
```

```

        obj.PFiltro.DominioT =
FiltroPrototipo_PHYDYAS(obj.PHY.TiempoSimb*2,obj.PHY.dt,obj.PFiltro.K); %
Respuesta al impulso del filtro prototipo
        obj.No.MuestrasPFiltro = length(obj.PFiltro.DominioT); %
Número de muestras del filtro prototipo
        obj.No.TotalMuestras =
obj.No.MuestrasPFiltro+(obj.No.Simbolos-1)*obj.Imp.MuestrasSimb; % Número
de muestras totales
        PFiltroFFT =
fft(circshift(obj.PFiltro.DominioT,obj.No.MuestrasPFiltro/2)); %
Respuesta en frecuencia del filtro prototipo
        PFiltroFFT(abs(PFiltroFFT)./PFiltroFFT(1)<10^-14) = 0; % Para
reducir la complejidad computacional
        obj.PFiltro.DominioF = PFiltroFFT/max(PFiltroFFT); %
Coeficientes en frecuencia del filtro prototipo
        % Indices para el solapamiento en tiempo de los símbolos FBMC
        PosdSolap =
[ones(obj.No.MuestrasPFiltro,obj.No.Simbolos);zeros(obj.Imp.MuestrasSimb*
(obj.No.Simbolos-1),obj.No.Simbolos)];
        IndxdSolap = zeros(obj.No.MuestrasPFiltro,obj.No.Simbolos);
        for i_k = 1:obj.No.Simbolos
            PosdSolap(:,i_k) = circshift(PosdSolap(:,i_k),(i_k-
1)*obj.Imp.MuestrasSimb);
            IndxdSolap(:,i_k) = find(PosdSolap(:,i_k));
        end
        obj.Imp.PosdSolap = logical(PosdSolap);
        obj.Imp.IndxdSolap = IndxdSolap;
        obj.Imp.NFFT =
round(obj.No.MuestrasPFiltro./obj.Imp.FactorSolap); % Tamaño de la FFT
        % Factor de normalización para que la potencia = 1
        obj.Imp.FactorNorm =
sqrt(obj.Imp.NFFT^2/obj.No.SubportDatos)/(obj.Imp.FactorSolap);
    end

    function SignalTX = Modulacion(obj, DatosTX)
        % El argumento de entrada es una matriz de tamaño
        % "# de subportadoras/2" por "# de símbolos FBMC"
        SignalTX = zeros(size(obj.Imp.PosdSolap));
        Datos_S_P = reshape(DatosTX,[],obj.No.Simbolos); % Se
convierte de S/P
        DatosOQAM = oqammod(Datos_S_P); % Se realiza la modulación OQAM
        if obj.Imp.UsarPPN
            % Método Red polifase (PPN)
            DatosTemp = zeros(obj.Imp.NFFT,obj.No.Simbolos);

            DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.SubportDatos),:)
            = DatosOQAM*obj.Imp.FactorNorm; % Se normaliza para tener potencia = 1
            % Se realiza la IFFT, se filtra y se solapan los símbolos
            SignalTX(obj.Imp.PosdSolap)
            = repmat(iff(DatosTemp),[obj.Imp.FactorSolap
1]).*repmat(obj.PFiltro.DominioT,[1 obj.No.Simbolos]);
            SignalTX = sum(SignalTX,2); % Se suman los símbolos
solapados
        else
            % Método IFFT extendida
            GenPFiltros =
zeros(size(obj.PFiltro.DominioF,1),obj.No.SubportDatos);
            for i_1 = 1:obj.No.SubportDatos
                % Se generan los filtros para cada subportadora a
                % partir de los coeficientes del filtro prototipo

```

```

        GenPFiltros(:,i_1) = circshift(obj.PFiltro.DominioF,
obj.Imp.FactorSolap*(i_1-1+obj.Imp.FreqInter+obj.No.SubportxBG));
        end
        SignalTemp =
GenPFiltros*DatosOQAM*(obj.Imp.FactorNorm*obj.Imp.FactorSolap); % Se
filtra y normaliza para tener potencia = 1
        % Se realiza la IFFT extendida y se solapan los símbolos
        SignalTX(obj.Imp.PosdSolap) =
circshift(iffc(SignalTemp),[-obj.No.MuestrasPFiltro/2 0]);
        SignalTX = sum(SignalTX,2); % Se suman los símbolos
solapados
    end
end

function DatosRX = Demodulacion(obj, SignalRX)
    % Se demodula y retorna una matriz de tamaño
    % "# de subportadoras" por "# de símbolos FBMC"
    Signal_SimbFBMC = SignalRX(obj.Imp.IndxdSolap); % Se obtienen
los símbolos FBMC de la señal recibida (S/P)
    if obj.Imp.UsarPPN
        % Método Red polifase (PPN)
        SignalFiltrada =
Signal_SimbFBMC.*repmat(obj.PFiltro.DominioT,[1 obj.No.Simbolos]); % Se
filtra los símbolos recibidos
        DatosTemp = fft(sum(reshape(SignalFiltrada,[obj.Imp.NFFT
obj.Imp.FactorSolap obj.No.Simbolos]),2)); % Se realiza la FFT
        % Se normaliza para tener potencia = 1
        DatosOQAM =
DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.SubportDatos),:)/
(obj.Imp.FactorNorm*obj.Imp.FactorSolap^2);
    else
        % Método FFT extendida
        GenPFiltros =
zeros(size(obj.PFiltro.DominioF,1),obj.No.SubportDatos);
        for i_1 = 1:obj.No.SubportDatos
            % Igual que en el transmisor
            GenPFiltros(:,i_1) = circshift(obj.PFiltro.DominioF,
obj.Imp.FactorSolap*(i_1-1+obj.Imp.FreqInter+obj.No.SubportxBG));
            end
            DatosTemp =
fft(circshift(Signal_SimbFBMC,[obj.No.MuestrasPFiltro/2 0])); % Se realiza
la FFT extendida
            DatosOQAM =
(GenPFiltros'*DatosTemp)/(obj.Imp.FactorNorm*obj.Imp.FactorSolap^2); % Se
filtra y normaliza para tener potencia = 1
            end
            Datos_P_S = oqamdemod(DatosOQAM); % Se realiza la demodulación
OQAM
            DatosRX = Datos_P_S(:); % Se convierte de P/S
        end
    end
end
end

function PFiltro = FiltroPrototipo_PHYDYAS(T0,dt,OF)
    t_fil = -(OF*T0/2):dt:(OF*T0/2-dt);
    switch OF % Coeficientes del filtro prototipo en el dominio de la
frecuencia de acuerdo al factor de solapamiento

```

```

    case 2
        H = sqrt(2)/2;
    case 3
        H = [0.91143783 0.41143783];
    case 4
        H = [0.97195983 sqrt(2)/2 0.23514695];
    otherwise
        error('El factor de solapamiento debe ser un entero entre 2 y
4');
end
% Filtro Prototipo en el dominio del tiempo (Respuesta al impulso)
PFiltro =
1+2*sum(repmat(H,length(t_fil),1).*cos(2*pi*repmat(t_fil',1,length(H)).*r
epmat(1:length(H),length(t_fil),1)/((length(H)+1)*T0)),2);
end

function [ SimbOQAM ] = oqammod( SimbQAM )
% Conversión de QAM a OQAM
[L, K]=size(SimbQAM); % L = # de subportadoras/2, K = # de símbolos
SimbOQAM=zeros(2*L,K);
for i_simb = 1:K
    if rem(i_simb,2)==1 % Símbolos impares
        SimbOQAM(1:2:end,i_simb) = real(SimbQAM(:,i_simb));
        SimbOQAM(2:2:end,i_simb) = 1i*imag(SimbQAM(:,i_simb));
    else % Símbolos pares
        SimbOQAM(1:2:end,i_simb) = 1i*imag(SimbQAM(:,i_simb));
        SimbOQAM(2:2:end,i_simb) = real(SimbQAM(:,i_simb));
    end
end
end

function [ SimbQAM ] = oqamdemod( SimbOQAM )
% Conversión de OQAM a QAM
[L, K]=size(SimbOQAM); % L = # de subportadoras, K = # de símbolos
SimbQAM=zeros(L/2,K);
for i_simb = 1:K
    if rem(i_simb,2)==1 % Símbolos impares
        rQAM = real(SimbOQAM(1:2:end,i_simb));
        iQAM = imag(SimbOQAM(2:2:end,i_simb));
    else % Símbolos pares
        iQAM = imag(SimbOQAM(1:2:end,i_simb));
        rQAM = real(SimbOQAM(2:2:end,i_simb));
    end
    SimbQAM(:,i_simb)=complex(rQAM,iQAM);
end
end

```

ANEXO III

```
classdef UFMC < handle

    properties (SetAccess = private)
        No           % para almacenar parámetros adimensionales
        PHY          % para almacenar parámetros físicos
        Imp          % para almacenar parámetros de implementación
    end

    methods
        % Constructor de la clase
        function obj = UFMC(varargin)
            % Se inicializan los parámetros
            if numel(varargin) == 8
                obj.No.SubportDatos = varargin{1}; % Número de
                subportadoras de datos
                obj.No.SubportxBG = varargin{2}; % Número de
                subportadoras por banda de guarda
                obj.No.Simbolos = varargin{3}; % Número de símbolos
                obj.PHY.DeltaF = varargin{4}; % Espaciamiento entre
                subportadoras (Hz)
                obj.PHY.Fs = varargin{5}; % Frecuencia de muestreo
                (Hz)
                obj.Imp.FreqInter = varargin{6}; % Desplazamiento en
                frecuencia (para propósitos de visualización)
                obj.PHY.TiempoZG = varargin{7}; % Duración de los
                intervalos de guarda (s)
                obj.PHY.LongFiltro = varargin{8}; % Longitud del filtro
                (s)
            elseif numel(varargin) == 0
                % Valores predeterminados de LTE
                obj.No.SubportDatos = 72;
                obj.No.SubportxBG = 12;
                obj.No.Simbolos = 14;
                obj.PHY.DeltaF = 15e3;
                obj.PHY.Fs = 15e3*72*2;
                obj.Imp.FreqInter = 0;
                obj.PHY.TiempoZG = 0;
                obj.PHY.LongFiltro = 1/(14*15e3);
            else
                error('El número de variables de entrada debe ser 0
                (valores predeterminados) u 8');
            end
            % Configurar parámetros dependientes
            obj.ConfigParamDependientes();
        end

        function ConfigParamDependientes(obj)
            obj.No.TotalSubport =
            obj.No.SubportDatos+2*obj.No.SubportxBG; % Número total de subportadoras =
            obj.Imp.MuestrasZG = round( obj.PHY.TiempoZG * obj.PHY.Fs );
            % Número de muestras de la banda de guarda
            obj.Imp.NFFT = round( obj.PHY.Fs / obj.PHY.DeltaF ); % Tamaño
            de la FFT
            % Factor de normalización para que la potencia = 1
        end
    end
end
```

```

obj.Imp.FactorNorm =
sqrt(obj.Imp.NFFT/obj.No.SubportDatos^2)*21.15/(1.385^(log2(obj.Imp.NFFT/
obj.No.SubportDatos)-1));
obj.PHY.dt = 1 / obj.PHY.Fs; % Tiempo entre muestras
obj.Imp.LongFiltro = round(obj.PHY.LongFiltro *obj.PHY.Fs)+1;
% Longitud del filtro
% Filtrado de subbanda con ventana Dolph-Chebyshev
FiltroTX_Tiempo = chebwin(obj.Imp.LongFiltro,60); % Filtro
base en tiempo
FiltroTX_Freq = fft(FiltroTX_Tiempo,obj.Imp.NFFT); % Filtro
base en frecuencia
LongSubBanda_LTE = 12; % Número de subportadoras por RB
% Encontrar longitudes de subbanda válidas
FactorNoSubportDatos = factor(obj.No.SubportDatos);
temp = [];
if length(FactorNoSubportDatos)<=2
longitud de subbanda
temp = FactorNoSubportDatos'; % Posibles valores de
end
for i_fac=2:length(FactorNoSubportDatos)-1
aux = prod(nchoosek([1 FactorNoSubportDatos],i_fac),2);
temp = [temp; aux]; % Posibles valores de longitud de
subbanda
end
PosibleLongSubBanda = unique(temp); % Se descarta valores
repedidos
[~,a] = min(abs(PosibleLongSubBanda - LongSubBanda_LTE)); % Se
elige una longitud lo más cercana al tamaño de un RB de LTE
obj.No.LongSubBanda = PosibleLongSubBanda(a); % Se almacena la
longitud elegida
% Se muestra una advertencia si la longitud de la subbanda no
% es igual a un RB de LTE
if not(LongSubBanda_LTE==obj.No.LongSubBanda) &&
(obj.Imp.LongFiltro>1)
warning(['El subfiltrado puede no ser preciso. Longitud de
subbanda deseada: ' int2str(LongSubBanda_LTE) '. Longitud elegida: ',...
int2str(obj.No.LongSubBanda)]);
end
% Generación de los filtros de cada subbanda
obj.No.NoSubBandas =
ceil(obj.No.SubportDatos/obj.No.LongSubBanda); % Se obtiene el número de
subbandas
FiltroTX = nan(obj.Imp.NFFT,obj.No.NoSubBandas);
for i_subband = 1: obj.No.NoSubBandas
% Se generan los filtros a partir del filtro base
FiltroTX(:,i_subband) =
ifft(circshift(FiltroTX_Freq,[ceil(obj.No.LongSubBanda/2)-
1+obj.Imp.FreqInter+obj.No.SubportxBG+...
(i_subband-1)*obj.No.LongSubBanda 0]));
end
obj.Imp.RespImpFiltro = FiltroTX(1:obj.Imp.LongFiltro,:); %
Filtros finales por subbanda
obj.Imp.MuestrasSimb = obj.Imp.NFFT+obj.Imp.LongFiltro-1; %
Número de muestras por símbolo UPMC
obj.PHY.TiempoSimb = obj.Imp.MuestrasSimb * obj.PHY.dt; %
Tiempo (s) de un símbolo UPMC
obj.No.TotalMuestras =
obj.Imp.MuestrasSimb*obj.No.Simbolos+2*obj.Imp.MuestrasZG; % Número de
muestras totales

```

```

        H_FiltroRX =
fft([obj.Imp.RespImpFiltro(:,1);zeros(obj.Imp.NFFT-
obj.Imp.LongFiltro,1)]); % Función de transferencia del filtro
        obj.Imp.FiltroEqRX =
1./H_FiltroRX(obj.Imp.FreqInter+obj.No.SubportxBG+(1:obj.No.LongSubBanda)
); % Filtro de equalización en recepción
    end

    function SignalTX = Modulacion(obj,DatosTX)
        % El argumento de entrada es una matriz de tamaño
        % "# de subportadoras" por "# de símbolos UFMC"
        SignalTX_Temp = zeros(obj.Imp.NFFT+obj.Imp.LongFiltro-
1,obj.No.NoSubBandas);
        SignalTX_Temp2 = zeros(obj.Imp.NFFT+obj.Imp.LongFiltro-
1,obj.No.Simbolos);
        Datos_S_P =
reshape(DatosTX,[],obj.No.Simbolos)*obj.Imp.FactorNorm; % Se convierte de
S/P y se normaliza para tener potencia = 1
        for i_k=1:obj.No.Simbolos
            for i_subband = 1:obj.No.NoSubBandas
                DatosTemp = zeros(obj.Imp.NFFT,1);
                % Se seleccionan las subportadoras de la subbanda
                SubbandIdx = (1:obj.No.LongSubBanda)+(i_subband-
1)*obj.No.LongSubBanda;

                DatosTemp(obj.Imp.FreqInter+obj.No.SubportxBG+SubbandIdx) =
                Datos_S_P(SubbandIdx,i_k); % Se añaden las subportadoras de guarda
                SignalTemp = ifft(DatosTemp); % Se realiza la IFFT
                SignalTX_Temp(:,i_subband) =
conv(SignalTemp,obj.Imp.RespImpFiltro(:,i_subband)); % Se filtra la
subbanda
            end
            SignalTX_Temp2(:,i_k) = sum(SignalTX_Temp,2); % Se suma y
obtiene el símbolo UFMC
        end
        % Se convierte de P/S y se añaden los intervalos de guarda
        SignalTX =
[zeros(obj.Imp.MuestrasZG,1);reshape(SignalTX_Temp2,[],1);zeros(obj.Imp.M
uestrasZG,1)];
    end
    function DatosRX = Demodulacion(obj, SignalRX)
        % Se demodula y retorna una matriz de tamaño
        % "# de subportadoras" por "# de símbolos UFMC"
        % Se eliminan los intervalos de guarda y se convierte de S/P
        SignalTemp = reshape(SignalRX((obj.Imp.MuestrasZG+1):(end-
obj.Imp.MuestrasZG)),obj.Imp.MuestrasSimb,obj.No.Simbolos);
        SimbTemp = fft(SignalTemp,obj.Imp.NFFT*2); % Se realiza la FFT
de tamaño 2N
        SimbTemp = SimbTemp(1:2:end,:); % Se realiza un diezmado de
factor 2
        DatosTemp =
SimbTemp(obj.Imp.FreqInter+(1:obj.No.TotalSubport),:)/(obj.Imp.FactorNorm
); % Se normaliza para tener potencia = 1
        % Se equalizan los símbolos QAM recibidos
        DatosEq = reshape(DatosTemp(obj.No.SubportxBG+1:end-
obj.No.SubportxBG,:),obj.No.LongSubBanda,[]).*obj.Imp.FiltroEqRX;
        DatosRX = DatosEq(:); % Se convierte de P/S
    end
end
end
end

```



```

    FBMC_Factor_Solapamiento(1),... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN...      % Implementación de una red polifase (PPN)
    );
% FBMC con K = 3
FBMC3 = FBMC(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,...  % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos FBMC
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    FBMC_Factor_Solapamiento(2),... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN...      % Implementación de una red polifase (PPN)
    );
% FBMC con K = 4
FBMC4 = FBMC(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,...  % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos FBMC
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    FBMC_Factor_Solapamiento(3),... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN...      % Implementación de una red polifase (PPN)
    );

%% Objeto OFDM
OFDM = OFDM(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,...  % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos OFDM
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    OFDM_CP,...     % Duración del prefijo cíclico (s)
    T_IG...         % Duración de los intervalos de guarda (s)
    );

%% Objeto UFMC
UFMC = UFMC(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,...  % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos UFMC
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    T_IG,...         % Duración de los intervalos de guarda (s)
    UFMC_LongFiltro... % Longitud del filtro (s)
    );

%% Información
fprintf('=====\n');

```

```

fprintf('=====\\n');
fprintf('          | TF spacing | AB utilizado | AB Total |\\n');
fprintf('OFDM (con CP) | %10.2f | %8.2f MHz | %6.2f MHz
|\\n', OFDM.PHY.TiempoSimb*OFDM.PHY.DeltaF, OFDM.PHY.DeltaF*OFDM.No.SupportD
atos/1e6, ...
OFDM.PHY.DeltaF*OFDM.No.TotalSupport/1e6);
fprintf('FBMC-OQAM          | %10.2f | %8.2f MHz | %6.2f MHz
|\\n', FBMC2.PHY.TiempoSimb*FBMC2.PHY.DeltaF*2, FBMC2.PHY.DeltaF*FBMC2.No.Su
bportDatos/1e6, ...
FBMC2.PHY.DeltaF*FBMC2.No.TotalSupport/1e6);
fprintf('UFMC          | %10.2f | %8.2f MHz | %6.2f MHz
|\\n', UFMC.PHY.TiempoSimb*UFMC.PHY.DeltaF, UFMC.PHY.DeltaF*UFMC.No.SupportD
atos/1e6, ...
UFMC.PHY.DeltaF*UFMC.No.TotalSupport/1e6);
fprintf('=====\\n');
fprintf('=====\\n');

%% Simulación
P_FBMC2 = zeros(FBMC2.No.TotalMuestras,1);
P_FBMC3 = zeros(FBMC3.No.TotalMuestras,1);
P_FBMC4 = zeros(FBMC4.No.TotalMuestras,1);
P_OFDM = zeros(OFDM.No.TotalMuestras,1);
P_UFMC = zeros(UFMC.No.TotalMuestras,1);

PSD_FBMC2_Simulacion = zeros(FBMC2.No.TotalMuestras,1);
PSD_FBMC3_Simulacion = zeros(FBMC3.No.TotalMuestras,1);
PSD_FBMC4_Simulacion = zeros(FBMC4.No.TotalMuestras,1);
PSD_OFDM_Simulacion = zeros(OFDM.No.TotalMuestras,1);
PSD_UFMC_Simulacion = zeros(UFMC.No.TotalMuestras,1);

f_FBMC2 = (0:FBMC2.No.TotalMuestras-1)/(FBMC2.No.TotalMuestras*dt);
f_FBMC3 = (0:FBMC3.No.TotalMuestras-1)/(FBMC3.No.TotalMuestras*dt);
f_FBMC4 = (0:FBMC4.No.TotalMuestras-1)/(FBMC4.No.TotalMuestras*dt);
f_OFDM = (0:OFDM.No.TotalMuestras-1)/(OFDM.No.TotalMuestras*dt);
f_UFMC = (0:UFMC.No.TotalMuestras-1)/(UFMC.No.TotalMuestras*dt);

tic; % Se inicia un cronómetro
for i_rep = 1:No_Repeticion
    % Se generan los bits de datos
    Data_FBMC = randi([0 1],NoSubP_Datos/2*NoSimbolos*log2(M_QAM),1);
    Data_OFDM = randi([0 1],NoSubP_Datos*NoSimbolos*log2(M_QAM),1);
    Data_UFMC = randi([0 1],NoSubP_Datos*NoSimbolos*log2(M_QAM),1);
    % Se realiza la modulación 16-QAM (para FBMC se utiliza modulación
OQAM)
    x_FBMC =
qammod(Data_FBMC,M_QAM,'InputType','bit','UnitAveragePower',true);
    x_OFDM =
qammod(Data_OFDM,M_QAM,'InputType','bit','UnitAveragePower',true);
    x_UFMC =
qammod(Data_UFMC,M_QAM,'InputType','bit','UnitAveragePower',true);
    % Se añaden las portadoras de guarda y se modula
    s_FBMC2 = FBMC2.Modulacion(x_FBMC);
    s_FBMC3 = FBMC3.Modulacion(x_FBMC);
    s_FBMC4 = FBMC4.Modulacion(x_FBMC);
    s_OFDM = OFDM.Modulacion(x_OFDM);
    s_UFMC = UFMC.Modulacion(x_UFMC);
    % Se calcula la potencia de la i-ésima simulación y se añade a la
% potencia acumulada
    P_FBMC2 = P_FBMC2+abs(s_FBMC2).^2;

```

```

P_FBMC3 = P_FBMC3+abs(s_FBMC3).^2;
P_FBMC4 = P_FBMC4+abs(s_FBMC4).^2;
P_OFDM = P_OFDM+abs(s_OFDM).^2;
P_UFMC = P_UFMC+abs(s_UFMC).^2;
% Se calcula la PSD de la i-ésima simulación y se añade a la PSD
% acumulada
PSD_FBMC2_Simulacion =
PSD_FBMC2_Simulacion+(abs(fft(s_FBMC2)).^2)/FBMC2.No.TotalMuestras;
PSD_FBMC3_Simulacion =
PSD_FBMC3_Simulacion+(abs(fft(s_FBMC3)).^2)/FBMC3.No.TotalMuestras;
PSD_FBMC4_Simulacion =
PSD_FBMC4_Simulacion+(abs(fft(s_FBMC4)).^2)/FBMC4.No.TotalMuestras;
PSD_OFDM_Simulacion =
PSD_OFDM_Simulacion+(abs(fft(s_OFDM)).^2)/OFDM.No.TotalMuestras;
PSD_UFMC_Simulacion =
PSD_UFMC_Simulacion+(abs(fft(s_UFMC)).^2)/UFMC.No.TotalMuestras;

Tiempo_Sim = toc; % Se almacena el tiempo actual del cronómetro
if (mod(i_rep,25)==0) || (i_rep==5)
    Tiempo_Rest = Tiempo_Sim/i_rep*(No_Repeticion-i_rep); % Se realiza
una estimación del tiempo restante para finalizar la simulación
    Hora = floor(Tiempo_Rest/3600);
    Min = floor(rem(Tiempo_Rest,3600)/60);
    Seg = (rem(rem(Tiempo_Rest,3600),60));
    fprintf('Simulación %d de %d. ',i_rep,No_Repeticion);
    if i_rep<No_Repeticion
        fprintf('Tiempo restante aprox.: %d h %d min %.3f
s.\n',Hora,Min,Seg);
    end
end
end
Hora = floor(Tiempo_Sim/3600);
Min = floor(rem(Tiempo_Sim,3600)/60);
Seg = (rem(rem(Tiempo_Sim,3600),60));
if mod(i_rep,25)==0
    fprintf('Tiempo total de simulación: %d h %d min %.3f
s.\n',Hora,Min,Seg);
else
    fprintf('Simulación %d de %d. Tiempo total de simulación: %d h %d min
%.3f s.\n',i_rep,No_Repeticion,Hora,Min,Seg);
end
% Se calcula el promedio
P_FBMC2 = P_FBMC2/No_Repeticion;
P_FBMC3 = P_FBMC3/No_Repeticion;
P_FBMC4 = P_FBMC4/No_Repeticion;
P_OFDM = P_OFDM/No_Repeticion;
P_UFMC = P_UFMC/No_Repeticion;

PSD_FBMC2_Simulacion = PSD_FBMC2_Simulacion/No_Repeticion;
PSD_FBMC3_Simulacion = PSD_FBMC3_Simulacion/No_Repeticion;
PSD_FBMC4_Simulacion = PSD_FBMC4_Simulacion/No_Repeticion;
PSD_OFDM_Simulacion = PSD_OFDM_Simulacion/No_Repeticion;
PSD_UFMC_Simulacion = PSD_UFMC_Simulacion/No_Repeticion;
% Se normaliza a 0dB
PSD_FBMC2_Simulacion = PSD_FBMC2_Simulacion/max(PSD_FBMC2_Simulacion);
PSD_FBMC3_Simulacion = PSD_FBMC3_Simulacion/max(PSD_FBMC3_Simulacion);
PSD_FBMC4_Simulacion = PSD_FBMC4_Simulacion/max(PSD_FBMC4_Simulacion);
PSD_OFDM_Simulacion = PSD_OFDM_Simulacion/max(PSD_OFDM_Simulacion);
PSD_UFMC_Simulacion = PSD_UFMC_Simulacion/max(PSD_UFMC_Simulacion);

```

```

ColorFBMC2 = [1 0 1]*0.6;
ColorFBMC3 = [0 1 1]*0.6;
ColorFBMC4 = [0 0 1]*0.6;
ColorOFDM = [1 0 0]*0.7;
ColorUFMC = [0 1 0]*0.8;
% Gráfica de PSD de Simulación vs delta f
figure;
plot(f_FBMC2/delta_f-FI-
ceil(NoSubP_Total/2),10*log10(PSD_FBMC2_Simulacion),'Color',ColorFBMC2);
hold on;
plot(f_FBMC3/delta_f-FI-
ceil(NoSubP_Total/2),10*log10(PSD_FBMC3_Simulacion),'Color',ColorFBMC3);
plot(f_FBMC4/delta_f-FI-
ceil(NoSubP_Total/2),10*log10(PSD_FBMC4_Simulacion),'Color',ColorFBMC4);
plot(f_OFDM/delta_f-FI-
ceil(NoSubP_Total/2),10*log10(PSD_OFDM_Simulacion),'Color',ColorOFDM);
plot(f_UFMC/delta_f-FI-
ceil(NoSubP_Total/2),10*log10(PSD_UFMC_Simulacion),'Color',ColorUFMC);
plot([-NoSubP_Datos/2 -NoSubP_Datos/2],[-200,20],'color',[0.9 0.8 0]); %
Posición de la primera subportadora
plot([NoSubP_Datos/2-1 NoSubP_Datos/2-1],[-200,20],'color',[0.9 0.8 0]); %
Posición de la última subportadora
plot([-NoSubP_Total/2 -NoSubP_Total/2],[-200,20],'color',[0.2 0.3 0.2]); %
Límite superior del canal
plot([NoSubP_Total/2-1 NoSubP_Total/2-1],[-200,20],'color',[0.2 0.3
0.2]); % Límite inferior del canal
ylim([-145 5]); % Límites en el eje Y
xlim([-NoSubP_Datos*2 NoSubP_Datos*2]); % Límites en el eje X
grid on;
legend({'FBMC, K = 2','FBMC, K = 3','FBMC, K = 4','OFDM','UFMC'});
xlabel('Frecuencia Normalizada, f/\Deltaf');
ylabel('Power Spectral Density [dB]');
title('Simulación');
%% Cálculo de la eficiencia espectral
Nsimb=1:280; % Número de símbolos para OFDM y UFMC
Nsimb_FBMC=1:2*max(Nsimb); % Número de símbolos para FBMC
BW=OFDM.PHY.DeltaF*OFDM.No.TotalSubport; % Ancho de banda total
% Obtención de las diferentes duraciones de ráfaga
t_OFDM=Nsimb*OFDM.PHY.TiempoSimb;
t_UFMC=Nsimb*UFMC.PHY.TiempoSimb;
t_FBMC2=FBMC2.Imp.FactorSolap/delta_f+0.5/delta_f*(Nsimb_FBMC-1);
t_FBMC3=FBMC3.Imp.FactorSolap/delta_f+0.5/delta_f*(Nsimb_FBMC-1);
t_FBMC4=FBMC4.Imp.FactorSolap/delta_f+0.5/delta_f*(Nsimb_FBMC-1);
% Obtención de la eficiencia espectral en función de la duración de ráfaga
ef_OFDM=(NoSubP_Datos*log2(M_QAM)*Nsimb./t_OFDM)/BW;
ef_UFMC=(NoSubP_Datos*log2(M_QAM)*Nsimb./t_UFMC)/BW;
ef_FBMC2=(NoSubP_Datos/2*log2(M_QAM)*Nsimb_FBMC./t_FBMC2)/BW;
ef_FBMC3=(NoSubP_Datos/2*log2(M_QAM)*Nsimb_FBMC./t_FBMC3)/BW;
ef_FBMC4=(NoSubP_Datos/2*log2(M_QAM)*Nsimb_FBMC./t_FBMC4)/BW;
% Gráfica de Eficiencia espectral vs Duración de ráfaga
figure;
plot(t_OFDM/1e-3,ef_OFDM,'-
o','color',ColorOFDM,'MarkerIndices',3:10:length(ef_UFMC),'MarkerSize',2,
'MarkerFaceColor',ColorOFDM); hold on;
plot(t_UFMC/1e-3,ef_UFMC,'--
o','color',ColorUFMC,'MarkerIndices',8:10:length(ef_UFMC),'MarkerSize',2,
'MarkerFaceColor',ColorUFMC);
plot(t_FBMC2/1e-3,ef_FBMC2,'color',ColorFBMC2);
plot(t_FBMC3/1e-3,ef_FBMC3,'color',ColorFBMC3);

```

```
plot(t_FBMC4/1e-3,ef_FBMC4,'color',ColorFBMC4);
grid on;
legend({'OFDM','UFMC','FBMC, K = 2','FBMC, K = 3','FBMC, K =
4'},'Location','SouthEast');
xlabel('Duración de ráfaga [ms]')
ylabel('Eficiencia Espectral [bps/Hz]')
title('Eficiencia Espectral')
```



```

    FBMC_Factor_Solapamiento(1),... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN...      % Implementación de una red polifase (PPN)
    );
% FBMC con K = 3
FBMC3 = FBMC(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,... % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos FBMC
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    FBMC_Factor_Solapamiento(2),... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN...      % Implementación de una red polifase (PPN)
    );
% FBMC con K = 4
FBMC4 = FBMC(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,... % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos FBMC
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    FBMC_Factor_Solapamiento(3),... % Factor de solapamiento; 1 = 2, 2 =
3, 3 = 4
    FBMC_PPN...      % Implementación de una red polifase (PPN)
    );
% Objeto OFDM
OFDM = OFDM(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,... % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos OFDM
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    OFDM_CP,...     % Duración del prefijo cíclico (s)
    T_IG...         % Duración de los intervalos de guarda (s)
    );
% Objeto UFMC
UFMC = UFMC(...
    NoSubP_Datos,... % Número de subportadoras de datos
    NoSubP_x_BG,... % Número de subportadoras por banda de guarda
    NoSimbolos,...  % Número de símbolos UFMC
    delta_f,...     % Espaciamiento entre subportadoras (Hz)
    Fs,...          % Frecuencia de muestreo (Hz)
    FI,...          % Desplazamiento en frecuencia (para propósitos de
visualización)
    T_IG,...        % Duración de los intervalos de guarda (s)
    UFMC_LongFiltro... % Longitud del filtro (s)
);

N_snr = length(SNR_dB);
BER_FBMC2 = zeros(N_snr,No_Repeticion);
BER_FBMC3 = BER_FBMC2;
BER_FBMC4 = BER_FBMC2;
BER_OFDM = BER_FBMC2;

```

```

BER_UFMC = BER_FBMC2;

%% Simulación
tic; % Se inicia un cronómetro
for i_rep = 1:No_Repeticion
    % Se generan los bits de datos
    Data_FBMC = randi([0 1],NoSubP_Datos/2*NoSimbolos*log2(M_QAM),1);
    Data_OFDM = randi([0 1],NoSubP_Datos*NoSimbolos*log2(M_QAM),1);
    Data_UFMC = randi([0 1],NoSubP_Datos*NoSimbolos*log2(M_QAM),1);
    % Se realiza la modulación 16-QAM (para FBMC se utiliza modulación
OQAM)
    x_FBMC
qammod(Data_FBMC,M_QAM,'InputType','bit','UnitAveragePower',true); =
    x_OFDM
qammod(Data_OFDM,M_QAM,'InputType','bit','UnitAveragePower',true); =
    x_UFMC
qammod(Data_UFMC,M_QAM,'InputType','bit','UnitAveragePower',true); =
    % Se añaden las portadoras de guarda y se modula
    TX_FBMC2 = FBMC2.Modulacion(x_FBMC);
    TX_FBMC3 = FBMC3.Modulacion(x_FBMC);
    TX_FBMC4 = FBMC4.Modulacion(x_FBMC);
    TX_OFDM = OFDM.Modulacion(x_OFDM);
    TX_UFMC = UFMC.Modulacion(x_UFMC);

    for i_SNR = 1:N_snr
        snrdb = SNR_dB(i_SNR);
        % Se añade ruido AWGN
        RX_FBMC2 = awgn(TX_FBMC2,snrdb,'measured');
        RX_FBMC3 = awgn(TX_FBMC3,snrdb,'measured');
        RX_FBMC4 = awgn(TX_FBMC4,snrdb,'measured');
        RX_OFDM = awgn(TX_OFDM,snrdb,'measured');
        RX_UFMC = awgn(TX_UFMC,snrdb,'measured');
        % Se demodula la señal recibida
        y_FBMC2 = FBMC2.Demodulacion(RX_FBMC2);
        y_FBMC3 = FBMC3.Demodulacion(RX_FBMC3);
        y_FBMC4 = FBMC4.Demodulacion(RX_FBMC4);
        y_OFDM = OFDM.Demodulacion(RX_OFDM);
        y_UFMC = UFMC.Demodulacion(RX_UFMC);
        % Se obtienen los bits recibidos
        DataRX_FBMC2
qamdemod(y_FBMC2,M_QAM,'UnitAveragePower',true,'OutputType','bit'); =
        DataRX_FBMC3
qamdemod(y_FBMC3,M_QAM,'UnitAveragePower',true,'OutputType','bit'); =
        DataRX_FBMC4
qamdemod(y_FBMC4,M_QAM,'UnitAveragePower',true,'OutputType','bit'); =
        DataRX_OFDM
qamdemod(y_OFDM,M_QAM,'UnitAveragePower',true,'OutputType','bit'); =
        DataRX_UFMC
qamdemod(y_UFMC,M_QAM,'UnitAveragePower',true,'OutputType','bit'); =
        % Se calcula el BER
        [~,BER_FBMC2(i_SNR,i_rep)] = biterr(Data_FBMC,DataRX_FBMC2);
        [~,BER_FBMC3(i_SNR,i_rep)] = biterr(Data_FBMC,DataRX_FBMC3);
        [~,BER_FBMC4(i_SNR,i_rep)] = biterr(Data_FBMC,DataRX_FBMC4);
        [~,BER_OFDM(i_SNR,i_rep)] = biterr(Data_OFDM,DataRX_OFDM);
        [~,BER_UFMC(i_SNR,i_rep)] = biterr(Data_UFMC,DataRX_UFMC);
    end
    Tiempo_Sim = toc; % Se almacena el tiempo actual del cronómetro
    if (mod(i_rep,25)==0) || (i_rep==5)
        Tiempo_Rest = Tiempo_Sim/i_rep*(No_Repeticion-i_rep); % Se realiza
una estimación del tiempo restante para finalizar la simulación

```

```

        Hora = floor(Tiempo_Rest/3600);
        Min = floor(rem(Tiempo_Rest,3600)/60);
        Seg = (rem(rem(Tiempo_Rest,3600),60));
        fprintf('%.1f%% Completado. ',i_rep/No_Repeticion*100);
        if i_rep<No_Repeticion
            fprintf('Tiempo restante aprox.: %d h %d min %.3f
s.\n',Hora,Min,Seg);
        end
    end
end
Hora = floor(Tiempo_Sim/3600);
Min = floor(rem(Tiempo_Sim,3600)/60);
Seg = (rem(rem(Tiempo_Sim,3600),60));
if mod(i_rep,25)==0
    fprintf('Tiempo total de simulación: %d h %d min %.3f
s.\n',Hora,Min,Seg);
else
    fprintf('%.1f%% Completado. Tiempo total de simulación: %d h %d min
%.3f s.\n',i_rep/No_Repeticion*100,Hora,Min,Seg);
end
ColorFBMC2 = [1 0 1]*0.6;
ColorFBMC3 = [0 1 1]*0.6;
ColorFBMC4 = [0 0 1]*0.6;
ColorOFDM = [1 0 0]*0.7;
ColorUFMC = [0 1 0]*0.8;
% Se grafica BER vs SNR
figure;
semilogy(SNR_dB,mean(BER_FBMC2,2),'+-','color',ColorFBMC2); hold on;
semilogy(SNR_dB,mean(BER_FBMC3,2),'x-','color',ColorFBMC3);
semilogy(SNR_dB,mean(BER_FBMC4,2),'*-','color',ColorFBMC4);
semilogy(SNR_dB,mean(BER_OFDM,2),'o-','color',ColorOFDM);
semilogy(SNR_dB,mean(BER_UFMC,2),'^-','color',ColorUFMC);
ylabel('Bit Error Rate');
xlabel('Signal to Noise Ratio [dB]');
title('BER vs SNR')
grid on;
% BER Teórico
EbNo_dB=SNR_dB-10*log10(log2(M_QAM)*(NoSubP_Datos/OFDM.Imp.NFFT)); % Se
obtiene los valores de Eb/No.
BER_Teorico = berawgn(EbNo_dB,'qam',M_QAM); % Se obtiene el BER Teórico de
QAM a través de un canal AWGN
semilogy(SNR_dB,BER_Teorico,'black')
xlim([min(SNR_dB) max(SNR_dB)]); % Límites en el eje X
ylim([10^-10 1]); % Límites en el eje Y
legend({'FBMC, K = 2','FBMC, K = 3','FBMC, K =
4','OFDM','UFMC','Teórico'},'Location','SouthWest');

%% Información
fprintf('=====\n');
fprintf('===== Información Básica (sin intervalos de guarda)
=====\n');
fprintf('
| TF spacing | AB utilizado | Tiempo | Bit
Rate |\n');
fprintf('OFDM (con CP) | %10.2f | %8.2f MHz | %7.2f ms | %7.2f Mbps
|\n',OFDM.PHY.TiempoSimb*OFDM.PHY.DeltaF,OFDM.PHY.DeltaF*OFDM.No.SupportD
atos/1e6,...
OFDM.PHY.TiempoSimb*OFDM.No.Simbolos/1e-
3,length(DataRX_OFDM)/(OFDM.PHY.TiempoSimb*OFDM.No.Simbolos)/1e6);

```

```

fprintf('FBMC-OQAM, K=2 | %10.2f | %8.2f MHz | %7.2f ms | %7.2f Mbps
|\n', FBMC2.PHY.TiempoSimb*FBMC2.PHY.DeltaF*2, FBMC2.PHY.DeltaF*FBMC2.No.Su
bportDatos/1e6, ...
    FBMC2.No.TotalMuestras*FBMC2.PHY.dt/1e-
3, length(DataRX_FBMC2)/(FBMC2.No.TotalMuestras*FBMC2.PHY.dt)/1e6);
fprintf('FBMC-OQAM, K=3 | %10.2f | %8.2f MHz | %7.2f ms | %7.2f Mbps
|\n', FBMC3.PHY.TiempoSimb*FBMC3.PHY.DeltaF*2, FBMC3.PHY.DeltaF*FBMC3.No.Su
bportDatos/1e6, ...
    FBMC3.No.TotalMuestras*FBMC3.PHY.dt/1e-
3, length(DataRX_FBMC3)/(FBMC3.No.TotalMuestras*FBMC3.PHY.dt)/1e6);
fprintf('FBMC-OQAM, K=4 | %10.2f | %8.2f MHz | %7.2f ms | %7.2f Mbps
|\n', FBMC4.PHY.TiempoSimb*FBMC4.PHY.DeltaF*2, FBMC4.PHY.DeltaF*FBMC4.No.Su
bportDatos/1e6, ...
    FBMC4.No.TotalMuestras*FBMC4.PHY.dt/1e-
3, length(DataRX_FBMC4)/(FBMC4.No.TotalMuestras*FBMC4.PHY.dt)/1e6);
fprintf('UFMC
| %10.2f | %8.2f MHz | %7.2f ms | %7.2f Mbps
|\n', UFMC.PHY.TiempoSimb*UFMC.PHY.DeltaF, UFMC.PHY.DeltaF*UFMC.No.SupportD
atos/1e6, ...
    UFMC.PHY.TiempoSimb*UFMC.No.Simbolos/1e-
3, length(DataRX_UFMC)/(UFMC.PHY.TiempoSimb*UFMC.No.Simbolos)/1e6);
fprintf('=====
=====|\n');
fprintf('=====
=====|\n');

```

ORDEN DE EMPASTADO