



1859

UNIVERSIDAD NACIONAL DE LOJA
FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES CARRERA DE INGENIERÍA EN
ELECTRÓNICA Y TELECOMUNICACIONES

**Evaluación del Desempeño de Nuevos Esquemas de Modulación y Acceso Múltiple
para 5G**

Tesis Presentada Para Obtener el Título de
Ingeniero en Electrónica y
Telecomunicaciones

AUTOR:

Fabián Alejandro Carrión Morillo.

DIRECTOR:

Ing. Juan Gabriel Ochoa Aldeán Mg. Sc.

Loja – Ecuador

2019

CERTIFICACIÓN

Ing. Juan Gabriel Ochoa Aldeán Mg. Sc.

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en “Evaluación del Desempeño de Nuevos Esquemas de Modulación y Acceso Múltiple para 5G” previa a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones, realizado por el señor egresado: Fabián Alejandro Carrión Morillo, la misma que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, abril de 2019.


A rectangular box containing a handwritten signature in blue ink. The signature is stylized and appears to be 'J. G. Ochoa Aldeán'.

Ing. Juan Gabriel Ochoa Aldeán Mg. Sc.

DIRECTOR DE TESIS

AUTORÍA

Yo **FABIÁN ALEJANDRO CARRIÓN MORILLO** declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en Repositorio Institucional – Biblioteca virtual.



Cédula: 1105109068

Fecha: 15 de mayo de 2019

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo “**FABIÁN ALEJANDRO CARRIÓN MORILLO**”, declaro ser autor de la tesis titulada: “**EVALUACIÓN DEL DESEMPEÑO DE NUEVOS ESQUEMAS DE MODULACIÓN Y ACCESO MULTIPLE PARA 5G**”, como requisito para optar al grado de: **INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el repositorio digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, 19 de agosto de dos mil diecinueve.

Firma:



Autor: Fabián Alejandro Carrión Morillo

Cédula: 1105109068

Dirección: Bolívar 219-25 entre Cariamanga y Céllica

Correo Electrónico: facarrionm@unl.edu.ec

Teléfono: 2571119 **Celular:** 593-0969795198

DATOS COMPLEMENTARIOS

Director de tesis: Ing. Juan Gabriel Ochoa Aldeán Mg. Sc.

Tribunal de grado: Ing. Andy Fabricio Vega León Mg. Sc.,

Ing. Marianela del Cise Carrión Gonzales Mg. Sc.,

Ing. Santiago Abraham Medina León Mg. Sc.

DEDICATORIA

Este trabajo de titulación está dedicado a mi Madre Elena Morillo, quien ha sido el pilar fundamental para el desarrollo y cumplimiento de esta meta.

AGRADECIMIENTO

Agradezco a Dios, a mi madre, mi padre, mi familia, a la Universidad Nacional de Loja y a mis docentes, por brindarme su ayuda y conocimientos que han sido la asesoría necesaria para cumplir con la meta de ser Ingeniero en Electrónica y Telecomunicaciones.

ÍNDICE DE CONTENIDO

DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
ÍNDICE DE FIGURAS.....	IX
ÍNDICE DE TABLAS.....	XI
1. TÍTULO.....	1
2. RESUMEN.....	2
2.1. SUMMARY	3
3. INTRODUCCIÓN.....	5
4. REVISIÓN DE LITERATURA	7
4.1. PROCESO PARA DEFINIR 5G	7
4.1.1. Escenarios futuristas y cumplimiento 5G.....	7
4.1.2. Requisitos para 5G.....	8
4.2. ESQUEMAS DE MODULACIÓN PROPUESTOS PARA 5G	10
4.2.1. OFDM	10
4.2.2. OFDM FILTRADO.....	16
4.2.3. Filter Bank Multi Carrier (FBMC).....	22
4.2.4. Universal-Filtered Multi-Carrier (UFMC).....	26
4.2.5. WOLA (Weighted Overlap y Add basado en OFDM)	30
4.3. ACCESO MÚLTIPLE	36
4.3.1. OFDMA	36
4.3.2. SC-FDMA.....	38
4.3.3. NOMA (Acceso múltiple no ortogonal)	39
5. MATERIALES Y MÉTODOS.....	46
6. RESULTADOS.....	52
6.1. Densidad Espectral de Potencia.....	52
6.1.1. FBMC.....	53
6.1.2. OFDM.....	54
6.1.3. WOLA.....	55
6.1.4. f-OFDM.....	56
6.1.5. UFMC	57
6.2. Consumo Energético	58
6.2.1. FBMC.....	59
6.2.2. OFDM.....	60
6.2.3. WOLA.....	61

6.2.4. f-OFDM.....	62
6.2.5. UFMC	63
6.3. Retardo de transmisión.....	64
6.3.1. Relación de error de bit.....	64
6.4. NOMA vs OFDMA.....	66
6.5. TABLA COMPRATIVA DE ESQUEMAS DE MODULACIÓN.....	69
6.6. TABLA COMPARTIVA NOMA vs OFDMA.....	70
7. DISCUSIÓN	71
8. CONCLUSIONES.....	73
9. RECOMENDACIONES	75
10. BIBLIOGRAFÍA.....	76
11. ANEXOS.....	2
11.1. CODIGO DE PROGRAAMA GUIDE UI_5G.....	2

ÍNDICE DE FIGURAS

Figura 1: Portadora única vs múltiples portadoras (OFDM)	11
Figura 2: <i>Diseño de propuesta para 5G NR de forma de onda y acceso múltiple</i>	15
Figura 3: Esquema de transmisión F-OFDM	17
Figura 4: Esquema de recepción f-OFDM	17
Figura 5: Flexibilidad y coexistencia de diferentes tipos de ondas en f-OFDM	18
Figura 6: Respuesta al impulso de diferentes filtros menos de 1000 muestras	21
Figura 7: Respuesta al impulso de diferentes filtros más de 1000 muestras	21
Figura 8: Diagrama de bloques de OFDM	23
Figura 9: Diagrama de bloques de FBMC	23
Figura 10: Espectro de OFDM	24
Figura 11: Espectro de FBMC	24
Figura 12: Diagrama de UFMC transmisión	27
Figura 13: Diagrama UFMC recepción	28
Figura 14: Espectro superpuesto de 6 diferentes RB UFMC con 12 subportadoras cada uno que llevan símbolos QPSK	29
Figura 15: Comparación de un RB de OFDM con un RB de UFMC	30
Figura 16: Transmisor WOLA-OFDM	32
Figura 17: WOLA-OFDM Receptor	33
Figura 18: Coexistencia de 2 usuarios WOLA	34
Figura 19: Densidad espectral de potencia de los 3 esquemas	36
Figura 20: Comparación esquema de transmisión OFDMA vs SC-FDMA	39
Figura 21: Downlink en NOMA en una celda con una estación base y dos usuarios ...	40
Figura 22: Comparación del rendimiento BER de SCMA, MUSA y PDMA en canales de desvanecimiento de Rayleigh.	43
Figura 23: Versión utilizada de MATLAB	48
Figura 24: Entorno de simulación	51
Figura 25: Densidad Espectral de Potencia FBMC	53
Figura 26: Densidad Espectral de Potencia	54
Figura 27: Densidad Espectral de Potencia WOLA	55
Figura 28: Densidad Espectral de Potencia f-OFDM	56
Figura 29: Densidad Espectral de Potencia UFMC	57

Figura 30: Consumo de energía FBMC	59
Figura 31: Consumo de energía OFDM	60
Figura 32: Consumo de Potencia WOLA	61
Figura 33: Consumo de Potencia f-OFDM	62
Figura 34: Consumo de Potencia UFMC	63
Figura 35: Relación de error de bit	64
Figura 36: Acercamiento relación de error de bits	65
Figura 37: Taza de bits por usuario	67
Figura 38: Eficiencia NOMA vs OFDMA.....	68

ÍNDICE DE TABLAS

Tabla 1: Requerimientos mínimos para 5G.....	8
Tabla 2: Parametros generales, configuración LTE	34
Tabla 3: Parámetros de OFDMA.....	38
Tabla 4: PARAMETROS DE SIMULACIÓN.....	50
Tabla 5: Comparativa de Modulación.....	69
Tabla 6: Comparativa de acceso múltiple	70

1. TÍTULO

“EVALUACIÓN DEL DESEMPEÑO DE NUEVOS ESQUEMAS DE MODULACIÓN
Y ACCESO MÚLTIPLE PARA 5G”

2. RESUMEN

El uso efectivo del espectro electromagnético disponible nos brinda la posibilidad de una comunicación de alta fidelidad. 5G es la próxima generación de comunicaciones móviles y por lo tanto se han propuesto varios esquemas de modulación y acceso múltiple para cumplir con este propósito.

OFDM es la base de la cuarta generación de comunicaciones móviles llamada *LTE-Advanced* en sus variantes de modulación y acceso múltiple OFDMA y SC-OFDM, por lo tanto, el esquema de modulación que sucede debe ser compatible con estas tecnologías. OFDM ha cumplido con los requisitos que la norma se propuso, pero hay otras tecnologías que se proponen para llevar a cabo la transmisión de información en 5G, esquemas basados en filtros y con compatibilidad con tecnologías de antenas múltiples y con el uso de ondas milimétricas.

Los esquemas más opcionados al comienzo de esta investigación fueron FBMC, UFMC, F-OFDM y OFDM que con el uso de ventanas se llama WOLA y debido al uso del prefijo cíclico se llama CP-OFDM.

Todos los esquemas basados en filtros se ven afectados por la complejidad del uso de filtros y por una mayor carga de procesamiento en los dispositivos, el comportamiento de estos esquemas se describe en el desarrollo de la investigación y se comparará la eficiencia en el espectro, la eficiencia energética y el retraso, de esta manera poder evaluar cada esquema en un software de simulación y obtener una mejor idea de cuál será el esquema de modulación y acceso múltiple que se utilizará en 5G.

En el campo del acceso múltiple, el acceso múltiple ortogonal se ha utilizado hasta ahora y la tecnología que propone algunas industrias es NOMA porque presenta muchas características diferentes y aparentemente mejores a las de OFDMA, ya que podrían

ofrecer un mejor uso del espectro electromagnético, pero el costo de procesamiento es mayor. Este comportamiento también se evalúa en el software de simulación.

Finalmente, este documento presenta los resultados de gráficos que se compararon con el comportamiento de cada esquema.

2.1. SUMMARY

The effective use of the electromagnetic spectrum gives us the possibility of high fidelity communication. 5G is the next generation of mobile communications and therefore several modulation and multiple access schemes have been improved to fulfill this purpose.

OFDM is the base of the fourth generation of mobile communications called LTE-Advanced in its variants of modulation and multiple access OFDMA and SC-OFDM, therefore, the modulation scheme that happens must be compatible with these technologies. OFDM has fulfilled the requirements of the standard itself, but there are other technologies that are proposed to carry out a transmission of information in 5G, schemes in the filters and compatibility with multiple antenna technologies and with the use of millimeter waves.

The most optional schemes at the beginning of this investigation were FBMC, UFMC, F-OFDM and OFDM that with the use of windows is called WOLA and due to the use of the cyclic prefix it is called CP-OFDM.

All the schemes are based on the filters shown in the results are described in the development of the research and it is understood the spectrum efficiency, energy efficiency and delay, in this way you can use a simulation scheme and obtain a better idea that it is a modulation and multiple access scheme that is used in 5G.

In the field of multiple access, orthogonal multiple access has been used so far and the technology proposed by some industries is NOMA because it presents many different characteristics and apparently better than those of OFDMA, it offers a better use of the electromagnetic spectrum, but the cost processing is greater. This behavior is also evaluated in the simulation software.

Finally, this document presents the results of the graphs that are compared with the behavior of each scheme.

3. INTRODUCCIÓN

Los sistemas de comunicaciones móviles siempre se encuentran en continua evolución debido a las exigencias que presentan los usuarios a la hora de hacer uso de esta tecnología, por tal motivo antes del posible lanzamiento de 5G existen tecnologías que han abierto paso al nuevo sistema de comunicaciones móviles.

La necesidad de tecnologías que brinden más comodidades a los usuarios han hecho que se construyan complejos sistemas de comunicación que décadas atrás solamente eran ciencia ficción, la sociedad de la información en la que ahora estamos inmersos ha sido fruto de un progreso constante, con el tiempo el uso de tecnologías inalámbricas ha dado paso a las comunicaciones móviles que actualmente cursa su cuarta generación denominada *Long Term Evolution* (LTE), la cual nos ofrece velocidades de hasta 100Mbps en ambientes con alta movilidad.

Este progreso constante convergerá en la quinta generación de comunicaciones móviles que será capaz de atender a una mayor cantidad de dispositivos con servicios de mejor calidad. En 5G se pretende densificar la red para ello es imprescindible mejorar el uso del espectro radioeléctrico de tal forma que se aproveche todo el ancho de banda disponible.

Hasta el momento el uso de modulaciones ortogonales nos ha permitido una aceptable canalización del espectro, el acceso múltiple OFDMA es la utilizada en 4G y ha cumplido con las expectativas propuestas por el estándar *IMT-Advanced*, la futura generación de comunicaciones móviles pretende hacer un mejor uso del espectro electromagnético de tal manera que se logren cumplir los objetivos propuestos antes de su lanzamiento, por tal motivo se han propuesto nuevos esquemas de modulación y acceso múltiple para atender los requerimientos de transmisión de 5G, modulaciones basadas en filtros y tecnologías no ortogonales para acceso al medio son las que se perfilan como candidatas para poder portar la información desde el transmisor hasta el dispositivo móvil, la

utilización de frecuencias altas (ondas milimétricas) así como de tecnologías *Multiple Input Multiple Output massive* (MIMO massive).

Para establecer conclusiones claras del perfil más apropiado en cuanto a modulación y acceso múltiple para 5G, será necesario el sustento teórico de investigaciones previas planteado en el primer objetivo, más la evaluación en el simulador planteado en el segundo objetivo, comparar las nuevas tecnologías de acceso al medio permitirá entender cuál será el perfil más apropiado de esquema de modulación y acceso múltiple para la quinta generación de comunicaciones móviles. La comparación de los esquemas planteado en el tercer objetivo establecerá la posibilidad de que este documento de investigación se convierta en un marco de referencia sobre modulación y acceso múltiple.

Los esquemas que al inicio de esta investigación fueron los más perfilados son: OFDM, *Filtered Bank MultiCarrier* (FBMC) OFDM filtrado, *Universal filtered multicarrier* (UFMC) y *Weighted Overlap and Add based OFDM* (WOLA) [OFDM enventanado]. En cuanto a acceso múltiple OFDMA será comparado como las nuevas tecnologías de acceso al medio no ortogonal.

Esta evaluación se presenta en los resultados de las simulaciones donde se analizan graficas de parámetros esenciales como: la densidad espectral, la eficiencia energética y retardo de transmisión, estos resultados nos brindan la información necesaria para distinguir cuál es el esquema que presenta un mejor funcionamiento para determinados escenarios, de esta forma poder prever cuál es el candidato óptimo para la implementación del nuevo sistema de comunicaciones inalámbricas.

4. REVISIÓN DE LITERATURA

4.1. PROCESO PARA DEFINIR 5G

El Grupo de Trabajo (WP) 5D del UIT-R es responsable de los aspectos generales de radio de los sistemas internacionales de telecomunicaciones móviles (IMT). Actualmente están en funcionamiento IMT-2000 (3G) e IMT-Advanced (4G). WP5D es el principal responsable dentro del UIT-R por cuestiones relacionadas con el componente terrestre de las IMT, incluidas cuestiones técnicas, operativas y relacionadas con el espectro. Para este propósito el grupo desarrolla y mantiene recomendaciones incluidas las especificaciones de interfaz de radio para los sistemas IMT-2000 en la Recomendación UIT-R M.1457 y para IMT-Advanced en la Recomendación UIT-R M.2012. WP5D ahora está trabajando en IMT 2020, que corresponde a 5G.

Esta recomendación se completará con dos informes técnicos. Un informe sobre tendencias tecnológicas que describirá los aspectos técnicos de los sistemas IMT terrestres durante 2015-2020 y el futuro, incluida la evolución de las IMT. Un segundo informe describirá la viabilidad técnica de las IMT en las bandas por encima de 6 GHz. Este trabajo inicial dentro del UIT-R se relaciona bien con las actividades de investigación exploratorias globales en curso sobre 5G en *Mobile and wireless communications Enablers for Twenty-twenty (2020) Information Society (METIS)* y otros proyectos. (Andrews, J. G., Buzzi, S., 2014)

4.1.1. Escenarios futuristas y cumplimiento 5G.

La sociedad del 2020 será una sociedad conectada. El IoT (internet of things), junto con los sistemas de sensores inteligentes e integrados y las redes de sensores en el hogar, cambiará la forma en que las personas llevan sus vidas. La gente de "vida inteligente" requerirá una conectividad móvil constante y ubicua a la red para cargar sus datos de actividad y los comandos de control de IoT, generando así un flujo de datos de enlace ascendente de "informes masivos". La comunicación masiva de máquina a

máquina y la comunicación crítica de máquina a máquina jugarán un papel fundamental en la prestación de servicios y las operaciones de la industria. (Mitra, Agrawal, 2015).

Las redes ad-hoc de vehículos (VANET) avanzan constantemente. Para 2020, las redes VANET integradas con redes celulares estarán operativas como la nube VANET, lo que llevará a un sistema de transporte más inteligente y seguro. Cuando la cantidad de dispositivos conectados a Internet supere decenas o cientos de miles de millones en la próxima década, la descarga de datos en red en bandas sin licencia jugará un papel fundamental en el balanceo de carga de la red, proporcionando servicios de velocidad de bits garantizados y una reducción en la señalización de control. Por lo tanto, es importante que 5G proporcione una compatibilidad perfecta con redes densas y heterogéneas para satisfacer la alta demanda de tráfico en tiempo real, de modo que los usuarios finales experimenten una conectividad sin problemas a la red. (Mitra, Agrawal, 2015).

4.1.2. Requisitos para 5G

Los requisitos para la RAN están especificados en la norma TR 38.913 y se definen en los siguientes términos:

- *Key Performance Indicators* (KPIs) grupo de indicadores clave de rendimiento y objetivos.
- Escenarios de implementación: eMBB, mMTC y URLLC.
- Otros requisitos como: arquitectura, servicios y operación.

Tabla 1: Requerimientos mínimos para 5G

Fuente: Eiman Mohyeldin, 2016

N°	KPI	Objetivo	Aplicación		
			eMBB	mMTC	URLLC
1	Velocidad de datos	DL 20 Gbps UL 10 Gbps	X		
2	Eficiencia Espectral	DL 30 bps/Hz UL 15 bps/Hz	X		
3	Velocidad de datos (usuario)	DL 100 Mbps UL 50 Mbps	X		
4	Trafico por área	10Mbps/m ²	X		
5	Latencia plano usuario	URLLC: 1ms eMBB: 4ms	X		X
6	Latencia en plano de control	20ms (10ms recomendados)	X		X
7	Densidad de conexión	1000000/Km ²		X	
8	Eficiencia energética	cualitativo	X		
9	Duración batería UE	10 años		X	
10	Cobertura	MCL=164dB para 160bps		X	
11	Movilidad	Hasta 500 km/h	X		X
12	Tiempo de interrupción en movilidad	0 ms	X		X
13	Ancho de banda	Mayor a 100 MHz Menor a 1 GHz en Ondas milimétricas	General		

La velocidad de datos deberá superar los 10Gbps y la velocidad en el borde de celda (el 95% de usuarios) será en orden de los 100Mbps, de esta manera el acceso inalámbrico tendrá las mismas velocidades manejadas por la fibra óptica. (Eiman Mohyeldin, 2016)

4.2. ESQUEMAS DE MODULACIÓN PROPUESTOS PARA 5G

4.2.1. OFDM

OFDM (Orthogonal Frequency Division Multiplexing) es el esquema de multiplexación base que utiliza 4G para proporcionar acceso al medio.

La multiplexación por división de frecuencia ortogonal (OFDM) en lugar de enviar la información como una transmisión única, divide la información en varias sub-corrientes paralelas y envía cada sub-flujo a una diferente frecuencia conocida como una subportadora.

Muchos nuevos sistemas de comunicación han seleccionado OFDM debido a sus buenas propiedades, como por ejemplo la tolerancia a la interferencia entre símbolos (ISI) y buena eficiencia espectral. Aunque la idea de OFDM se desarrolló en los años 60, el mayor impulso para OFDM fue la reducción de los precios de circuitos integrados y la posibilidad de utilizar la transformada rápida de Fourier. (Farhang, B., & Moradi, H., 2016).

La multiplexación por división de frecuencia ortogonal (OFDM) es una técnica basada en la modulación multiportadora (MCM) y multiplexación por división de frecuencia (FDM). OFDM puede ser considerado como un método de modulación o multiplexación. La idea básica detrás de la modulación multiportadora es dividir el ancho de banda de la

señal en subportadoras paralelas o bandas estrechas del ancho de banda total. (Srikanth, S., Pandian, P., 2012).

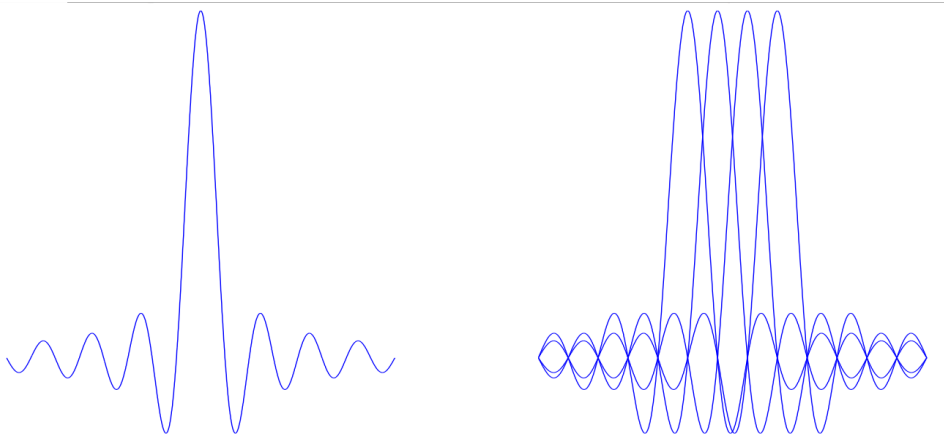


Figura 1: Portadora única vs múltiples portadoras (OFDM)

Fuente: Srikanth, S., Pandian, P., 2012

4.2.1.1. Creación del símbolo OFDM

El símbolo de OFDM es creado en el dominio digital antes de la transmisión, se le aplica modulaciones digitales comunes tales como BPSK, 16QAM, (5G pretende utilizar modulaciones de hasta 256QAM). Este flujo de datos es dividido en N flujos paralelos los cuales se convierten en un símbolo OFDM, así un símbolo OFDM se conforma de varias muestras de diferentes flujos de información.

$$X_k = \sum_{n=0}^{N-1} a_n e^{\frac{j2\pi kn}{N}} \quad (1)$$

Donde a_n es el símbolo de datos en la n-esima sub portadora, la ecuación es equivalente al n-esimo punto de la transformada inversa discreta de Fourier (IDTF). El espaciado de las subportadoras y la frecuencia son escogidas de tal manera que se cumpla a cabalidad el criterio de ortogonalidad.

La ortogonalidad significa, por definición, que el valor promedio en el tiempo T de la multiplicación de dos señales es cero. Este criterio atiende a la ecuación:

$$\frac{1}{T} = \int_T x(t)y(t)dt = 0 \quad (2)$$

Esto significa que las señales no están correlacionadas entre sí, es decir son señales diferentes e independientes. En OFDM los pulsos en forma de Sinc se usan como espectros de subportadora (como se puede ver en la figura 1). Conforme las propiedades de los pulsos sinc, los cruces por cero se encuentran en múltiplos de $1 / T$, con el uso de una frecuencia central de subportadora f y la utilización de pulsos sinc la ortogonalidad de cada subportadora se mantiene.

$$fi = fc + \frac{i}{T} \quad i = -\frac{N}{2} \dots \frac{N}{2} \quad (3)$$

Donde fc es la frecuencia central del canal y N es del número de subportadoras. De esta forma, cada subportadora tiene el máximo en su propia frecuencia central y cero en el centro de la subportadora vecina. Luego de la conversión de serie paralelo se aplica la transformada inversa discreta de Fourier (que en la práctica corresponde a la transformada rápida inversa de Fourier IFFT). Esto equivale a la transición del dominio de la frecuencia al dominio del tiempo. (Farhang, B., & Moradi, H., 2016).

Después de IFFT, todos los datos paralelos se suman y transmiten. Una de las características más importantes en el sistema OFDM es la división del canal de frecuencia selectiva en pequeños subcanales.

La conversión de un flujo de datos en serie rápido en N flujos de datos en paralelo más lentos permite la posibilidad de utilizar periodos de símbolos más largos. Los tiempos de

transmisión más largos permiten una mayor propagación del retardo que las duraciones del símbolo más cortas. Esta propiedad hace que OFDM sea adecuado para entornos difíciles de trayectos múltiples, porque los tiempos de símbolo más largos hacen que OFDM sea robusto frente a ISI. A pesar de que OFDM es muy resistente al ISI, es muy susceptible a los desplazamientos de frecuencia y al ruido de fase. Las variaciones menores en las frecuencias ceden directamente a la pérdida de ortogonalidad. (Farhang, B., & Moradi, H., 2016).

4.2.1.2. Prefijo cíclico

En el dominio de la frecuencia OFDM agrega un Prefijo cíclico, tiempo de guarda en el dominio del tiempo, que a pesar de que introduce los parámetros necesarios para poder reducir la interferencia entre símbolos generan un desperdicio considerable del ancho de banda haciendo que disminuya la eficiencia espectral (parámetro esencial a mejorar para 5G). El prefijo cíclico (CP) es un conjunto de muestras adicionales que se generan con el propósito de eliminar el ISI. El ISI no afecta a las muestras ya generadas por los bits de información ya que esta interferencia es absorbida por las muestras adicionales (prefijo cíclico). (Velázquez G., Vargas R., 2017).

Otra ventaja de usar CP es que induce una convolución circular. Las muestras que se ven afectadas por ISI, el receptor las eliminará. El prefijo cíclico es transmitido junto con la información, esto nos da la ventaja de evitar el ISI, además de tener desvanecimientos solo en algunas subportadoras y no es necesario modelar el pulso.

Las desventajas son la carga computacional del algoritmo FFT, el efecto Doppler en las subportadoras y una alta potencia con amplificadores lineales requeridos.

La señal recibida es la convolución del símbolo OFDM con CP y la respuesta al impulso del canal. Para simplificar la notación, la secuencia de símbolos OFDM $s[k]$ se representa

por s y de forma similar para la respuesta al impulso del canal y la secuencia recibida, entonces la señal recibida en ausencia de ruido se puede describir en términos de una ecuación matricial de la siguiente manera:

$$\mathbf{y} = \mathbf{H}\mathbf{s} \quad (4)$$

La matriz \mathbf{H} es la matriz del canal dado por la respuesta al impulso y los vectores \mathbf{y} , \mathbf{s} son la señal recibida y el símbolo OFDM respectivamente, estos elementos están dados por lo siguiente:

$$\begin{bmatrix} y_{N-1} \\ y_{N-2} \\ \dots \\ y_0 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & \dots & h_{M-1} & h_M & 0 & 0 & \dots & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & \dots & h_{M-2} & h_{M-1} & h_M & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & \dots & h_{M-3} & h_{M-2} & h_{M-1} & h_M & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & h_0 & h_1 & h_2 & h_3 & \dots & h_{M-1} & h_M \end{bmatrix} \begin{bmatrix} s_{N-1} \\ s_{N-2} \\ \dots \\ s_{N-M} \end{bmatrix} \quad (5)$$

La ecuación viene dada por la convolución lineal en tiempo discreto pero la matriz del canal puede ser reorganizado y la ecuación dará una interpretación equivalente mediante el uso de la convolución circular: (Velázquez G., Vargas R., 2017).

$$\begin{bmatrix} y_{N-1} \\ y_{N-2} \\ y_{N-3} \\ \dots \\ y_1 \\ y_0 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & \dots & h_{M-1} & h_M & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & \dots & h_{M-2} & h_{M-1} & h_M & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & \dots & h_{M-3} & h_{M-2} & h_{M-1} & h_M & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_4 & h_5 & h_6 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_2 & h_3 \\ h_3 & h_4 & h_5 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & \dots & h_1 & h_2 \\ h_2 & h_3 & h_4 & \dots & h_M & 0 & 0 & 0 & 0 & 0 & \dots & h_0 & h_1 \\ h_1 & h_2 & h_3 & \dots & h_{M-1} & h_M & 0 & 0 & 0 & 0 & \dots & 0 & h_0 \end{bmatrix} \begin{bmatrix} s_{N-1} \\ s_{N-2} \\ s_{N-3} \\ \dots \\ s_1 \\ s_2 \end{bmatrix} \quad (6)$$

Ahora la matriz tiene la forma $\mathbf{Y}=\mathbf{H}\mathbf{s}$. La matriz \mathbf{H} es la convolución circular de las N muestras de interés. (Velázquez G., Vargas R., 2017).

Si bien las compañías miembros de 3GPP están de acuerdo en que las formas de onda basadas en OFDM y el acceso múltiple se utilizarán en gran medida en 5G NR, la implementación de la forma de onda exacta y el acceso múltiple aún está en discusión.

Se están considerando múltiples variantes de OFDM para los diferentes casos de uso e implementaciones de 5G. El siguiente es un resumen de la propuesta de Qualcomm:

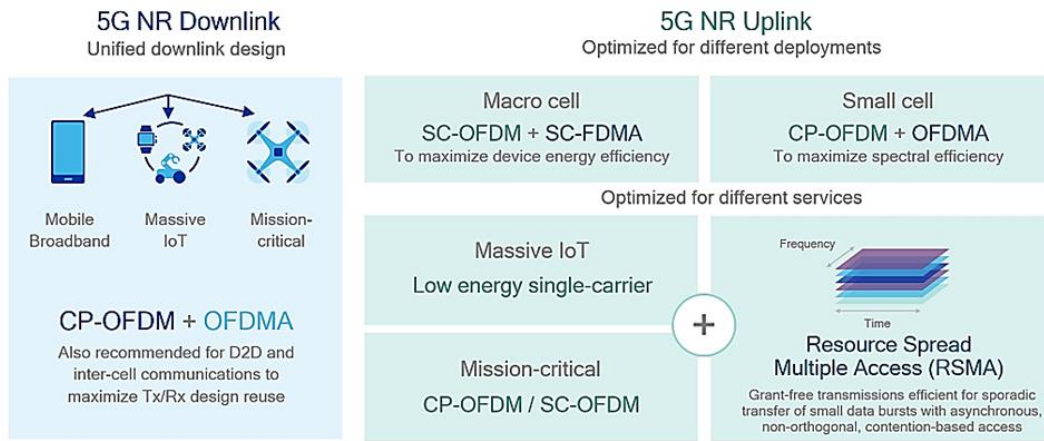


Figura 2: Diseño de propuesta para 5G NR de forma de onda y acceso múltiple

Fuente: Qualcomm Technologies, 2016

Para las transmisiones de enlace descendente, las redes 5G NR pueden beneficiarse de la simplicidad de tener una forma de onda unificada y un diseño de acceso múltiple en todos los servicios e implementaciones. La recomendación de Qualcomm es adoptar CP-OFDM con ventanas, que se utiliza mucho en los sistemas LTE de hoy en día, en combinación con el acceso múltiple OFDMA. Este enfoque establecido cumple con los requisitos de 5G para alta eficiencia espectral, baja complejidad y la implementación MIMO eficiente, al tiempo que brinda una solución para la localización de frecuencia OFDM. Los enfoques alternativos basados en OFDM agregan complejidad en la forma de diseño de radio y multiplexación, con beneficios marginales en implementaciones realistas. El mismo diseño también se recomienda para las comunicaciones entre dispositivos y entre células, lo que permite la máxima reutilización del diseño del transmisor y el receptor. (Qualcomm Technologies, 2016.)

4.2.2. OFDM FILTRADO

Para cumplir con los nuevos desafíos enfrentados por la forma de onda 5G, se presenta un nuevo habilitador para la forma de onda flexible, denominada como OFDM filtrado (f-OFDM).

Con la división y el filtrado basados en subbandas, los sistemas OFDM independientes están estrechamente contenidos en el ancho de banda asignado. De esta forma, f-OFDM es capaz de superar los inconvenientes de OFDM conservando sus ventajas. En primer lugar, con el filtrado basado en subbandas, el requisito de sincronización global se relaja y se puede soportar la transmisión asíncrona entre subbandas. En segundo lugar, con filtros adecuadamente diseñados para suprimir el OOBE, el consumo de la banda de guarda se puede reducir a un nivel mínimo. En tercer lugar, dentro de cada subbanda, la numerología optimizada se puede aplicar para satisfacer las necesidades de cierto tipo de servicios. En general, los nuevos requisitos de rendimiento que enfrenta la forma de onda 5G pueden cumplirse con f-OFDM y la eficiencia del espectro global puede mejorarse. Entre todos los candidatos de forma de onda 5G, f-OFDM aparece como el más prometedor, en términos del rendimiento general, la complejidad asociada, el costo y suavidad en el camino de evolución de 4G LTE. (Zhang, X., Jia, M., 2015)

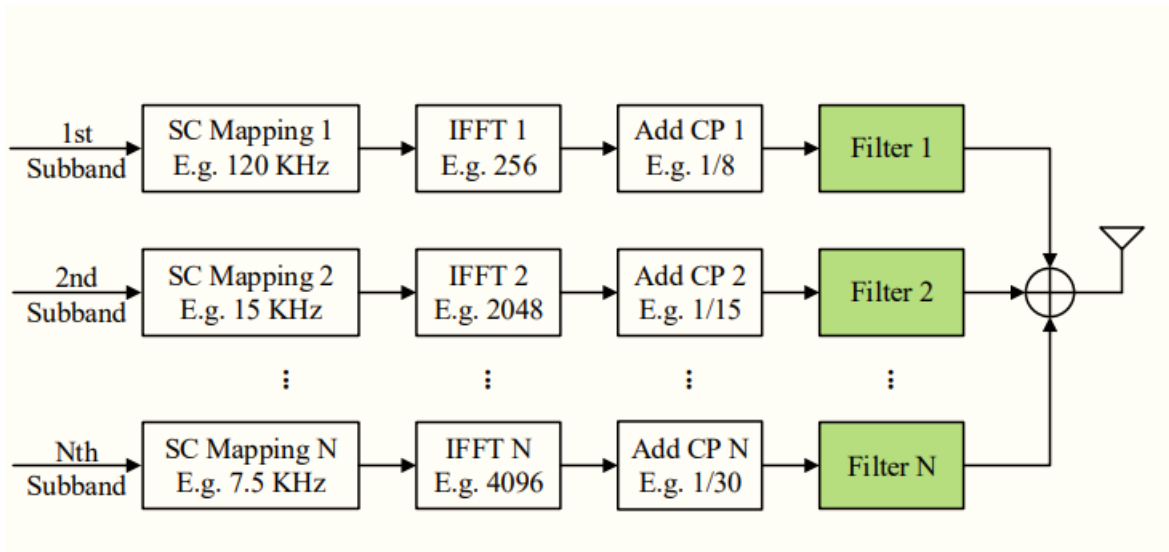


Figura 3: Esquema de transmisión F-OFDM

Fuente: Zhang, X., Jia, M., 2015

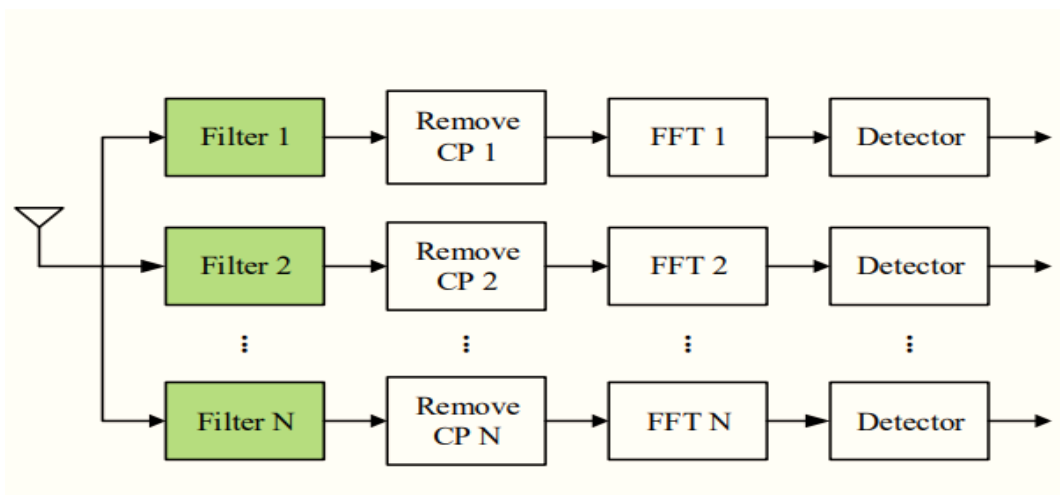


Figura 4: Esquema de recepción f-OFDM

Fuente: Zhang, X., Jia, M., 2015

Con f-OFDM, el ancho de banda asignado se dividirá en varias subbandas. En cada subbanda, una OFDM convencional (y posiblemente otra forma de onda) se adapta para satisfacer las necesidades de cierto tipo de servicio y las características de canal asociadas, por ejemplo, con un espaciado apropiado de subportadora (SC), longitud de prefijo cíclico (CP) y transmisión intervalo de tiempo (TTI), etc. El filtro basado en subbandas se aplica

para suprimir la interferencia entre subbandas, y la ortogonalidad en el dominio de tiempo entre símbolos OFDM consecutivos en cada subbanda se rompe intencionalmente para un OOB inferior con pérdida de rendimiento insignificante en otros aspectos

En consecuencia, la transmisión asíncrona a través de subbandas ahora puede ser compatible y la sincronización global ya no es necesaria, a diferencia de la OFDM convencional. Además, f-OFDM también proporciona reducciones significativas en el consumo de banda de guarda, lo que lleva a una utilización del espectro más eficiente. (Zhang, X., Jia, M., 2015)

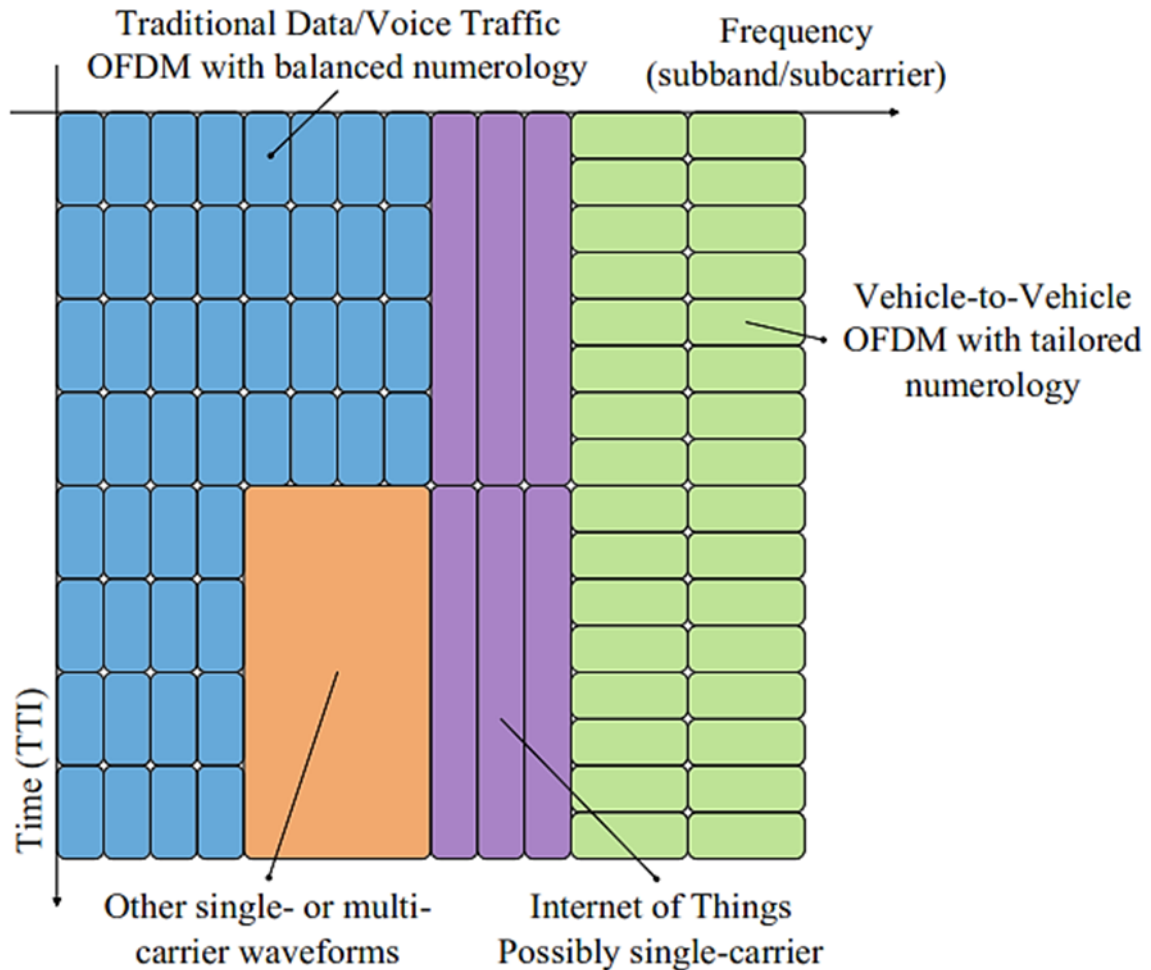


Figura 5: Flexibilidad y coexistencia de diferentes tipos de ondas en f-OFDM

Fuente: Zhang, X., Jia, M., 2015

La flexibilidad y la coexistencia de formas de onda habilitadas por f-OFDM. Como se puede ver en la figura 5, en lugar de una distribución uniforme como la empleada por OFDM en 4G LTE, la distribución / asignación tiempo-frecuencia de f-OFDM es mucho más flexible. Por ejemplo, para proporcionar latencia ultrabaja y alta fiabilidad para la comunicación de vehículo a vehículo, la duración de TTI se acorta mientras que el espaciado de la subportadora de OFDM se amplía, en comparación con la numerología OFDM utilizada en 4G LTE. De manera similar, para permitir una cobertura suficiente con bajo consumo de energía para internet-of-things, se incluye una forma de onda de portadora única adaptada, posiblemente con una ocupación de frecuencia pequeña (para aumentar la densidad de potencia de transmisión y superar la pérdida de penetración), una larga duración de TTI (explotando el canal cuasi estático para la confiabilidad de la transmisión). En general, se pueden incorporar diferentes formas de onda bajo el marco de f-OFDM, y la disposición de tiempo-frecuencia puede cambiar con el tiempo, adaptándose a los requisitos del servicio y a las características del canal del tiempo. (Zhang, X., Jia, M., 2015).

4.2.2.1. Diseño e implementación del filtro

Para habilitar el filtrado basado en subbandas y así disfrutar de los beneficios prometidos por f-OFDM, se necesitan filtros adecuadamente diseñados. En general, el diseño del filtro implica la compensación entre las características del dominio de tiempo y de frecuencia, y también se basa en la complejidad de la implementación. La dispersión de energía en el dominio del tiempo debe contenerse para restringir la interferencia entre símbolos (ISI), y la nitidez de la región de transición en el dominio de frecuencia también vale la pena. En aras de la complejidad, es recomendable implementar los filtros en el dominio de frecuencia utilizando el método de superposición y guardado, con el que se pueden aprovechar los beneficios de la transformada rápida de Fourier (FFT). Además, para lograr una reasignación de subbanda flexible, también es deseable un enfoque sistemático y conveniente que permita la generación en línea de filtros para cualquier requisito de espectro dado. Se consideraron dos tipos de filtros (Zhang, X., Jia, M., 2015).

La respuesta típica de tiempo y frecuencia se representa en las Figs. 6 y 7 (con un orden de banda de paso de 1024 y 720-KHz). Una breve discusión sobre estos filtros se da de la siguiente manera:

1) Filtros Sinc truncado blando: la respuesta de impulso de un filtro de paso bajo ideal es una función que es infinitamente larga. Para la implementación práctica, la función sinc se trunca suavemente con diferentes funciones de ventana: 1. ventana de Hann; 2. Ventana *Root-raised-cosine* (RRC). De esta forma, la respuesta de impulso de los filtros obtenidos se desvanecerá rápidamente (ver figura 6) y, por lo tanto, limitará el ISI introducido entre los símbolos OFDM consecutivos. Si bien es fácil de generar, los filtros sinc truncados blandos probablemente tendrán frecuencias más pequeñas, por ejemplo, aquellos con una magnitud inferior a -30 dB pueden excluirse de la implementación de dominio de frecuencia de punto fijo (ver figura 7), lo que podría conducir a ahorros adicionales de complejidad. (Zhang, X., Jia, M., 2015).

2) Filtros Equiripple: Diseñados utilizando el algoritmo de intercambio Remez se minimiza el error máximo entre la respuesta de frecuencia deseada y real, y así se puede obtener una región de transición más nítida, en comparación con los filtros de truncado blando (ver figura 7), que, como se discutió previamente, es muy deseable para aliviar la interferencia entre subbanda. Sin embargo, con una región de transición extremadamente estrecha, la respuesta de impulso de los filtros compactos muestra discontinuidades en la cabeza y la cola, lo que podría ser un factor limitador del rendimiento cuando la relación señal-ruido operante (SNR) es relativamente alta y el ISI se convierte en el problema dominante, y esto no sucede con los filtros de sincronización de arco blando. Además, el algoritmo de intercambio de Remez requiere una optimización iterativa y, por lo tanto, es inconveniente para la generación de filtros en línea. (Zhang, X., Jia, M., 2015).

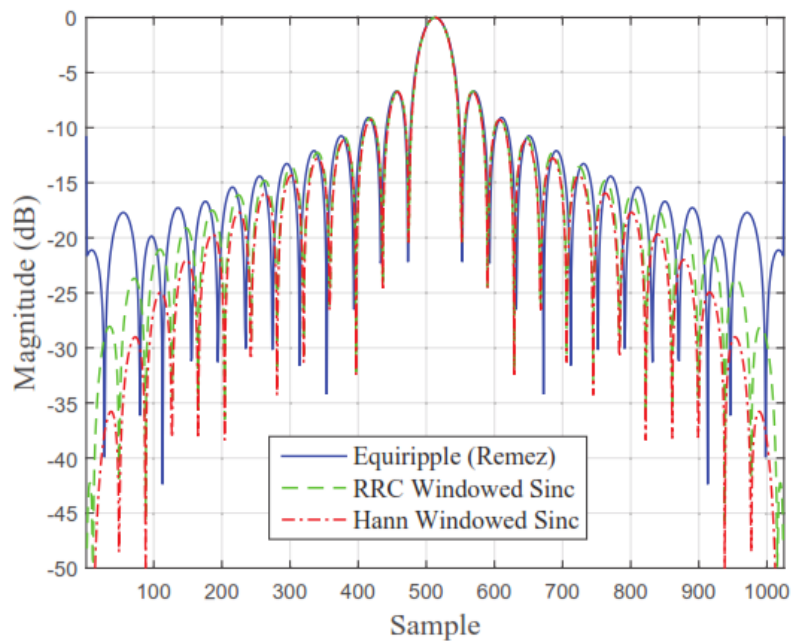


Figura 6: Respuesta al impulso de diferentes filtros menos de 1000 muestras

Fuente: Zhang, X., Jia, M., 2015

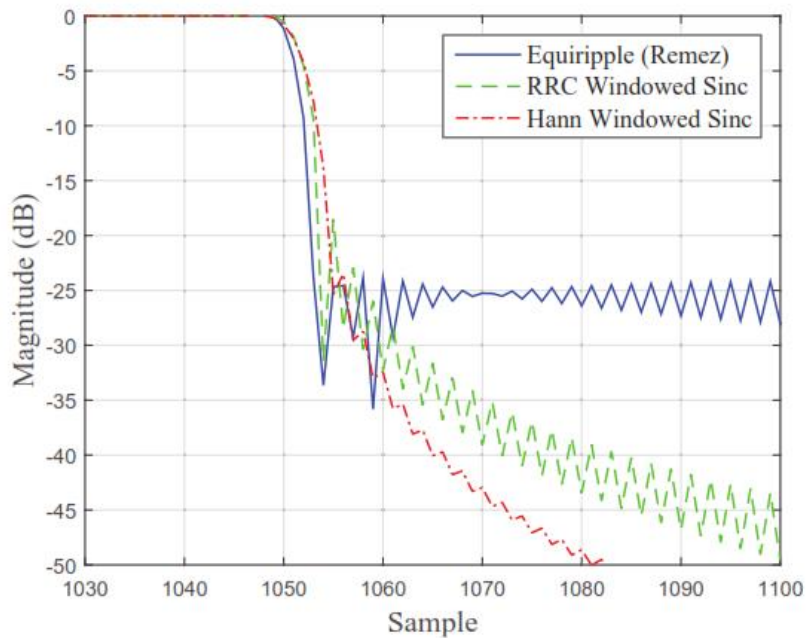


Figura 7: Respuesta al impulso de diferentes filtros más de 1000 muestras

Fuente: Zhang, X., Jia, M., 2015

4.2.3. Filter Bank Multi Carrier (FBMC)

La técnica de FBMC en conjunto con la modulación por amplitud de cuadratura desplazada *offset quadrature amplitud modulation* (OQAM), pueden alcanzar una interferencia entre símbolos y una interferencia entre portadoras menor que OFDM, sin necesidad de utilizar el prefijo cíclico (CP) y, en su lugar, emplear una forma de pulso bien diseñada, que satisfaga las condiciones de reconstrucción.

Los filtros de síntesis y análisis de las subportadoras en OFDM tienen lóbulos laterales relativamente grandes debido a la forma de pulso rectangular de las subportadoras, ocasionando filtraciones de potencia hacia las bandas de otros usuarios, que causan interferencias indeseables. Por consiguiente, en OFDM es necesario extender la duración del símbolo por un tiempo de guarda que exceda la duración de la respuesta al impulso; si bien de esta forma se soluciona el problema de interferencias entre bandas, se reduce el desempeño del sistema por tener que utilizar más ancho de banda. (Sahin, A., Güven, I., & Arslan, H., 2012).

Esta situación puede mejorarse si los filtros que sintetizan y analizan las señales subportadoras tienen lóbulos laterales pequeños; por lo tanto, una opción muy conveniente en aplicaciones de acceso múltiple es la técnica de FBMC, que puede ser diseñada con lóbulos laterales arbitrariamente pequeños y, por consiguiente, puede conservarse la temporización y la duración de los símbolos.

Si bien FBMC requiere un mayor procesamiento que OFDM, este tiene un menor costo que las técnicas de compensación de señal que se necesitan aplicar a un sistema OFDM para mitigar algunos de sus inconvenientes. Los diagramas de bloques de las figuras, muestran la diferencia de implementación entre OFDM y FBMC. (Martínez, E. O., & Agudelo, 2013).

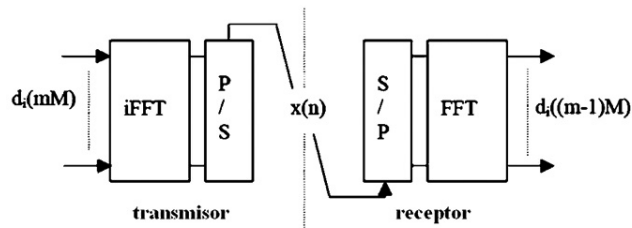


Figura 8: Diagrama de bloques de OFDM

Fuente: Martínez, E. O., & Agudelo, 2013

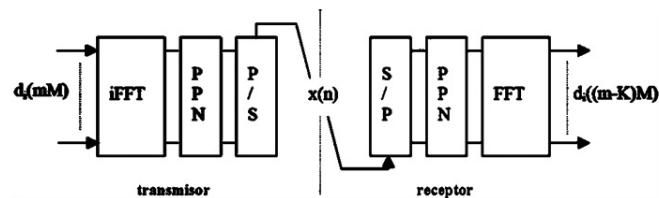


Figura 9: Diagrama de bloques de FBMC

Fuente: Martínez, E. O., & Agudelo, 2013

En principio, la implementación de FBMC es más compleja que OFDM por la adición de los bloques PPN (PolyPhase Network), que implican un mayor procesamiento por los filtros digitales que agrega al sistema. No obstante, se concluye que una implementación basada en FBMC lleva a un desempeño muy superior y con una complejidad computacional menor que OFDM, como consecuencia de utilizar filtros casi perfectos para separar las subportadoras, lo que evita la necesidad de tener que recurrir a alguno de los métodos para la cancelación de interferencia cuando se utiliza OFDM. (Martínez, E. O., & Agudelo, 2013).

En las figuras se puede observar comparativamente las diferencias entre OFDM y FBMC. Como puede verse, FBMC tiene un ancho de banda más confinado, con lóbulos laterales mínimos, a diferencia de OFDM.

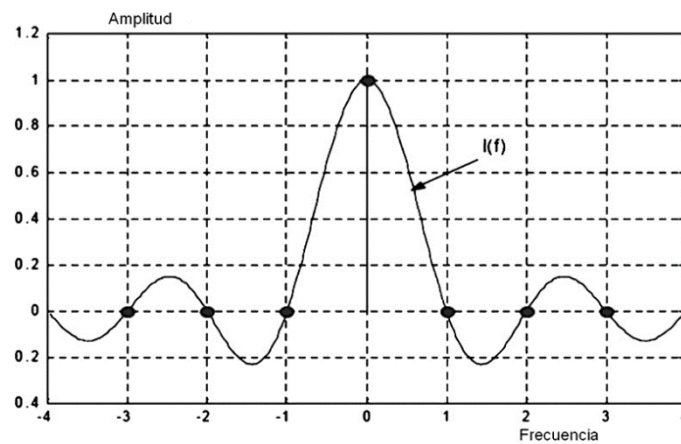


Figura 10: Espectro de OFDM

Fuente: Martínez, E. O., & Agudelo, 2013

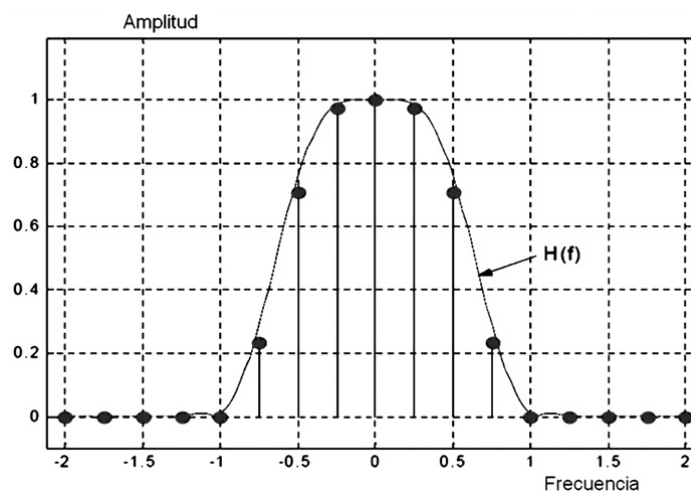


Figura 11: Espectro de FBMC

Fuente: Martínez, E. O., & Agudelo, 2013

Por las características de FBMC mencionadas, al combinarla con modulaciones como OQAM o *Cosine Modulated FilterBank* (CMFB) se logra una velocidad de bits que maximiza la eficiencia de la transmisión, sin necesidad del tiempo de guarda o del prefijo cíclico de OFDM. Adicionalmente, como se expone, la existencia de subcanales independientes o grupos de subcanales con flujos de datos continuos, permite que los

datos recibidos en la BS desde diferentes móviles puedan ser procesados independientemente y de una forma adaptable. (Martínez, E. O., & Agudelo, 2013).

4.2.3.1. El filtro en FBMC

La mayoría de los beneficios de FBMC parten de que las subportadoras no adyacentes están separadas casi perfectamente por medio de un banco de filtros bien diseñados, por lo que el diseño y la construcción de este banco de filtros es un aspecto crucial del sistema.

1) Chang estableció que la condición de ortogonalidad para los esquemas de transmisión de portadora múltiple, considerando filtros de banda limitada y OQAM, es que los subcanales pueden estar separados la mitad de la velocidad del símbolo, sin ICI y sin interferencia entre símbolos (ISI, InterSymbol Interference), cuando los datos están escalonados en subcanales alternados en fase y en cuadratura; 2) Saltzberg generalizó la idea de transmisión de portadora múltiple, demostrando que la condición de Chang es también verdadera cuando se intercambian los ejes de tiempo y frecuencia. (Sahin, A., Güven, I., & Arslan, H., 2012).

Un sistema FBMC puede construirse a partir de un filtro prototipo cuyas dispersiones de tiempo y frecuencia igualen las del canal; así mismo, debe conservarse la máxima eficiencia espectral. Para evitar la ISI, este filtro prototipo debe diseñarse tal que sea un filtro Nyquist y que, con ciertas restricciones en su simetría, también pueda evitar la ICI.

En general, FBMC es flexible en el diseño de los filtros prototipo, lo que permite que estos se diseñen para distintos propósitos.

Los filtros prototipo están clasificados en tres tipos:

Filtros limitados en el tiempo, filtros limitados en la banda y filtros localizados. Los filtros limitados en el tiempo tienen una duración finita, por lo que pueden implementarse con facilidad; además, tienen un espectro de longitud infinita.

Entre ellos, se encuentran los filtros rectangulares, los filtros Hamming, Hanning y Blackman, los filtros de pulsos de duración finita óptima/filtros prolató (que presentan la respuesta de escalón más rápida para un ancho de banda dado), los filtros Kaiser y los filtros PHYDYAS. (Martínez, E. O., & Agudelo, 2013).

Por su parte, los filtros limitados en banda se caracterizan por un ancho de banda finito y una longitud infinita en el dominio del tiempo, lo que los hace problemáticos al considerar la construcción práctica de las ranuras y marcos. Entre estos filtros están el filtro coseno elevado, el filtro coseno raíz elevada y los filtros coseno medio y sinc medio.

Finalmente, los filtros localizados se basan por igual en el dominio del tiempo y de la frecuencia y tienden a comportarse como un pulso gaussiano. Algunos son: Filtro gaussiano, filtro de pulsos Hermite, filtro de algoritmo de transformada ortogonal isotrópica y filtro de función gaussiana extendida.

Comparativamente, con FBMC se puede lograr una interferencia entre símbolos y entre portadoras menor que con OFDM, sin necesidad del CP; igualmente, FBMC permite reducir los lóbulos laterales de cada subportadora, casi eliminando la interferencia por acceso múltiple. Tal vez la principal desventaja de FBMC es el procesamiento adicional; sin embargo, este es menor que el que requiere OFDM para mitigar algunos de los problemas que presenta en un ambiente multiusuario, por ejemplo, en un sistema de comunicaciones inalámbricas multipunto a punto. (Martínez, E. O., & Agudelo, L. B., 2013).

4.2.4. Universal-Filtered Multi-Carrier (UFMC)

Mediante el uso de esta técnica llamada multiportadora universal filtrada (UFMC), el efecto de la interferencia del lóbulo lateral en los subcanales adyacentes inmediatos puede reducirse significativamente. Esto ofrece una mejor robustez de ICI y

una mejor adecuación para el funcionamiento del espectro fragmentado. Además, la técnica UFMC utiliza longitudes de filtro más cortas en comparación con las longitudes de fijación cíclica OFDM, lo que la hace aplicable para la comunicación de ráfagas cortas. La técnica propuesta se puede considerar como un candidato potencial para futuros sistemas inalámbricos (5G) que deben soportar una gran cantidad de dispositivos de bajo costo para integrar el próximo IoT y MMC. (Schaich, F., & Wild, T., 2014)

Este esquema se desarrolla en base al principio de multiplexación por división de frecuencia (FDM) en el que dividimos el flujo de datos de entrada en varios sub-flujos de velocidad más baja. El diagrama de bloques de un sistema que emplea UFMC se representa en la Figura

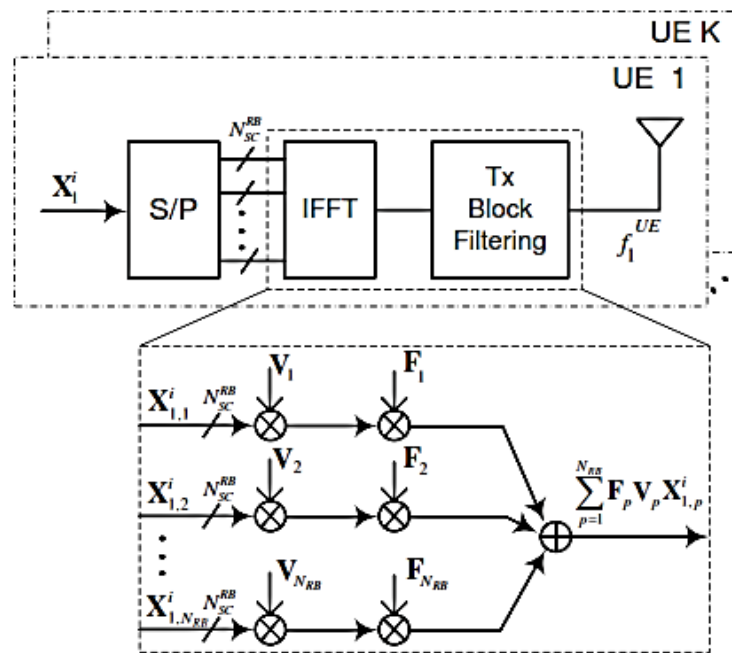


Figura 12: Diagrama de UFMC transmisión

Fuente: Schaich, F., & Wild, T., 2014

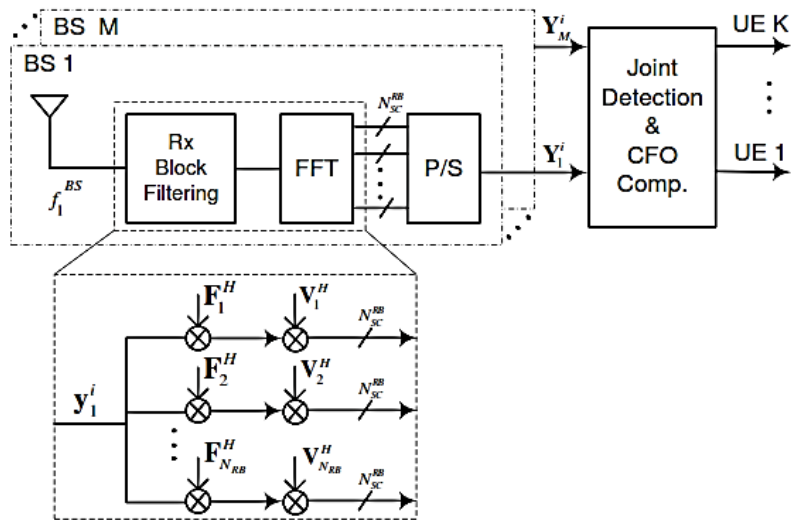


Figura 13: Diagrama UFMC recepción

Fuente: Schaich, F., & Wild, T., 2014

Como se muestra en esta figura, la secuencia de datos del k-ésimo usuario X está dividido dentro N_{RB} sub-flujos denotados por $X_{k,p} \in \mathbb{C}^{N_{RB} N_{SC}}$ para $p \in 1, 2, \dots, N_{RB}$. Entonces, usamos una forma de pulso con bordes lisos en cada bloque de recursos en el dominio del tiempo que conduce a una reducción sustancial en fuga fuera de banda en el dominio de la frecuencia. De esta manera, minimizamos la interferencia perjudicial de los subcanales adyacentes del bloque de recursos vecinos. En la figura, ilustramos el espectro de potencia de las señales UFMC y OFDM en dominio de la frecuencia. En este experimento, usamos un filtro finito de chebyshev de respuesta impulsiva (FIR). La Figura. 20 (a) muestra el espectro UFMC para $N_{RB} = 6$ donde cada RB contiene 12 subportadoras. En la Fig. 20 (b), comparamos el espectro de potencia de un RB de UFMC con ese de la OFDM. Como se puede observar en (b), el espectro OFDM tiene altos niveles de lóbulos laterales resultantes de la Pulso rectangular en el dominio del tiempo. Esto causa severo ICI y degradación del rendimiento, si la ortogonalidad entre subportadoras colapsa (por ejemplo, por CFO). En contraposición a OFDM, UFMC ofrece un pulso de frecuencia bien localizado. Además, el UFMC propuesto ofrece una mayor eficiencia espectral en comparación con OFDM, ya que no requiere prefijo cíclico. (Schaich, F., & Wild, T., 2014).

El dominio del tiempo recibió la señal en un sistema CoMP-UFMC se, puede expresar como:

$$Y_m^i = \sum_{k=1}^K h_m^k T(\varepsilon_m^k) \check{F}^H \check{V}^H X_k^i + z_m^i \quad (7)$$

Donde

$$X_k^i = [X_{k,1}^{iT}, X_{k,2}^{iT}, \dots, X_{k,N_{RB}}^{iT}]^T \in C^{N_{sc} \times 1} \quad (8)$$

$$X_{k,p}^i = [X_{k,p}^i(0), X_{k,p}^i(1), \dots, X_{k,p}^i(N_{SC}^{RB} - 1)]^T \quad (9)$$

$$\check{V}^H = \text{diag}\{V_1, V_2, \dots, V_{N_{RB}}\}, \quad (10)$$

$$\check{F}^H = [F_1, F_2, \dots, F_{N_{RB}}] \quad (11)$$

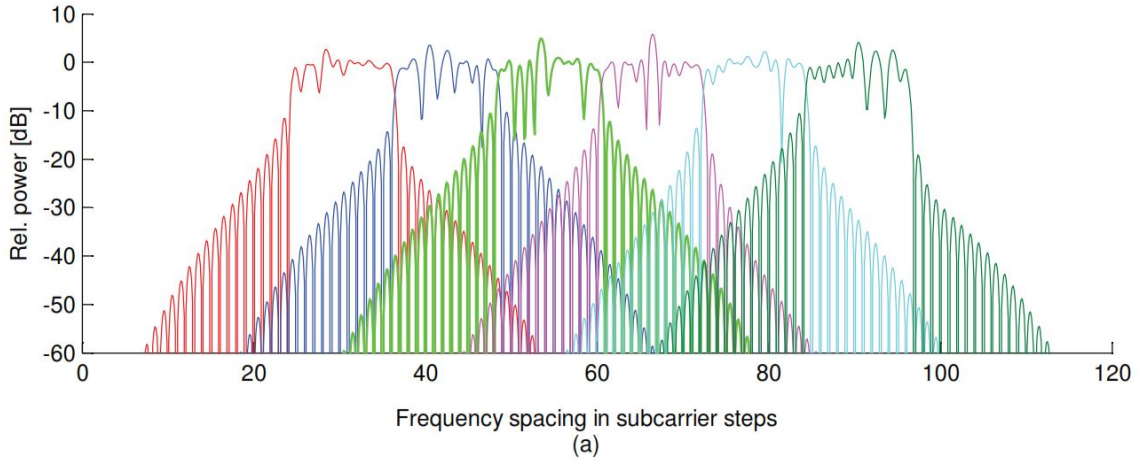


Figura 14: Espectro superpuesto de 6 diferentes RB UFMC con 12 subportadoras cada uno que llevan símbolos QPSK

Fuente: Schaich, F., & Wild, T., 2014

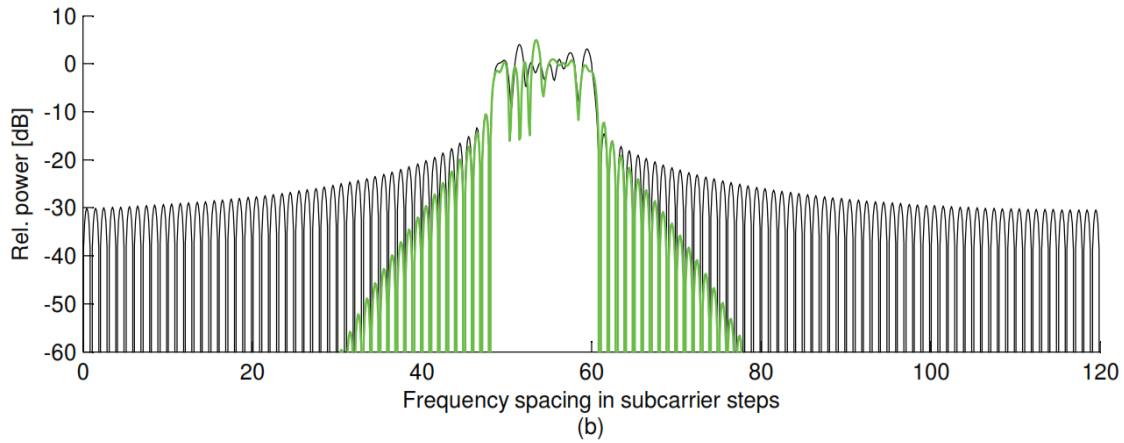


Figura 15: Comparación de un RB de OFDM con un RB de UFMC

Fuente: Schaich, F., & Wild, T., 2014

4.2.5. WOLA (Weighted Overlap y Add basado en OFDM)

WOLA-OFDM podría ser un candidato prometedor de forma de onda, superando tanto al CP-OFDM como al UFMC en cualquier escenario asíncrono. Las redes (5G) están en curso y son impulsadas por aplicaciones de comunicaciones (MTC) e *Internet of Thing* (IoT). Estas aplicaciones imponen nuevos retos debido a la conectividad inalámbrica masiva entre máquinas. Uno de estos desafíos es el acceso esporádico que requiere esquemas de sincronización relajados y acceso a un espectro fragmentado para limitar la sobrecarga de señalización.

Una primera clase de estas formas de onda reúne las que adoptan un modelado de pulso de per subcarrier para reducir la emisión de OOB y proporcionar flexibilidad en el diseño de la forma de onda: multiplexación por división de frecuencia generalizada (GFDM) y modulación de amplitud de cuadratura de offset basada en banco de filtros multiportadora (FBMC-OQAM) han sido muy estudiados. Sin embargo, estas formas de onda no están libres de inconvenientes. Por un lado, el FBMC-OQAM tiene la interferencia intrínseca

ya que las condiciones de ortogonalidad se cumplen solo en el campo real. Esto lo hace incompatible con la técnica de entradas múltiples de salida múltiple (MIMO), que se considera una tecnología clave para las redes inalámbricas. (Zayani, R., Medjahdi, Y., 2016).

En WOLA-OFDM se evita el uso convencional de la forma de pulso rectangular. Pero, en su lugar, se utiliza un pulso con bordes suaves. Estos bordes suaves se agregan a la extensión cíclica mediante una ventana de dominio de tiempo. Esto se traduce en una mejor contención espectral. La transición suave entre la última muestra de un símbolo dado y la primera muestra del siguiente símbolo se proporciona con la multiplicación de punto a punto de la función de ventana y el símbolo OFDM con prefijo cíclico y sufijo cíclico (ver Figura 23). Para crear el prefijo cíclico, copiamos y agregamos las muestras $CP + W$ desde la última parte de un símbolo dado hasta su comienzo. Además, el sufijo cíclico es agregar las primeras W muestras de un símbolo dado a su final. Por lo tanto, el símbolo de dominio de tiempo WOLA-OFDM se extiende cíclicamente de N muestras a $N + CP + 2W$. Dado que, tenemos que cumplir con los estándares, los símbolos adyacentes se superponen en la región de transición de borde, lo que resulta en una sobrecarga similar a la del CP-OFDM clásico. (Zayani, R., Medjahdi, Y., 2016).

Después de la inserción de la extensión cíclica, se aplica una ventana de longitud $W = N + CP + 2W$ muestras. De hecho, se han estudiado y comparado muchas funciones de ventanas en términos de mejorar la supresión del lóbulo lateral. La solución directa es definir el borde de la ventana del dominio del tiempo como un pulso de raíz levantada (RRC), consideramos el pulso Meyer RRC que combina el pulso RRC en el dominio del tiempo con la función auxiliar Meyer que se define como

$$v(k) = k^4(35 - 84k + 70k^2 - 20k^3) \quad (12)$$

$$F_w[k] = \begin{cases} 1 & \text{si } |k| \leq (1 - \alpha s) L/2 \\ \frac{1}{2} \left[1 + \cos \left(\pi v \left(\frac{|k| - \frac{(1 - \alpha s)L}{2}}{\alpha s L} \right) \right) \right] & \text{si } (1 - \alpha s) \frac{L}{2} < |k| \leq (1 + \alpha s) \frac{L}{2} \\ 0 & \text{otros casos} \end{cases} \quad (13)$$

αs es el símbolo Rolloff que controla la función de ventana y se expresa como la relación entre la longitud de la ventana y la longitud del símbolo.

Vale la pena señalar que la ventana de transmisión se utiliza para mejorar la contención espectral de la señal transmitida. Como se ilustra en la Figura 24, el procesamiento WOLA se aplica en el lado del receptor para mejorar la supresión de interferencia entre usuarios asíncrona

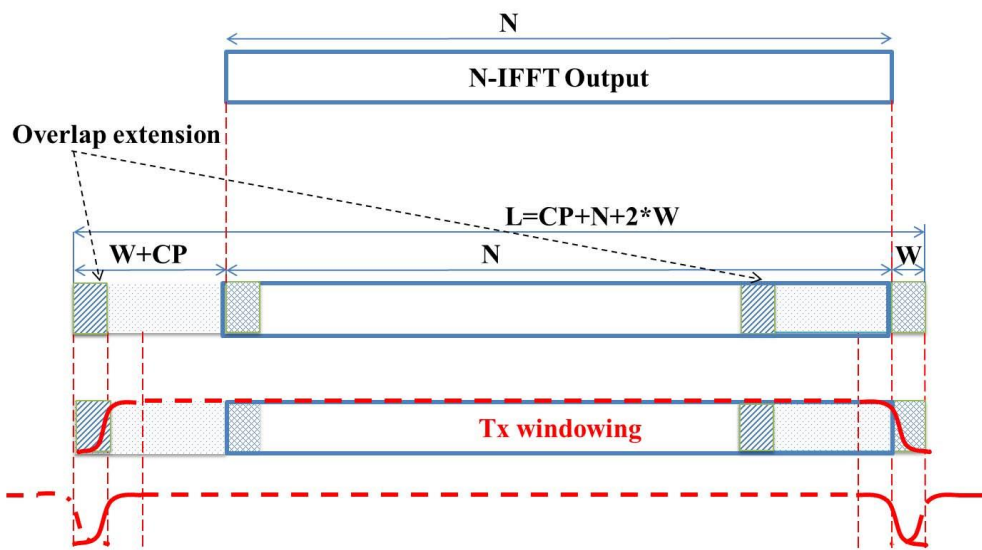


Figura 16: Transmisor WOLA-OFDM
Fuente: Zayani, R., Medjahdi, Y., 2016.

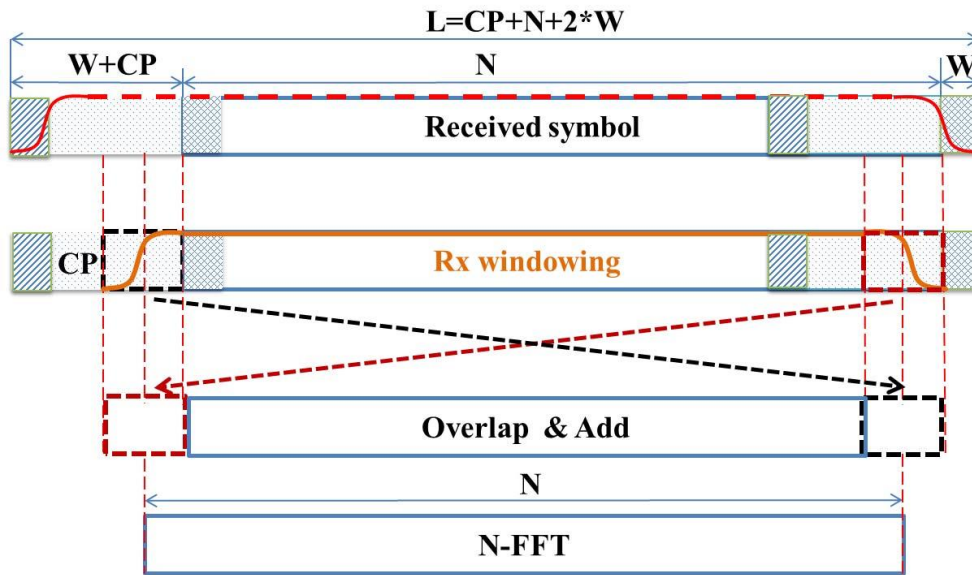


Figura 17: WOLA-OFDM Receptor

Fuente: Zayani, R., Medjahdi, Y., 2016.

Para comparar el rendimiento de estos contendientes de forma de onda, se considera un escenario multiusuario asíncrono. La descripción de este escenario es la siguiente:

Consideramos que dos usuarios coexisten en la misma área y comparten una banda de frecuencia determinada. Ambos usuarios transmiten de forma asíncrona en bandas adyacentes con la misma potencia de transmisión. (Gerzaguet, R., Kténas, D., Cassiau, N., & Doré, J. B. 2016)

Como se muestra en la Figura 18, se asigna un conjunto de 3 RB (de color azul) al usuario de interés. El segundo usuario está ocupando 9 RB (de color rojo) en cada lado de la banda útil. Se asume que el usuario de interés está perfectamente sincronizado con su receptor. Este último se encuentra a la misma distancia de ambos transmisores.

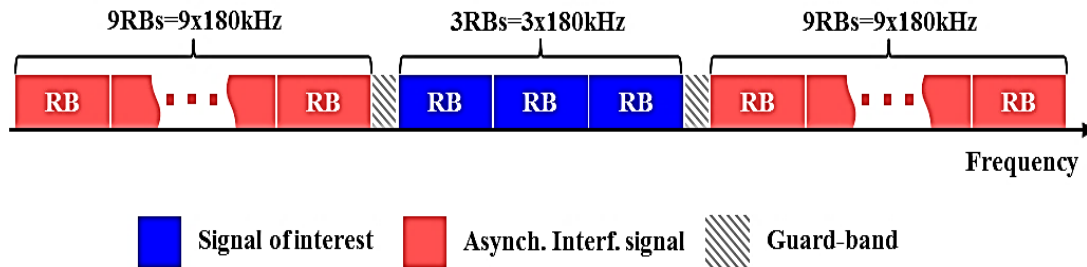


Figura 18: Coexistencia de 2 usuarios WOLA

Fuente: Zayani, R., Medjahdi, Y., 2016.

El asincronismo entre ambos usuarios puede causar una pérdida de ortogonalidad, lo que puede provocar interferencias asíncronas causadas por el segundo usuario. El resto de los parámetros se resumen en la Tabla, donde los parámetros generales se definen utilizando la configuración de LTE como referencia.

Tabla 2: Parametros generales, configuración LTE

General	
FFT	1024
Espacio entre subportadoras	15Khz
Frecuencia de muestreo	15.36MHZ
símbolos de datos de entrada	16QAM
RB	12
Número de RB útiles activos	3
Número de RB interferentes activos	18
CP-OFDM / WOLA-OFDM	
CP	72

Windowing	Meyer Root Raised cosine
Longitud de la ventana (W)	32
UFMC	
Filtro	Dolph-Chebyshev
Longitud del filtro ($LFIR = CP + 1$)	73
Atenuación de la banda de parada	40dB
Windowing	Coseno elevado

Fuente: Gerzaguet, R., Kténas, D., Cassiau, N., & Doré, J. B. 2016.

La comparación de PSD de los esquemas de modulación multiportadora (MCM) se muestra en la Figura 26. Solo se traza la segunda señal del usuario (interferente) para resaltar las emisiones de OOB en la banda de frecuencia de interés. Notamos claramente que el CP-OFDM clásico produce un lóbulo lateral espectral no despreciable de al menos -25dB. En contraste con CP-OFDM, podemos ver que WOLA-OFDM puede reducir considerablemente la OOB al proporcionar una ganancia de 20dB. Con respecto al UFMC, la ganancia obtenida se vuelve más significativa, lo que corresponde a los 40 dB de la atenuación de banda de parada considerada del filtro (consulte la Tabla 5). Vale la pena mencionar que la diferencia entre WOLA-OFDM y UFMC solo se puede observar para niveles de PSD por debajo de -45dB, lo que lleva a rendimientos similares. (Zayani, R., Medjahdi, Y., 2016)

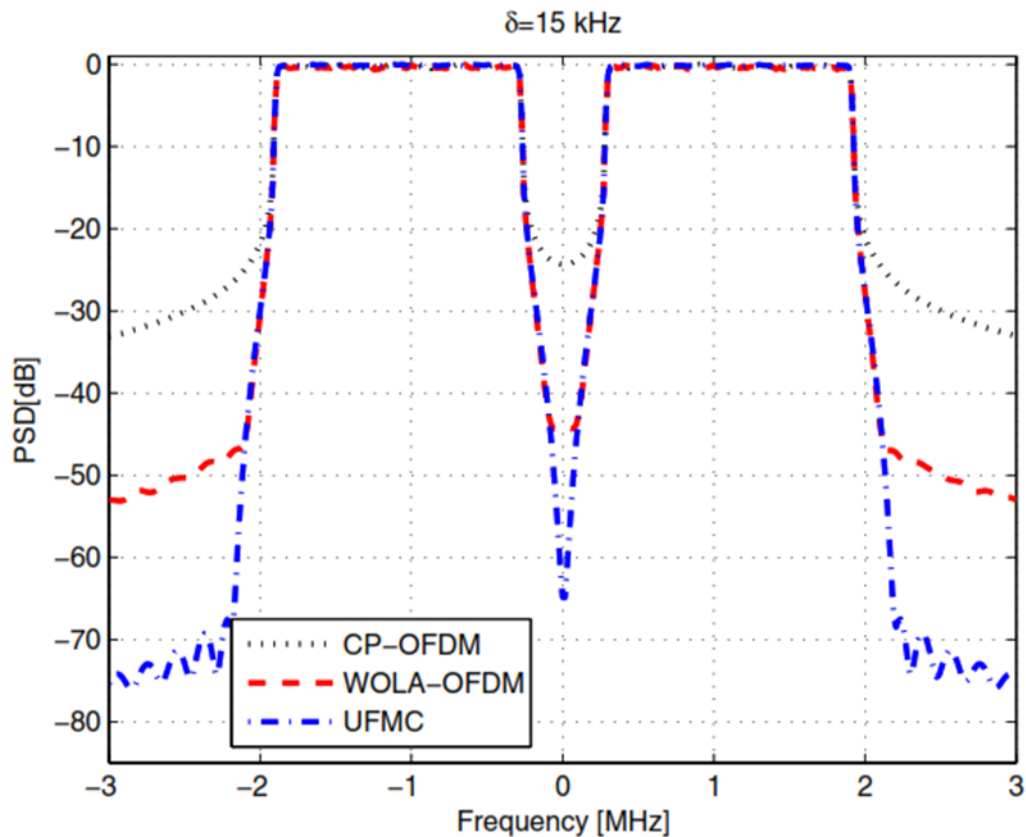


Figura 19: Densidad espectral de potencia de los 3 esquemas

Fuente: Zayani, R., Medjahdi, Y., 2016.

4.3. ACCESO MÚLTIPLE

4.3.1. OFDMA

El acceso múltiple por división ortogonal de frecuencia (OFDMA) se basa en la multiplexación por división de frecuencia ortogonal (OFDM); y, por lo tanto, hereda sus beneficios, al tiempo que permite explotar la diversidad multiusuario. Esto lleva a una administración de recursos de radio (RRM) más eficiente ya que el espectro se puede asignar a usuarios con las mejores condiciones de canal.

Debido a la alta eficiencia espectral y la transmisión robusta en presencia de desvanecimiento por trayectos múltiples, el OFDMA se utiliza como esquema de

modulación para el enlace descendente en los sistemas LTE. En el transmisor OFDMA, el espectro disponible se divide en número 12 subportadoras ortogonales. El espacio de la subportadora para el sistema LTE es de 15 KHz con una duración de símbolo OFDMA de 66,67 μ s. El flujo de datos de alta velocidad de bits pasa a través del modulador, donde se aplican esquemas de modulación adaptativos como BPSK, QPSK, 16-QAM y 64-QAM. Esta secuencia multinivel de símbolos modulados se convierte en componentes de frecuencia paralelos (subportadoras) mediante un convertidor de serie a paralelo. La etapa IFFT convierte estos símbolos de datos complejos en un dominio temporal y genera símbolos OFDM. Se usa una banda de guarda entre los símbolos OFDMA para cancelar la interferencia entre símbolos en el receptor. En LTE, esta banda de protección se denomina Prefijo cíclico (CP) y la duración del CP debe ser mayor que la respuesta al impulso del canal o la dispersión del retardo. El receptor no trata con el ISI, pero aún debe tener en cuenta el impacto del canal para cada subportadora que haya experimentado cambios de amplitud y una fase dependiente de la frecuencia. En LTE, OFDMA utiliza dos tipos de CP, es decir, CP normal y CP extendido. El CP normal se utiliza para altas frecuencias (áreas urbanas) y CP extendido para frecuencias más bajas (áreas rurales). En el receptor, primero se quita el CP y luego las subportadoras se convierten de secuencia paralela a serie. La etapa FFT convierte adicionalmente los símbolos en dominio de frecuencia seguido por ecualizador y demodulación (Zyren, J., McCoy, W. 2017).

Para mejorar la capacidad del sistema, los datos máximos y la confiabilidad de la cobertura, la señal transmitida a un usuario en particular se modifica para tener en cuenta la variación de la calidad de la señal a través de un proceso comúnmente denominado modulación y codificación adaptativa (AMC). AMC proporciona la flexibilidad para unir el esquema de codificación de modulación a las condiciones de canal promedio entre sí. Con AMC, la potencia de la señal transmitida se mantiene constante durante un intervalo de cuadro, y el formato de modulación y codificación se cambia para que coincida con la calidad actual de la señal recibida o las condiciones del canal. Por ejemplo, AMC puede emplear QPSK para canales ruidosos y 16 QAM para canales más claros. El primero es más robusto y puede tolerar niveles más altos de interferencia, pero tiene una menor velocidad de transmisión. El último tiene una velocidad de bits dos veces mayor, pero es más propenso a errores debido a la interferencia y el ruido; por lo tanto, requiere una

codificación de corrección de errores hacia adelante (FEC) más fuerte que a su vez significa más bits redundantes y menor velocidad de bits de información. En el enlace descendente, las subportadoras se dividen en bloques de recursos. Esto permite que el sistema divida las subportadoras en partes pequeñas, sin mezclar los datos a través del número total de subportadoras para un ancho de banda dado. El bloque de recursos consta de 12 subportadoras para una única ranura de tiempo de 0,5 ms de duración. La estructura del bloque de recursos físicos (PRB). Existen diferentes números de bloques de recursos para diferentes anchos de banda de señal en LTE, como se muestra en la Tabla

Tabla 3: Parámetros de OFDMA

Ancho de Banda (Mhz)	1.4	3	5	10	15	20
PRB ancho de banda (Khz)	180					
Ancho de banda subportadora (Khz)	15					
Número de RB	6	15	25	50	75	100

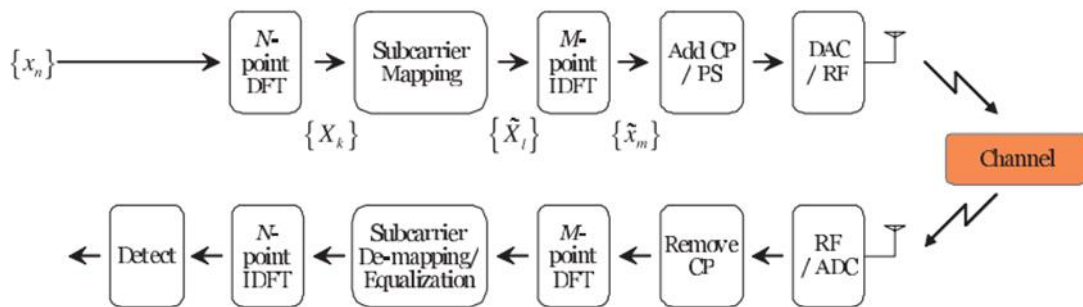
Fuente: Zyren, J., McCoy, W. 2007

4.3.2. SC-FDMA

A diferencia de OFDMA, el SC-FDMA usa una etapa de DFT (transformada discreta de Fourier) de punto N adicional en el transmisor y una etapa de IDFT de N-punto en el receptor. En SC-FDMA, los datos se mapean en la constelación de señales de acuerdo con la modulación QPSK, 16-QAM o 64-QAM, dependiendo de las condiciones del canal de manera similar a OFDMA. Considerando que, los símbolos QPSK / QAM no modulan directamente las subportadoras; estos símbolos pasan a través de un convertidor de serie a paralelo seguido de un bloque DFT que produce una representación de dominio de frecuencia discreta de los símbolos QPSK / QAM. La forma del pulso va seguida por un elemento DFT, pero es opcional y algunas veces necesita dar forma a la señal de salida desde DFT. Si la conformación del pulso está activa, entonces en la señal actual, se

produce la extensión del ancho de banda. Los símbolos de Fourier discretos de la salida del bloque DFT se asignan luego con las subportadoras en el bloque de asignación de subportadora. Después de mapear el dominio de la frecuencia; el modulado las subportadoras pasan a través de IDFT para la conversión de dominio de tiempo. El resto de la operación del transmisor es similar a OFDMA (Zyren, J., McCoy, W. 2017).

SC-FDMA



OFDMA

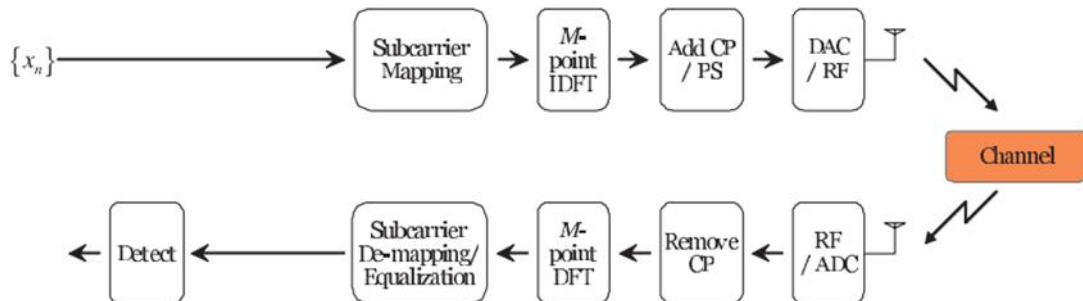


Figura 20: Comparación esquema de transmisión OFDMA vs SC-FDMA

Fuente: Zyren, J., McCoy, W. 2017

4.3.3. NOMA (Acceso múltiple no ortogonal)

En los últimos años, los esquemas de acceso múltiple no ortogonal (NOMA) han recibido una atención significativa para las redes celulares de quinta generación (5G).

La razón principal para adoptar NOMA en 5G se debe a su capacidad de servir a múltiples usuarios utilizando los mismos recursos de tiempo y frecuencia. Ahí salen dos técnicas

principales de NOMA: power-domain y code-domain, además de estas técnicas, que son las más conocidas, existen otras técnicas que se detallan de manera general más adelante. Power-domain NOMA logra la multiplexación en el dominio de potencia, mientras que code-domain NOMA logra la multiplexación en el dominio del código. (Islam S, Zeng M, Dobre O, 2017).

NOMA en el dominio de la potencia explota la codificación de superposición en el transmisor y la cancelación de interferencia sucesiva (SIC) en el receptor, multiplexando así a los usuarios en el dominio de potencia. Como se muestra en la figura, la estación base (BS) envía las señales superpuestas a dos usuarios, donde el usuario 1 tiene mayor ganancia de canal que el usuario 2.

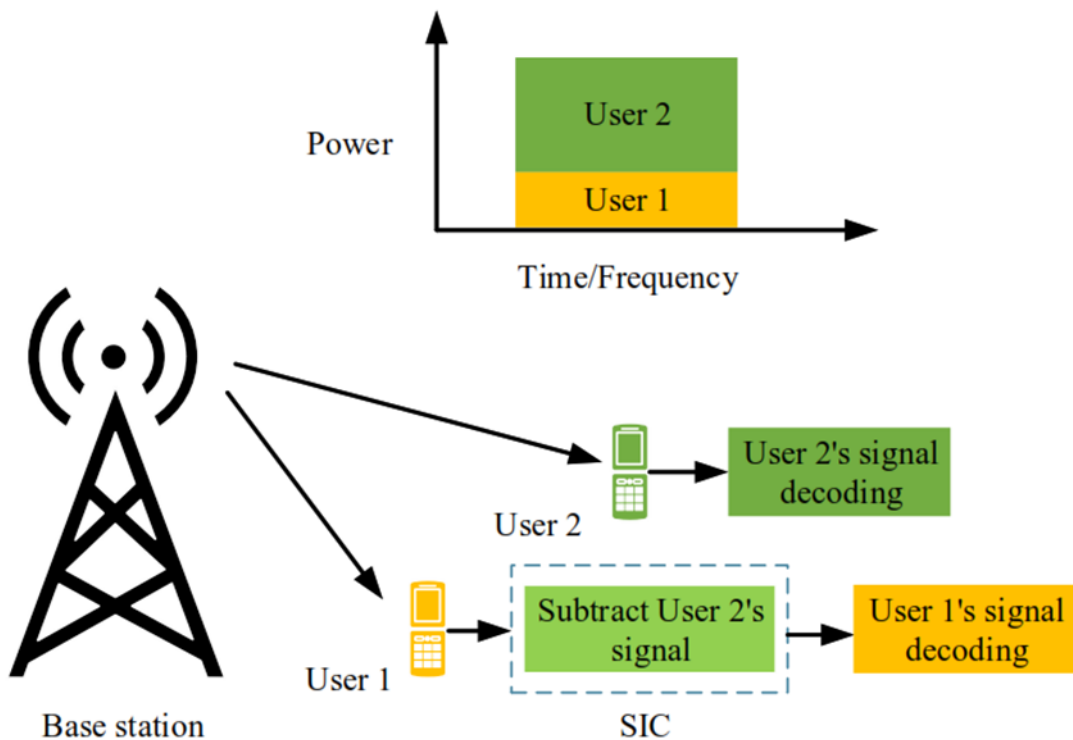


Figura 21: Downlink en NOMA en una celda con una estación base y dos usuarios

Fuente: Islam S, Zeng M, Dobre O, 2017

En NOMA Power-domain, el usuario con mayor ganancia de canal y el usuario con menor ganancia de canal son conocidos como el usuario fuerte y el usuario débil respectivamente. El usuario fuerte primero resta la señal del usuario débil a través de SIC, y luego decodifica su propia señal; el usuario débil considera la señal del usuario fuerte como ruido y detecta su propia señal directamente. Con una peor ganancia de canal y más interferencia, al usuario débil se le asigna más potencia en NOMA para garantizar la equidad. (Islam S, Zeng M, Dobre O, 2017).

4.3.3.1. SCMA (Sparse code multiple Access)

SCMA es un esquema mejorado basado en LDS-OFDM que explota diversas secuencias de dispersión para realizar sobrecarga. A diferencia de LDS-OFDM, el mapeo de bit a constelación y la dispersión con firmas de baja densidad se combinan juntas en SCMA, es decir, los flujos de bits se asignan directamente a diferentes palabras de código dispersas. En particular, las palabras de código de diferentes usuarios se toman de diferentes libros de códigos, que se generan mediante una constelación multidimensional. Al hacer un uso completo de la escasez de palabras de código, en el receptor, el MUD (multi-user detection) basado en MPA (message passing algorithm) se puede usar para separar símbolos con una complejidad aceptable. En esta subsección, se discutirá primero el modelo de sistema de SCMA y luego se presentará el MUD basado en MPA. (Wang, B, Lu Z, Xie T, 2015)

4.3.3.2. MUSA (Multi-user shared access)

MUSA es otro esquema de NOMA a través de la multiplexación de dominio de código, que puede considerarse como un esquema mejorado de CDMA. A diferencia de CDMA convencional, MUSA puede realizar sobrecarga utilizando secuencias de propagación de baja correlación en el transmisor. En el receptor, el SIC se realiza para separar símbolos superpuestos de acuerdo con la diferencia de relación de potencia de señal a interferencia recibida más ruido (SINR).

En el sistema MUSA de enlace ascendente, todos los símbolos transmitidos de un usuario se multiplican con la misma secuencia de propagación. Por supuesto, las diferentes secuencias de expansión también se pueden usar para diferentes símbolos para el mismo usuario, lo que puede obtener además el beneficio del promedio de interferencia. Luego, todos los símbolos después de la difusión se transmiten por los mismos recursos de frecuencia de tiempo, por ejemplo, subportadoras de OFDM. Sin pérdida de generalidad, suponemos que un usuario simplemente transmite un símbolo cada vez. (Wang, B, Lu Z, Xie T,2015)

4.3.3.3. PDMA (Pattern division multiple access)

PDMA es un nuevo esquema de NOMA que se puede realizar en múltiples dominios. En el transmisor, PDMA utiliza patrones no ortogonales que están diseñados para maximizar la diversidad y minimizar las superposiciones de múltiples usuarios. Entonces, la multiplexación se puede realizar en dominio de código, dominio de potencia, dominio espacial o sus combinaciones. La multiplexación en el dominio de código es similar a la de SCMA, pero el número de subportadoras conectadas a los mismos símbolos en el gráfico de factor puede ser diferente. En el caso de la multiplexación en el dominio de potencia, la asignación de potencia debe considerarse cuidadosamente bajo la restricción de potencia total.

La multiplexación en el dominio del espacio, es decir, PDMA espacial, se puede combinar con la técnica de múltiples antenas. La ventaja de PDMA espacial en comparación con MIMO multiusuario (MU-MIMO) es que PDMA no requiere precodificación conjunta para realizar la ortogonalidad espacial, lo que reduce significativamente la complejidad del sistema.

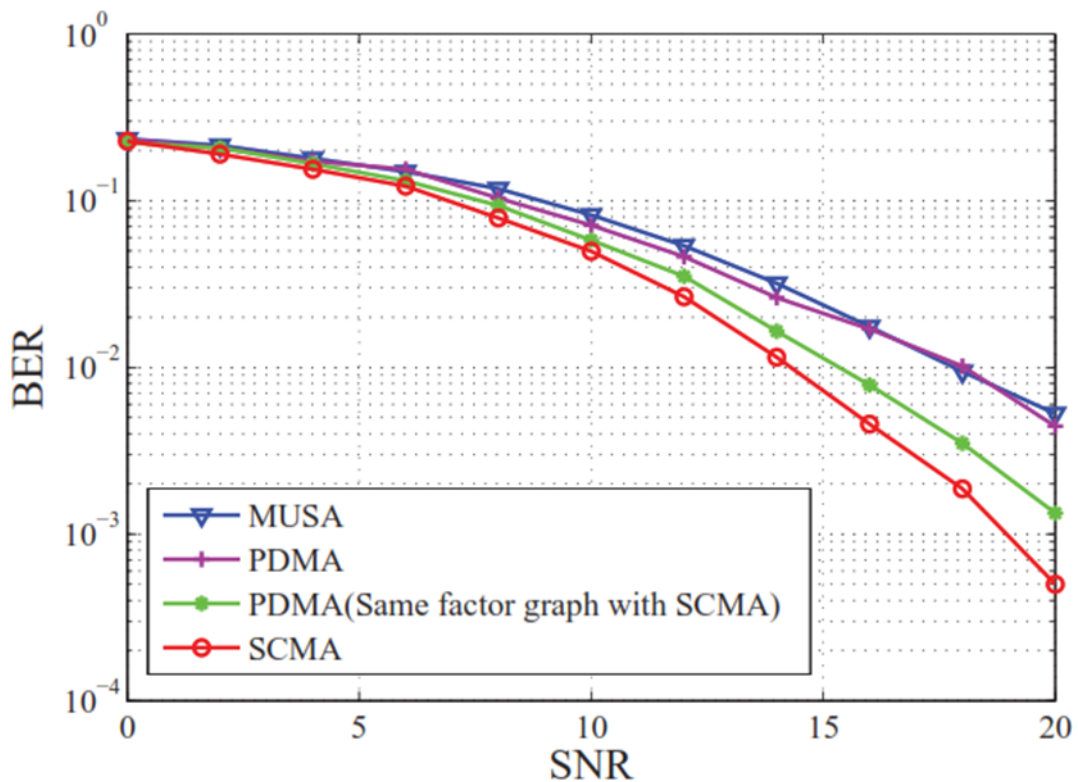


Figura 22: Comparación del rendimiento BER de SCMA, MUSA y PDMA en canales de desvanecimiento de Rayleigh.

Fuente: Wang, B., Wang, 2015

4.3.3.4. Beneficios de NOMA

NOMA domina el acceso múltiple ortogonal convencional (OMA) en varios aspectos, tales como:

- 1) Alcanza una eficiencia espectral superior al servir a múltiples usuarios al mismo tiempo y con el mismo recurso de frecuencia, y al mitigar la interferencia a través del SIC.
- 2) Aumenta la cantidad de usuarios atendidos simultáneamente y, por lo tanto, puede admitir conectividad masiva.

3) Debido a la naturaleza de transmisión simultánea, un usuario no necesita pasar por un intervalo de tiempo programado para transmitir su información, y por lo tanto, experimenta una latencia menor.

4) NOMA puede mantener la imparcialidad del usuario y la calidad de servicio diversa mediante un control de potencia flexible entre los usuarios fuertes y débiles; en particular, a medida que se asigna más potencia a un usuario débil, NOMA ofrece un mayor rendimiento en el límite de la célula y, por lo tanto, mejora la experiencia del usuario de borde de celda. (Islam S, Zeng M, Dobre O, 2017)

4.3.3.5. Limitaciones de NOMA

Deben abordarse diversas limitaciones y problemas de implementación para explotar todas las ventajas de NOMA, tales como:

1) Cada usuario necesita decodificar la información de todos los demás usuarios con peores ganancias de canal (que están en el mismo clúster) antes de decodificar su propia información, que conduce a una mayor complejidad del receptor y al consumo de energía en comparación con OMA.

2) Cuando se produce un error en SIC en un usuario, es probable que la decodificación posterior de la información de todos los demás usuarios se lleve a cabo erróneamente. Esto implica mantener el número de usuarios en cada grupo razonablemente bajo para reducir el efecto de la propagación de errores.

3) Para obtener los beneficios alegados de la multiplexación de dominio de potencia, se requiere una considerable diferencia de ganancia de canal entre los usuarios fuertes y

débiles. Esto restringe intuitivamente la cantidad efectiva de pares de usuarios, lo que a su vez reduce la ganancia de la suma de NOMA.

4) Cada usuario necesita devolver su información de ganancia de canal a la BS, y NOMA es inherentemente sensible a la incertidumbre en la medición de esta ganancia. (Islam S, Zeng M, Dobre O, 2017).

5. MATERIALES Y MÉTODOS

La metodología de investigación es de carácter mixto (cualitativa y cuantitativa) y se utilizaron los siguientes métodos:

Método científico: consiste en recolectar información del fenómeno a tratar (en este caso modulación y acceso múltiple para 5G), luego realizar un hipotesis de razonamiento y crear un entorno de pruebas (GUIDE de simulación) para poder tener resultados que se puedan sustentar con argumentos veraces.

Método investigativo: Utilizado para reunir información previa con la finalidad de construir los sustentos teóricos sobre los nuevos esquemas de modulación y acceso múltiple que se han propuesto para 5G. Con el uso de este método fue posible redactar todos los capítulos teóricos que presenta esta tesis, además de que a través del mismo se pudo juntar todos los sustentos necesarios para poder realizar un entorno gráfico (GUIDE) de simulación de los parámetros de cada modulación.

Método deductivo: Consiste en tomar como base las investigaciones previas, de esta manera partir de resultados de investigaciones anteriores sobre la modulación y el acceso múltiple para 5G, así luego de evaluar los resultados de otros investigadores es posible crear un marco referencial un poco más completo sobre el presente tema.

Con este método fue posible utilizar tanto el sustento teórico, así como también simulaciones previas encontradas y de esta manera poder unir todo este material y generar un programa y documento que sea un sustento verídico de la comparación de los diferentes esquemas de modulación y acceso múltiple para la quinta generación de comunicaciones móviles, además de que sea posible realizar un programa que presente una interfaz lo más amigable posible con el usuario.

Método inductivo: Es una forma de razonamiento por medio de la cual se pasa del conocimiento de casos particulares a un conocimiento más general que refleja lo que hay de común en los fenómenos individuales, con este método encontraremos las similitudes que existen entre los diferentes esquemas propuestos.

El uso de este método de investigación que se aplica por medio de la observación de graficas de eficiencia, retardo y de rendimiento, nos ayuda a obtener los resultados sobre el comportamiento que presenta cada esquema de modulación y acceso múltiple y de esta manera poder encontrar las similitudes y diferencias existentes entre cada esquema propuesto de esta manera encontrar el esquema que mejor se adapte a los requerimientos de 5G

Para poder llevar a cabo la simulación de los diferentes esquemas de modulación que se han propuesto para 5G se utilizó un computador portátil marca DELL con un procesador core i5 de cuarta generación con una velocidad de procesamiento de 1,70 GHz y con una memoria RAM de 4GB.

Se procede a instalar el software Matlab en su versión r2018a lanzada en el primer semestre de 2018(figura 23) que cuenta con una librería especializada llamada LTEtoolbox con soporte para simulación de diversas modulaciones las cuales son estudiadas en este documento, Una vez instalado el software se instaló una librería adicional lanzada en junio de 2018 con soporte para modulaciones para redes móviles de quinta generación llamada LTE System Toolbox 5G Library.

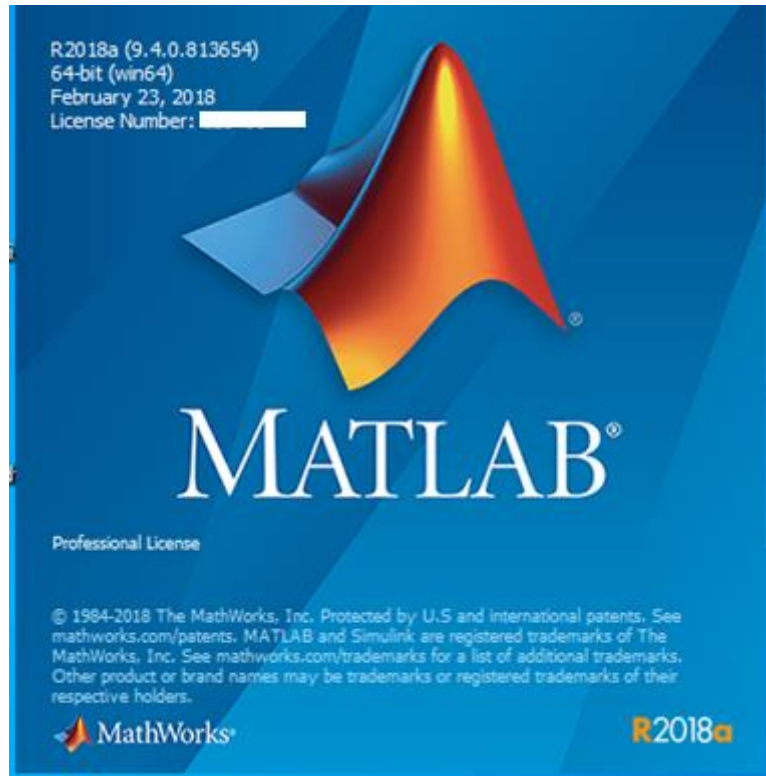


Figura 23: Versión utilizada de MATLAB

Fuente: El Autor

Para poder llevar a cabo la simulación de todos los esquemas y poder encontrar el más apropiado se empezó por buscar simulaciones previas que brinden una pauta de como empezar a realizar la evaluación de los diferentes esquemas de modulación, siendo esta aun una tecnología en desarrollo a la fecha. El proyecto más completo es el de la universidad de Viena, que han desarrollado algunas librerías compatibles con Matlab para poder realizar pruebas sobre redes 5G. La mejor forma para realizar pruebas con los diferentes esquemas de modulación y evaluar su comportamiento en diferentes escenarios es realizar un entorno grafico (GUIDE), tomando como base el proyecto de la universidad de Viena además de complementar con el conocimiento de programación en Matlab se decidió que este entorno de simulación permite editar los parámetros de cada esquema y obtener graficas de densidad espectral de potencias, retardo y eficiencia energética.

Las principales actividades desarrolladas para llevar a cabo la investigación comenzaron con la recolección de información de fuentes confiables, para esto el uso de Google scholar, la base de datos de IEEE, foros de Matlab, post realizados por empresas que llevan a cabo la investigación de 5G tales como Qualcomm o Ericsson, además de las publicaciones de los principales grupos de trabajo sobre 5G como son 5G-PPP, METIS y 5G FORUM.

Una vez entendido cuales son los esquemas mejor perfilados para 5G, así como su funcionamiento se procedió a realizar la evaluación de los esquemas, la evaluación en el entorno de simulación se hizo con los parámetros que utiliza *LTE-Advanced* ya que estos parámetros se mantendrán de la misma manera en 5G, los parámetros que se utilizaron para obtener las gráficas de simulación se detallan en la tabla 4, cabe recalcar que el entorno de simulación nos permite variar todos estos parámetros y realizar pruebas con más o menos portadoras por ejemplo o variar el orden de modulación digital, el entorno también nos permite variar el espacio entre subportadoras y el número de símbolos en el tiempo, en el caso de los esquemas que utilizan filtros también es posible variar la longitud de los filtros

En 5G la modulación digital que será implementada en la mayoría de dispositivos de comunicaciones móviles será 64QAM como lo explica en (Li, C., Wang, M., 2017) debido a que ofrece una buena tasa de transmisión de bits, una buena transmisión de información en ambientes con poca interferencia, además de ser una tecnología de modulación digital conocida y ya implementada. A pesar de que será el sistema de modulación digital mas usado, la modulación adaptativa en 5G nos permitirá también hacer uso de modulaciones de menor orden como 16QAM o de mayor orden para aplicaciones que así lo requieran como 256QAM. Por tal motivo las simulaciones llevadas a cabo en esta investigación se lo realiza con una modulación digital del orden de 64QAM.

Tabla 4: PARAMETROS DE SIMULACIÓN

PARAMETROS DE SIMULACIÓN					
ESQUEMA DE MODULACION	OFDM	F-OFDM	FBMC	WOLA	UFMC
Numero de subportadoras	24	24	24	24	24
Espacio entre subportadoras (KHz)	15	15	15	15	15
Número de símbolos en el tiempo	14	14	14	14	14
Longitud prefijo cíclico	1/14Esub	1/14Esub	X	0	0
Longitud del filtro RX	X	0.2	X	X	0.07
Longitud ventana TX				1/14Esub	
Longitud ventana RX				1/14Esub	
Modulación digital	64QAM	64QAM	64QAM	64QAM	64QAM

Fuente: El Autor

Estos parámetros presentados en la Tabla 4 son los esenciales para poder llevar a cabo la simulación de los diferentes esquemas de modulación para 5G, a más de estos parámetros existen otros que son característicos de cada esquema de modulación, es el caso de FBMC en el que se puede variar el filtro que se va a utilizar para la modulación,

El entorno de simulación realizado se lo puede observar en la figura 24, el cual se lo ejecutó con la ayuda del Ingeniero en electrónica y telecomunicaciones Diego Carrera actualmente cursando sus estudios de doctorado en el instituto tecnológico y de estudios superiores de monterrey el correo para contacto (dfcarrera79@me.com) quien puso a disposición de esta investigación su completa asesoría, además del Ingeniero en electrónica Dietmar Cerpa experto en MATLAB actualmente trabaja como freelancer PRO en la plataforma WORKANA para el desarrollo de aplicativos IDE en distintos lenguajes de programación como Android, c, c++, c# entre otros, el medio para contacto (<https://www.workana.com/freelancer/17b0c5d05df7d11ee674eaf820fa651b#section-certifications>).

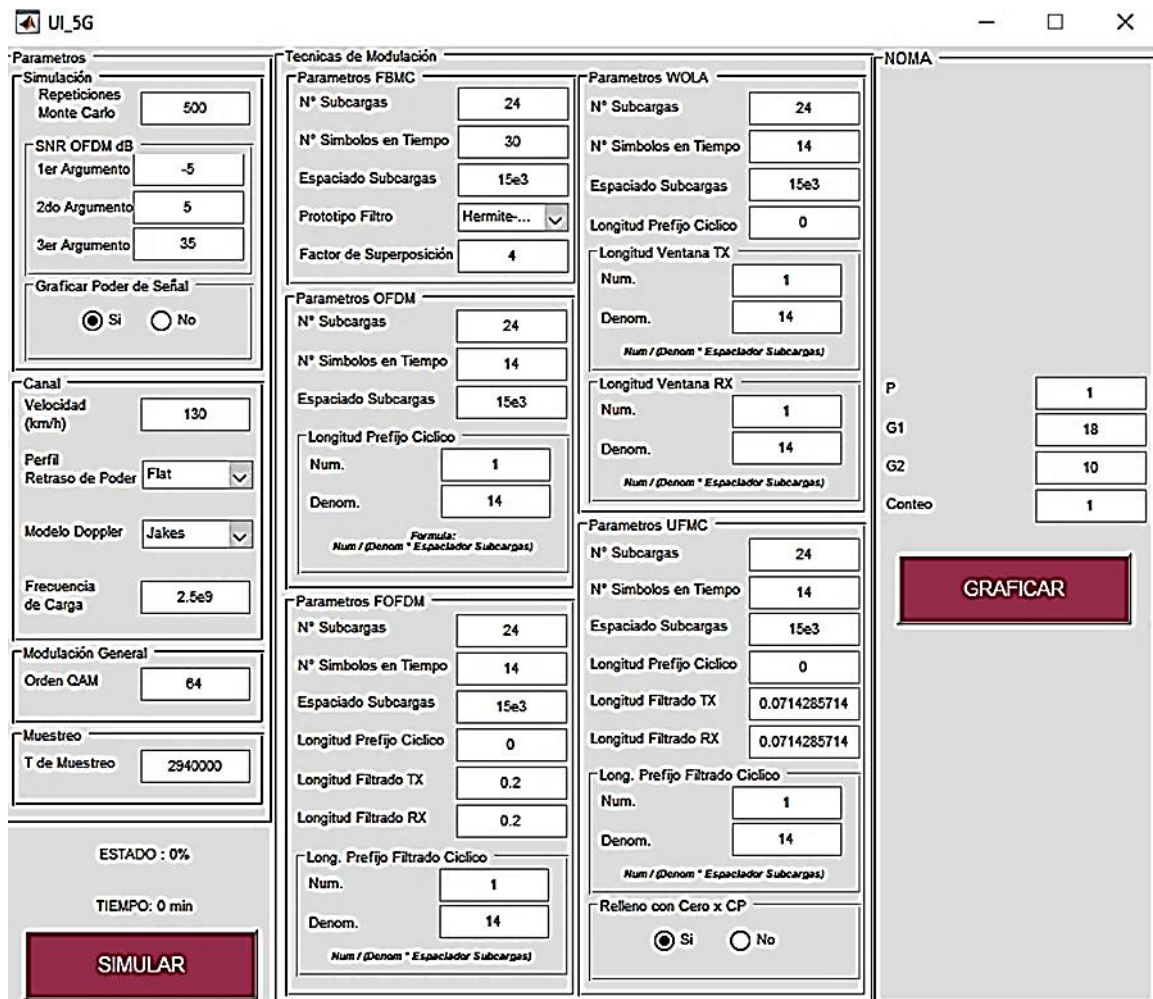


Figura 24: Entorno de simulación

Fuente: El Autor

6. RESULTADOS

El análisis de los resultados de la investigación es realizado a través de la información que nos brindan cada gráfica de simulación contrastando esta información con la teoría recabada de investigaciones previas, para las gráficas que se presenta a continuación los parámetros son los presentados en la tabla 4 del apartado de metodología, estos parámetros son los que se utilizaran para el despliegue de 5G, cada gráfica se evalúa en referencia al esquema que actualmente está en uso en las redes 4G-LTE y *LTE- Advanced* que es OFDM, al final del análisis de las gráficas se resume las diferencias y similitudes que presentan cada esquema en una tabla comparativa.

Como parte de la investigación rumbo a 5G también se crea un apartado donde se compara el acceso múltiple ortogonal OFDMA vs el acceso múltiple no ortogonal NOMA, en este caso específico hablando de acceso múltiple se presenta un escenario en donde dos usuarios comparten los mismos recursos de radio, pero se encuentran ubicados a distancias diferentes, el análisis de las gráficas se presenta más adelante.

6.1. Densidad Espectral de Potencia

La evaluación de la densidad espectral de potencia es sin duda el apartado que brinda la mayor información sobre el aprovechamiento del espectro electromagnético disponible, consciente de que el mayor aprovechamiento del espectro electromagnético se traduce en una mayor tasa de bits transmitidos, también en un mayor número de portadoras acomodadas en la misma porción de espectro y además mientras mejor sea aprovechado el espectro será posible generar más servicios para una mayor cantidad de usuarios, el análisis de este resultado lo observamos en las gráficas siguientes con una evaluación de cada esquema por separado, por último en la gráfica podremos observar de manera más general las diferencias que presentan cada esquema en cuanto a aprovechamiento del espectro se refiere.

6.1.1. FBMC

Número de sub portadoras = 24

Numero de símbolos en el tiempo = 14

Espacio entre subportadoras = 15Khz

Filtro Prototipo = Hermite-QAM

Factor de superposición = 4

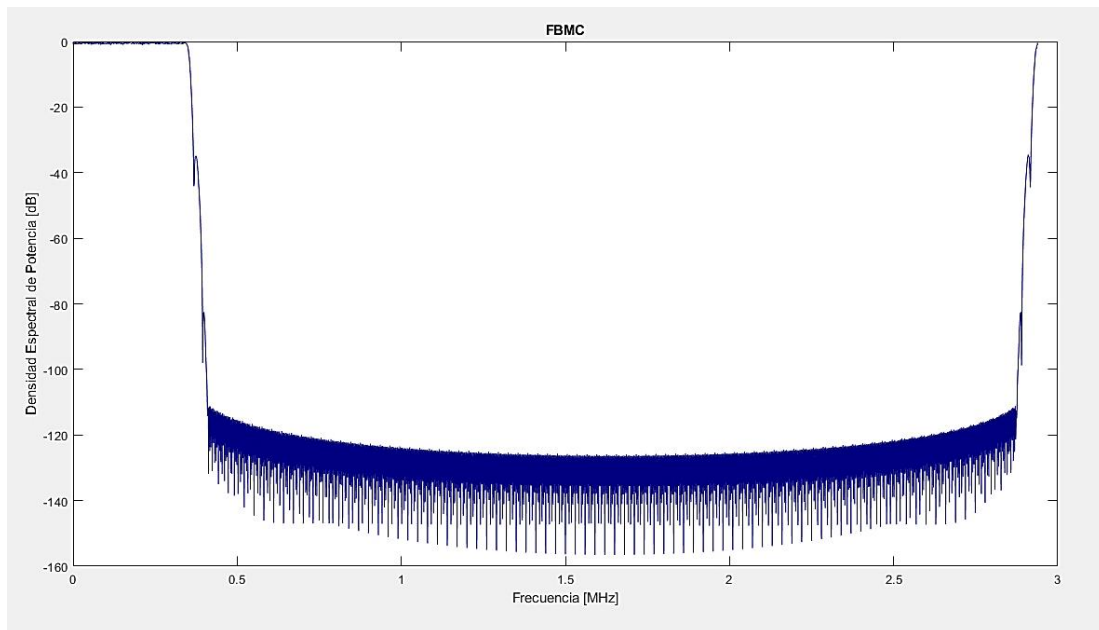


Figura 25: Densidad Espectral de Potencia FBMC

Fuente: El Autor

En la figura 25 observamos que el comportamiento de la onda FBMC presenta una pérdida de potencia espectral de alrededor de -160dB para los parámetros que se ingresan al principio del apartado, una transmisión óptima no tendría pérdidas de ningún sentido. Estos -160 dB reflejan que FBMC a pesar de que se enfoca en crear un conjunto de filtros para el tratamiento de las portadoras aun presenta pérdidas considerables en la transmisión, es precisamente este banco de filtrado común el que genera estas pérdidas ya que todas las portadoras se tratan de la misma forma, en independencia de la información que se transmita o al dispositivo al que se dirija, todas las portadoras pasan

por un mismo filtro y la única diferenciadora de información en este caso es la modulación adaptativa.

6.1.2. OFDM

Número de sub portadoras = 24

Numero de símbolos en el tiempo = 14

Espacio entre subportadoras = 15Khz

Prefijo Cíclico = 1/14(espacio entre subportadoras)

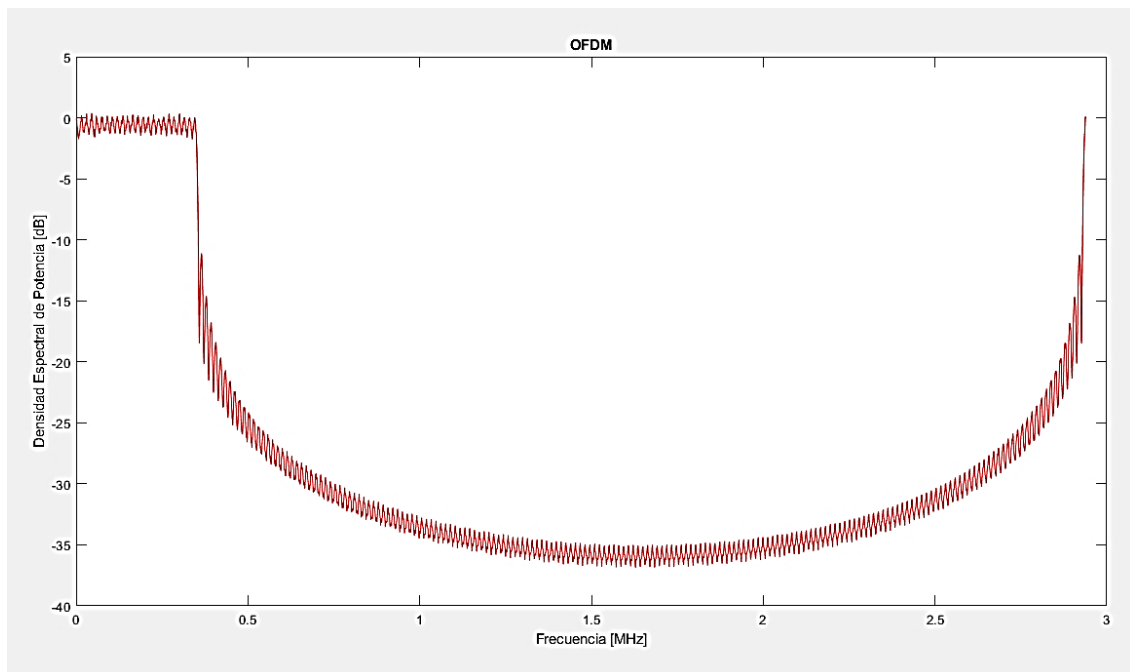


Figura 26: Densidad Espectral de Potencia

Fuente: El Autor

En la figura 26 OFDM presenta una pérdida de alrededor de -40dB, OFDM es claramente superior al comportamiento de FBMC ya que OFDM hace uso del prefijo cíclico en lugar de un banco de filtros común para las todas las subportadoras. El prefijo cíclico (CP-OFDM) a pesar de que teóricamente ocupa una porción mayor de espectro controla de forma óptima la interferencia entre portadoras contiguas, el espectro OFDM se acerca más al óptimo (sin pérdidas) por tal motivo es la tecnología que está vigente actualmente

y desde este punto una de las más perfiladas para llevar a cabo la transmisión de información 5G.

6.1.3. WOLA

Número de sub portadoras = 24

Numero de símbolos en el tiempo = 14

Espacio entre subportadoras = 15Khz

Prefijo Cíclico = 0

Longitud ventana de transmisión = $1 / (14 * 2 \text{espacio entre subportadoras})$

Longitud ventana de recepción = $1 / (14 * 2 \text{espacio entre subportadoras})$

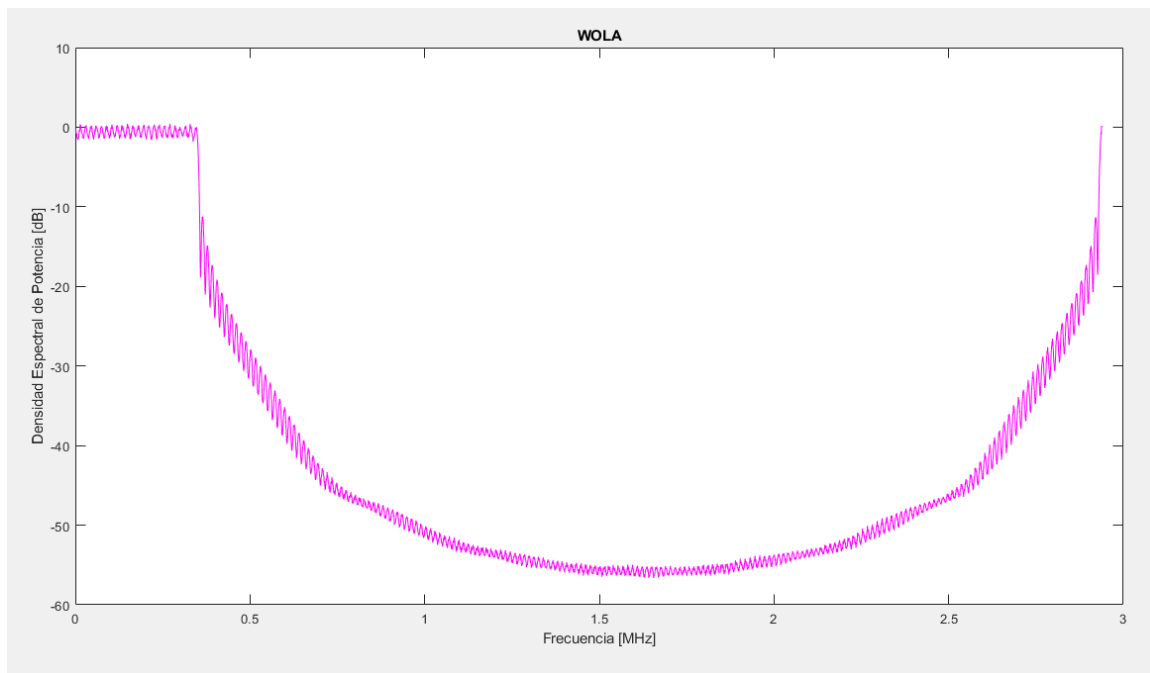


Figura 27: Densidad Espectral de Potencia WOLA

Fuente: El Autor

WOLA presenta una pérdida respecto a la densidad espectral de potencia de -60dB como se observa en la figura 27 teniendo un comportamiento superior a FBMC pero aun por debajo de OFDM, este esquema caracterizado por el uso de ventanas de transmisión tiene un comportamiento muy similar a CP-OFDM ya que tanto OFDM como WOLA rellenan

con ceros la trama de símbolos de información para que todo el espectro sea usado, de esta forma se combate el ISI e ICI, WOLA se encuentra aún por debajo de OFDM ya que se añade ventanas de símbolos en la transmisión y estas ventanas de símbolos (ceros) pasan por un proceso de demodulación un poco más complejo, este proceso supone un coste superior de espectro ya que las portadoras se transmiten acompañadas de una cola de símbolos OFDM que son utilizados para la corrección de errores.

6.1.4. f-OFDM

Número de sub portadoras = 24

Numero de símbolos en el tiempo = 14

Espacio entre subportadoras = 15Khz

Prefijo Cíclico = 0

Longitud filtro de transmisión = $1 \cdot (0.2 / \text{espacio entre subportadoras})$

Longitud filtro de recepción = $1 \cdot (0.2 / \text{espacio entre subportadoras})$

Longitud de prefijo filtrado cilicio = $1 / (14 \text{ espacio entre subportadoras})$

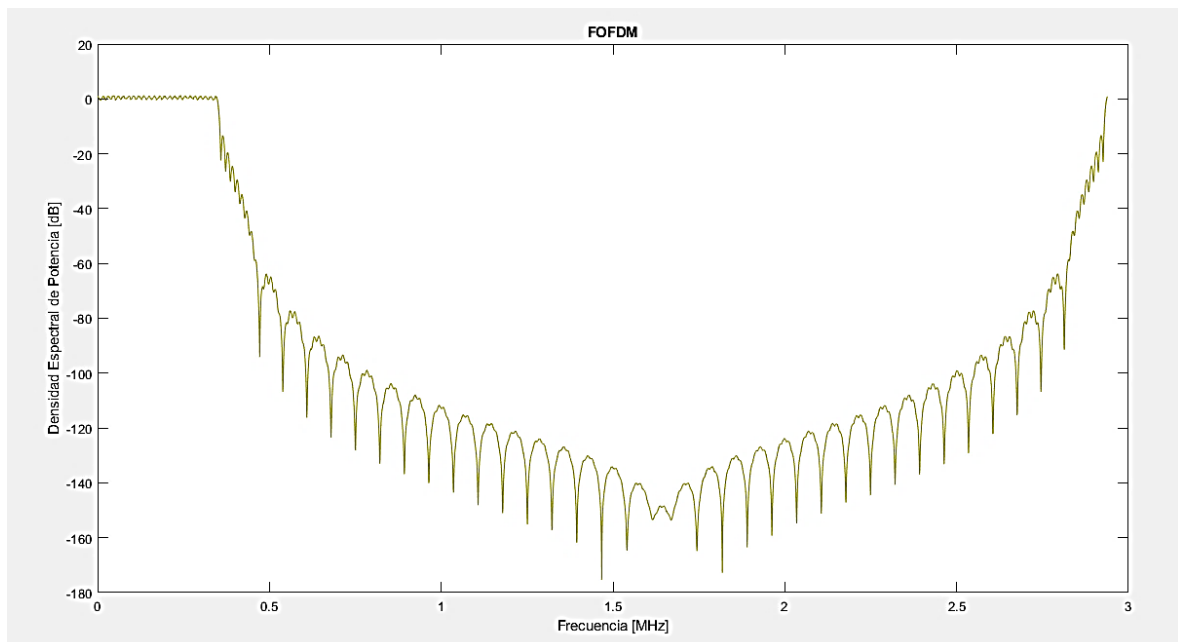


Figura 28: Densidad Espectral de Potencia f-OFDM

Fuente: El Autor

Al igual que FBMC f-OFDM presentan pérdidas considerables debido al uso de filtros en el caso de f-OFDM casi llega al punto de los -180dB como se observa en la figura 28 debido a que en lugar de usar un banco de filtros común para todas las portadoras f-OFDM pretende asignar un filtro específico para cada portadora, esto sin duda genera un mayor conste en el espectro, pero generaría ondas casi perfectas para aplicaciones específicas, es decir la canalización del espectro se enfoca en el uso que cada portadora tendría como se puede observar en la figura 5.

6.1.5. UFMC

Número de sub portadoras = 24

Numero de símbolos en el tiempo = 14

Espacio entre subportadoras = 15Khz

Prefijo Cíclico = 0

Longitud filtro de transmisión = $1/14 * (1/\text{espacio entre subportadoras})$

Longitud filtro de recepción = $1/14 * (1/\text{espacio entre subportadoras})$

Longitud de prefijo filtrado cíclico = $1 / (14 \text{ espacio entre subportadoras})$

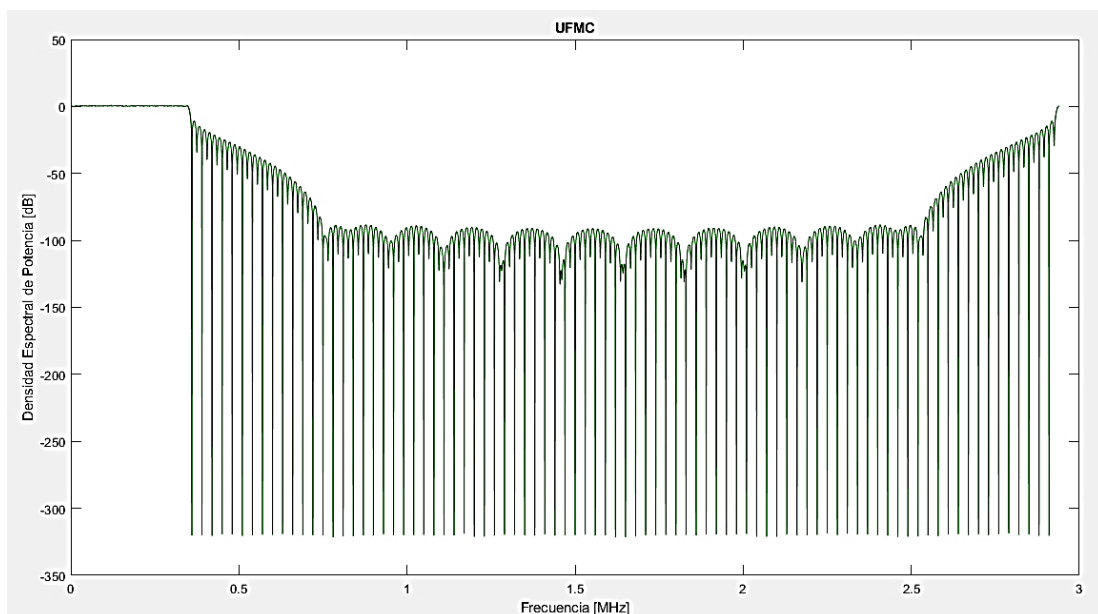


Figura 29: Densidad Espectral de Potencia UFMC

Fuente: El Autor

UFMC presenta pérdidas de casi -350dB como muestra la figura 28 claramente desde este análisis UFMC será el primer esquema de modulación que sea descartado camino a 5G, este comportamiento se debe que UFMC pretende que con un gran filtro (filtro universal) poder combatir todas las interferencias, errores y posibles ruidos que se puedan generar en la transmisión de información de todas las portadoras, a pesar de que desde el punto de vista de costes e implementación UFMC sería una opción viable es fácil darse cuenta solamente al observar esta grafica que su comportamiento no es el más óptimo para el aprovechamiento del espectro electromagnético, siendo este parámetro esencial para el despliegue de 5G.

6.2. Consumo Energético

Para la evaluación del consumo energético de cada esquema de modulación se mide la potencia de la señal en relación al tiempo este resultado nos brindara la información necesaria para saber cuál es el esquema que menor potencia consume, es decir el que se adapta mejor a los requerimientos de 5G. Al final del apartado podremos observar la unión de todos los consumos en una sola grafica para poder apreciar sus diferencias, además de una simulación de la potencia óptima para observar la diferencia entre un consumo real simulado y un consumo óptimo. Cabe recalcar que los parámetros para la simulación de consumo energético son los mismos que en el apartado 6.1 por lo tanto se sobre entienden.

6.2.1. FBMC

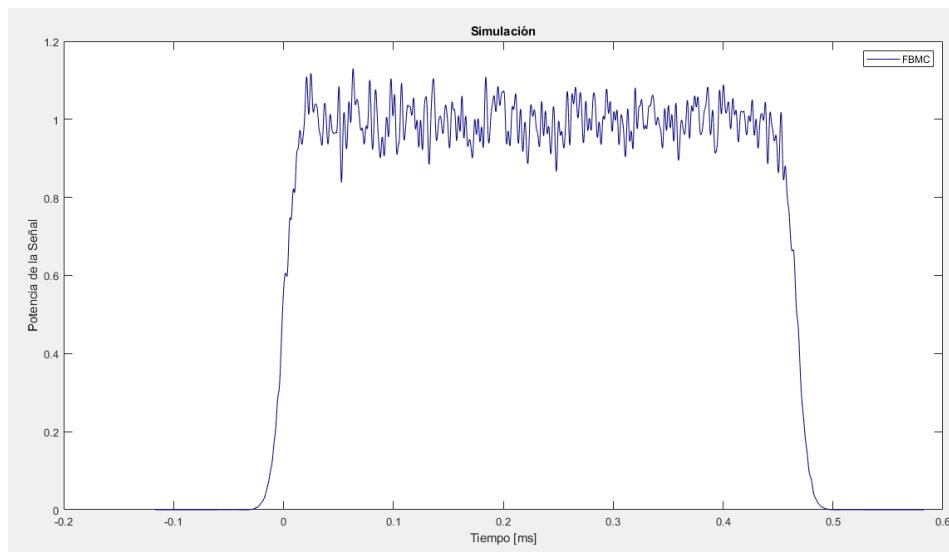


Figura 30: Consumo de energía FBMC

Fuente: El Autor

En la figura 30 de consumo de energía de FBMC al igual que las gráficas posteriores el consumo se ha normalizado es decir es referente a la unidad para de esta forma hacer un poco más sencillo el análisis de los resultados, ya que en el entorno de simulación sería más complejo tratar el tema de potencia debido a: la distancia entre transmisor receptor, los circuitos electrónicos de cada esquema y las condiciones de transmisión (clima, interferencias, velocidad, entre otras)

FBMC ocupa un conjunto de filtros luego de realizar la transformada de Fourier y antes de que las señales se transmitan por las antenas, este banco de filtros tiene un consumo moderado de potencia como podemos ver en la figura 30 ya que la señal oscila en torno a la unidad sin tener picos de consumo muy elevados, entonces podemos decir que FBMC tiene un consumo de potencia dentro de un rango normal.

6.2.2. OFDM

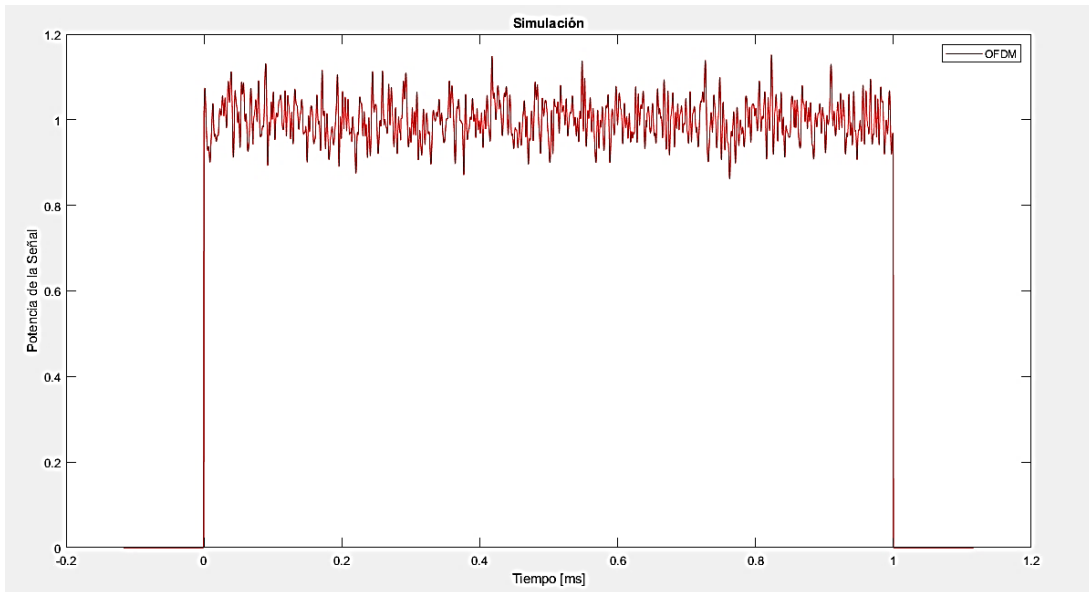


Figura 31: Consumo de energía OFDM

Fuente: El Autor

En la figura 31 el análisis de OFDM es muy similar al de FBMC a pesar de que OFDM en su estructura ocupa el prefijo cíclico en lugar de un banco de filtros, su consumo de potencia también lo podríamos considerar dentro del rango normal.

6.2.3. WOLA

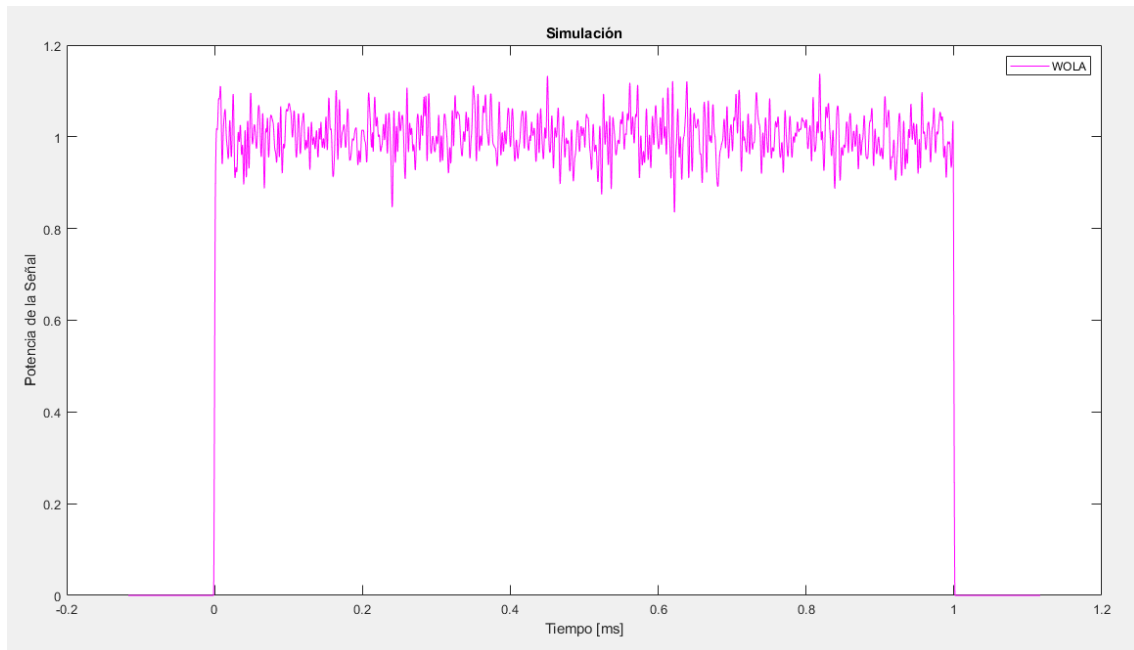


Figura 32: Consumo de Potencia WOLA

Fuente: El Autor

En la figura 32 el consumo de energía en WOLA es de comportamiento muy similar a OFDM ya que las ventanas de transmisión que ocupa WOLA presentan un consumo de potencia muy bajo de la misma forma que ocurre con el prefijo cíclico en OFDM por lo tanto también podríamos decir que WOLA presenta un consumo de energía en el rango de lo normal.

6.2.4. f-OFDM

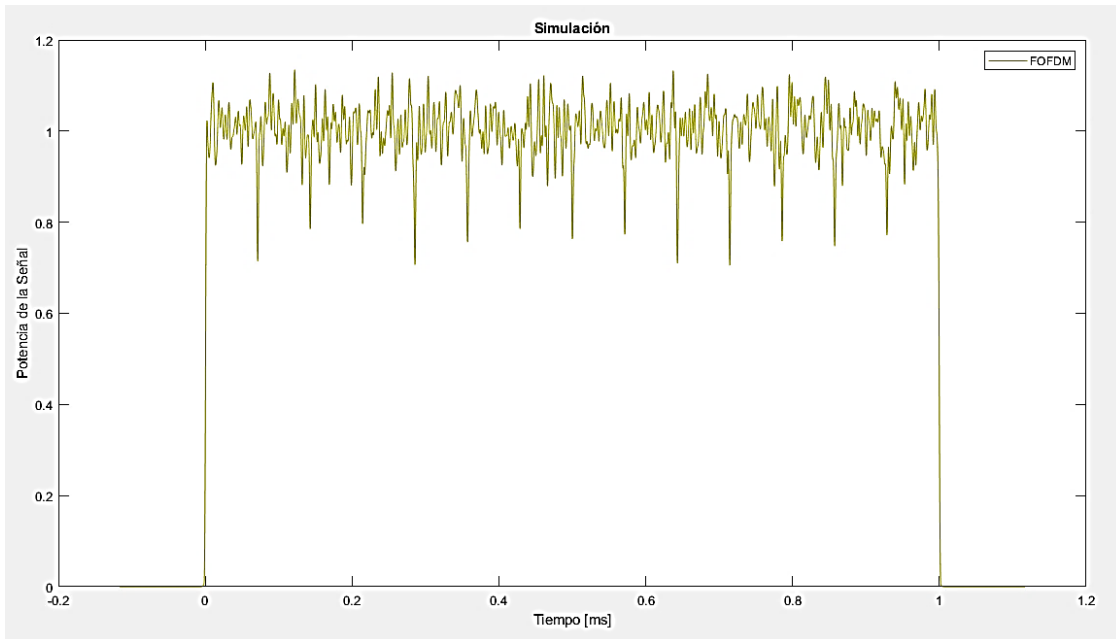


Figura 33: Consumo de Potencia f-OFDM

Fuente: El Autor

En la figura 33 el análisis de f-OFDM nos brinda un comportamiento distinto a WOLA FBMC y OFDM, de hecho es fácil distinguir los 14 símbolos transmitidos, esto se debe a la estructura que compone f-OFDM, estructura basada en un filtro para cada portadora, además podemos ver como el consumo de la señal decae casi hasta un 70% del total de potencia en cada cambio de símbolo debido a que el filtro se activa y se desactiva por cada símbolo, es evidente que este comportamiento es imperceptible al momento de hacer uso de las comunicaciones ya que como podemos observar se realiza en el orden de los mili segundos. Entonces debido a lo observado en la figura 33 se puede decir que f-OFDM presenta un comportamiento diferenciado de los esquemas de modulación anteriores, pero no precisamente por el consumo neto de energía sino más bien por el funcionamiento del esquema.

6.2.5. UFMC

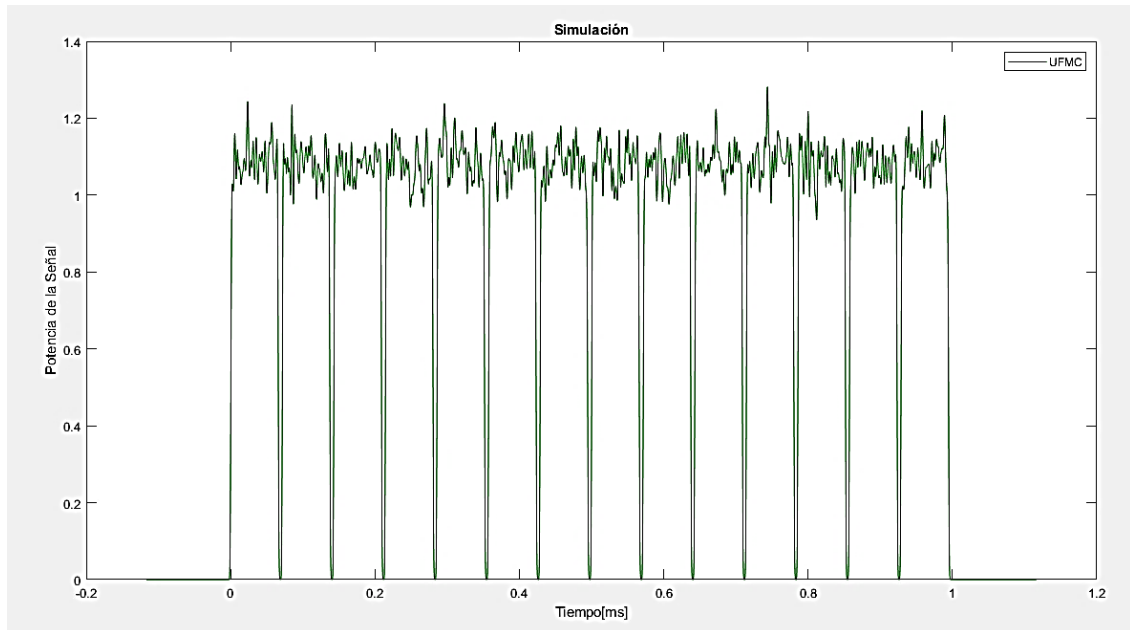


Figura 34: Consumo de Potencia UFMC

Fuente: El Autor

El consumo de energía de UFMC es claramente diferente a los demás esquemas ya que se distingue uno por uno los símbolos enviados como se observa en la figura 34, además si observamos la gráfica por la parte superior vemos que no hay picos de consumo de energía muy elevados, es decir en este apartado UFMC estaría por delante de los demás esquemas ya que presenta un consumo de energía relativamente menor a los esquemas de modulación anteriores, esto debido a que utiliza un filtro universal para el tratamiento de todas las portadoras sin distinción de trama y sin la utilización de prefijo cíclico, a pesar de que en la gráfica se observa que el consumo es relativamente menor el proceso de encender y apagar el filtro para el siguiente símbolo es muy brusco a diferencia de f-OFDM que solo decae hasta un 70% del total de potencia, en UFMC la señal cae en su totalidad pudiendo ser este motivo de retardos en la transmisión.

6.3. Retardo de transmisión

Para evaluar el retardo de transmisión que cada esquema presenta se utiliza el error de bit como referencia, es decir a más bits errados la comunicación tendrá un mayor retardo debido a que mientras mayor sea la tasa de error de bits mayor será el tiempo de recuperación de información por los algoritmos de corrección de errores en el proceso de demodulación de la señal. Cabe recalcar al igual que en el apartado anterior que los parámetros de evaluación son los mismos que en la densidad espectral de potencia.

En este caso específico se ha integrado todas las gráficas de error de bit en una misma figura ya que específicamente en este apartado la diferencia de un esquema respecto de otro esquema de modulación es muy poco notoria, es más se ha hecho un acercamiento que se presenta en la figura 36 donde se puede observar de mejor manera estas diferencias.

6.3.1. Relación de error de bit

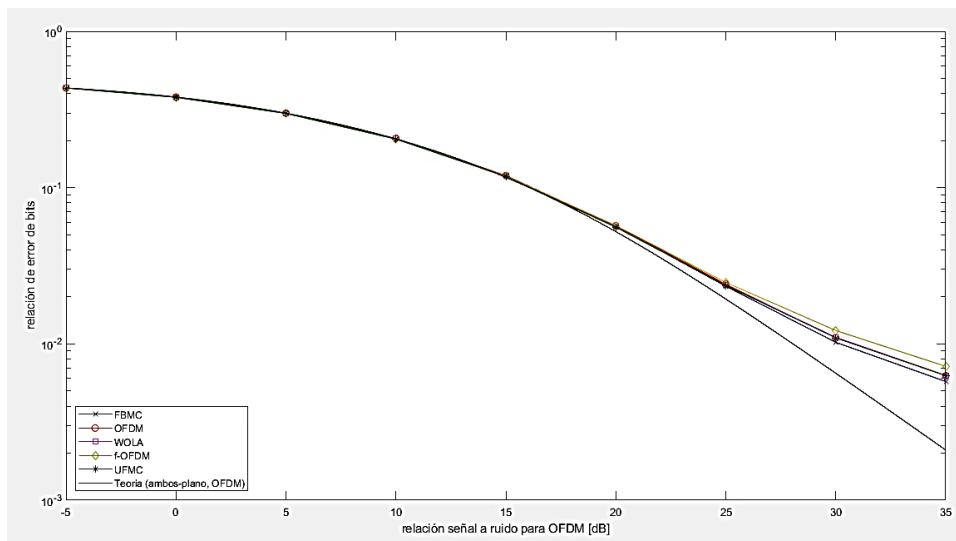


Figura 35: Relación de error de bit

Fuente: El Autor

En la figura 35 se puede observar que el comportamiento de todos los esquemas en este apartado es muy similar por tal motivo se ha creado un acercamiento que se presenta en

la figura 36 donde se puede observar de mejor manera cual esquema tiene mayor ratio de error de bit

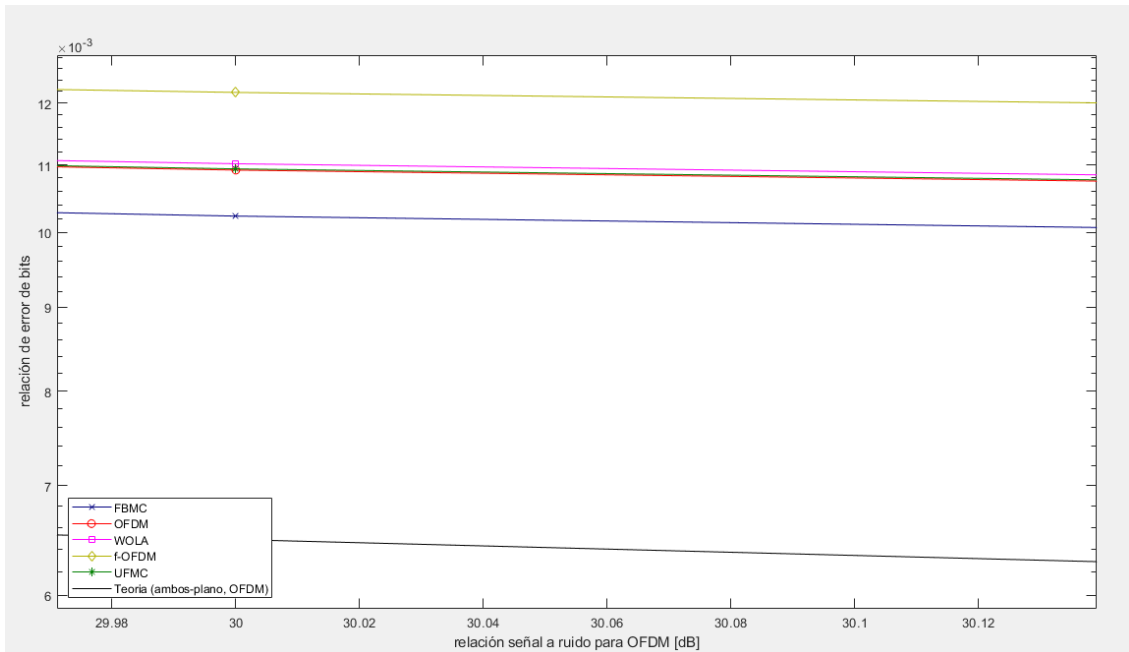


Figura 36: Acercamiento relación de error de bits

Fuente: El Autor

La figura 36 observamos el comportamiento de la relación de error de bit. FBMC es el esquema que más se acerca al óptimo con un error de bit de alrededor de 10.2×10^{-3} este comportamiento se debe a que junto con el filtro prototipo escogido (Hermite en este caso) se escoge una modulación digital que igual tiene un orden de 64QAM pero se la denomina OQAM u *offset QAM* que no hace más que eliminar los bits de paridad que no están en uso dentro de la modulación digital y de esta manera poder filtrar la señal con un aparente menor número de símbolos (aparente por que los símbolos siempre son los mismos solo que en este caso se elimina las paridades no utilizadas y hace que se envíen tramas de información un poco más cortas).

Por encima de FBMC con un comportamiento muy similar se observa a WOLA OFDM y UFMC cuyo error de bit oscila por alrededor de 11×10^{-3} bits errados, este

comportamiento casi idéntico nos da a entender que el proceso de modulación digital es muy parecido y en efecto es así ya que tanto en OFDM como en WOLA se añaden prefijo ciclos como ventanas de transmisión respectivamente, este proceso agrega bits a la cadena de transmisión de símbolos haciendo un poco más compleja la recepción de datos, a pesar de que el objetivo de agregar estos bits es para corregir errores el receptor los trata de la misma manera que trata a los demás bits de la trama por ende gráficamente también podrían ser parte de la relación de error de bits, en el caso de UFMC se puede decir que el error de bit en este esquema es el esperado ya que el hecho de que todas las portadoras pasen por el mismo filtro hace que el comportamiento sea muy similar a OFDM donde a todas las tramas de información se les agrega ceros al inicio de la trama para evitar la aparición de errores en el receptor, además UFMC no presenta ningún esquema diferenciador para la corrección del retardo de transmisión ya que este esquema presenta un enfoque un poco más enfocado la conservación de potencia.

Por último el esquema que presenta una mayor tasa de bits errados y por ende un mayor retardo en la transmisión es f-OFDM este comportamiento se debe sin duda a que los diferentes filtros que se pretende utilizar f-OFDM en la transmisión generaría un procesamiento mayor en la señal, es decir añade un bloque más al esquema de transmisión y este mismo bloque debe ser añadido en el receptor haciendo que se genere un mayor retardo en la transmisión, retardo que en la figura se observa como una mayor tasa de error de bits.

6.4. NOMA vs OFDMA

Todos los esquemas simulados y evaluados anteriormente presentan comportamientos ortogonales y la simulación que se ejecuta corresponde al proceso entero de modulación, es decir que las figuras anteriormente evaluadas nos servirán también para evaluar el comportamiento del acceso múltiple, pero camino a 5G también se propone una nueva tecnología de acceso al medio que son los sistemas no ortogonales en sus distintas variantes las cuales se detallan de manera general en el apartado 4.3.3, la tecnología de

acceso al medio que está actualmente en uso para 4G *LTE-Advanced* es la variante de acceso múltiple de OFDM denominada OFDMA por tal motivo las siguiente gráficas generan una comparativa entre NOMA y OFDMA para poder observar la diferenciación de los sistemas no ortogonales con los ortogonales, cabe recalcar que en estas graficas al tratarse de acceso múltiple se hace referencia al aprovechamiento del espectro que tienen los usuarios.

Para las figuras se asigna los siguientes datos

Ganancia usuario 1 = 10

Guanacia usuario 2 = 10

Conteo = 1

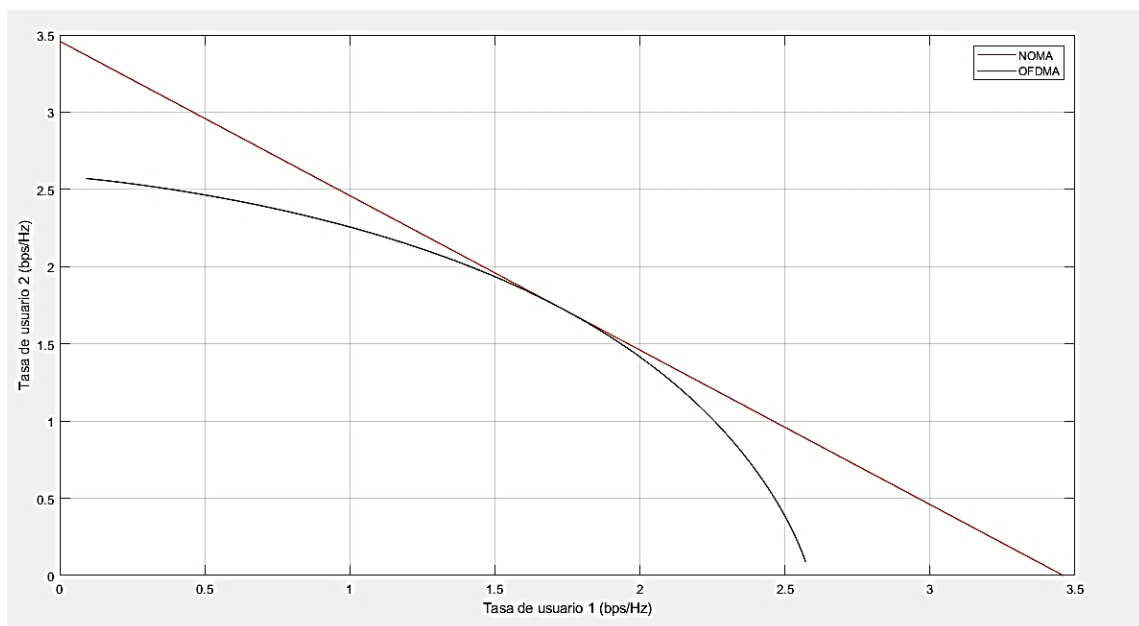


Figura 37: Taza de bits por usuario

Fuente: El Autor

En la figura 37 podemos ver la diferencia que presenta el acceso múltiple ortogonal OFDMA y NOMA acceso múltiple no ortogonal, se puede observar un mejor uso del espectro de NOMA ya que para los dos usuarios con este esquema de acceso múltiple se les asigna una mejor relación de bps/Hz, siempre y cuando sea en ambientes simétricos, es decir que los parámetros para la transmisión de ambos usuarios presenten

características similares. En este caso la curva que presenta NOMA es aparentemente más acertada en cuestión de asignación de recursos, pero el gran problema que presenta NOMA es la implementación en ambientes no simétricos ya que requiere una sobrecarga de procesamiento para determinados usuarios debido a su posición geográfica, los niveles de potencias de trasmisión o la codificación que sea utilizada.

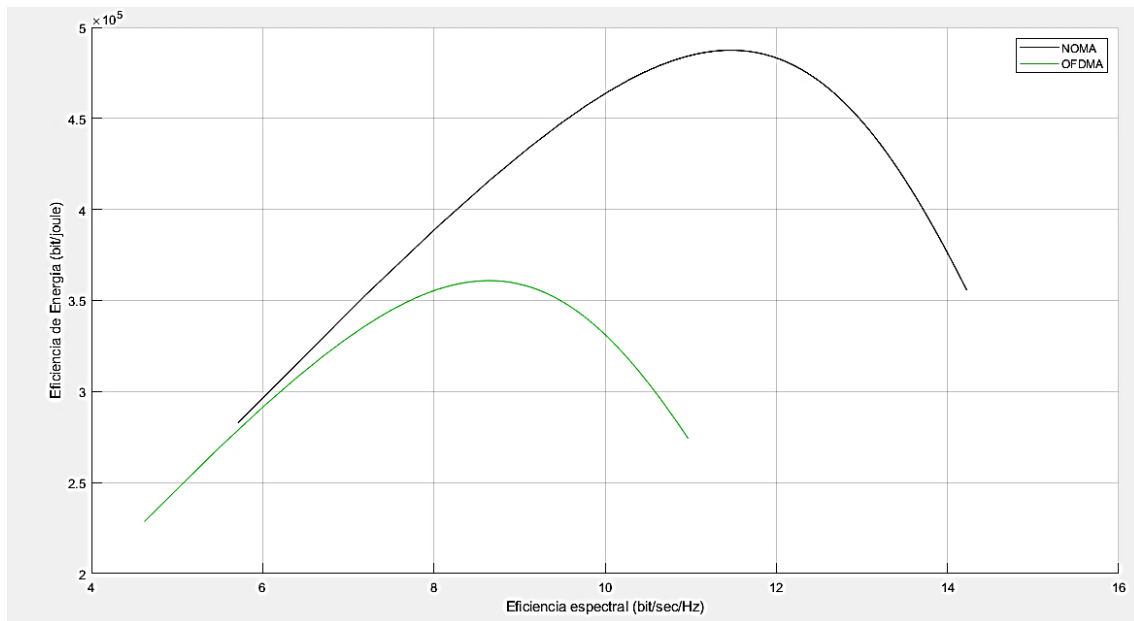


Figura 38: Eficiencia NOMA vs OFDMA

Fuente: El Autor

En la figura 38 podemos observar dos parámetros esenciales para poder evaluar el acceso múltiple, una comparación entre la eficiencia espectral y la eficacia de energía.

Podemos observar cómo NOMA presenta un aparente mejor comportamiento en cuanto a eficiencia espectral, ya que nos brinda más bits por Hz, en cuanto a eficiencia energética la curva nos dice que el consumo de energía se incrementa para ciertos usuarios, debido a la mayor carga de procesamiento que sufriría la demodulación de la información para poder ser presentada al usuario.

OFDMA por lo contrario es claramente más eficiente en cuanto a el consumo de energía en la gráfica se puede ver como no sobrepasa los 4 bits/ joule para el mismo escenario

que ocupa NOMA, por consiguiente, OFDMA aprovecha mejor los recursos de espectro y energía para la transmisión de información.

6.5. TABLA COMPRATIVA DE ESQUEMAS DE MODULACIÓN

Tabla 5: Comparativa de Modulación

Formas de Onda	Pros	Contras
f-OFDM	Tratamiento de cada portadora de forma independiente, asignación de un filtro por portadora, mejor asignación de los recursos de radio debido a canalización más eficiente del espectro	Carga de procesamiento elevada debido a la etapa de filtrado tanto en transmisor como en receptor.
CP-OFDM	Asignación flexible de frecuencia Fácil integración con MIMO	Alta Relación de fuga de canal adyacente (ACLR) lóbulo lateral decae lentamente Necesidad de multiplexación síncrona
CP-OFDM con WOLA	Todos los pros de CP-OFDM Mejor supresión de OOB respecto a CP-OFDM Procesamiento simple	
UFMC	Mejor supresión de OOB respecto a CP-OFDM, pero no mejor que CP-OFDM con WOLA	ISI de desvanecimiento multitrayecto (sin CP) Mayor complejidad de Tx que CP-OFDM

		Mayor complejidad de Rx (tamaño 2 FFT) que CP-OFDM
FBMC	Mejor que CP-OFDM con WOLA (pero la mejora se reduce con la no linealidad de PA)	Alta complejidad Tx / Rx debido a OQAM (la forma de onda se sintetiza por RB) La integración con MIMO no es trivial. Sujeto a ISI bajo canales no planos

Fuente: El Autor

6.6. TABLA COMPARTIVA NOMA vs OFDMA

Tabla 6: Comparativa de acceso múltiple

Acceso múltiple	Pros	Contras
OFDMA	Sin interferencia intracelular mayor eficiencia espectral y compatibilidad con MIMO	Necesita multiplexación síncrona Pérdida de enlaces para gran cantidad de usuarios
NOMA	Permite multiplexación asíncrona.	No es adecuado para una mayor eficiencia espectral Requiere mayor procesamiento para receptor

Fuente: El Autor

7. DISCUSIÓN

La finalidad de la presente investigación fue realizar una indagación y evaluación de los esquemas de modulación y acceso múltiple que se presentan para 5G, llevar a cabo una comparación entre los esquemas y poder encontrar los pros y contras que existen entre cada esquema de modulación, con toda la información recabada en las fuentes bibliográficas y con ayuda de las simulaciones de graficas de eficiencia espectral, de potencia y de retardo.

Es posible observar que los diferentes esquemas de modulación presentan comportamientos muy similares al momento de modular la misma cantidad de información, de hecho, las claras diferencias entre todos los esquemas no es su resultado como tal, sino más bien el proceso antes de obtener resultados, es decir, por ejemplo, en el caso de *filter bank multicarrier* (FBMC) la utilización de un banco de filtros lo suficientemente robusto y eficaz para la modulación de cada portadora es la gran traba para su implementación ya que es demasiado costosa, otro ejemplo es f-OFDM que a pesar de que de igual manera nos brinda una forma de onda muy cercana a la onda sin pérdida, la implementación de esta tecnología supondría colocar un filtro antes de la salida a la antena por cada portadora que se moduló, además de un coste económico muy elevado, la carga computacional para el sistema se multiplica según el número de portadoras que se utilice.

Por consiguiente, OFDM, sigue siendo el candidato más factible para el despliegue de 5G debido a la compatibilidad con LTE y con su fácil implementación, OFDM en su versión CP-OFDM que no es más que detallar que se utiliza un Prefijo Cíclico en la creación de los símbolos de OFDM, para disminuir prácticamente a cero la interferencia Inter símbolo y también servir como soporte para la sincronización de señales y la corrección de errores, el acceso múltiple OFDMA hereda todos los beneficios de OFDM para la recepción de información, permite aprovechar todos los recursos de radio y brinda una optimo tratamiento de las interferencias.

El desarrollo del software de simulación utilizado en este documento para realizar la evaluación de los diferentes esquemas de modulación nos brinda la información necesaria como para determinar que OFDM será el candidato más óptimo para utilizar en la transmisión de información de 5G.

El uso de tecnologías no ortogonales (NOMA) como las que se detallan en el apartado 4.3.3. es una opción efectiva para el acceso múltiple, pero al tratarse de una tecnología relativamente nueva y aun no implementada de forma masiva su optima utilización seria en escenarios concretos donde sea posible aprovechar todas las bondades de este sistema sin generar un procesamiento computacional demasiado elevado.

Los objetivos del presente documento de investigación fueron la: indagación de los nuevos esquemas de modulación, la implementación en Matlab de los esquemas (el código del programa se presenta en ANEXOS), comparar las características de cada esquema y construir un instrumento de investigación que sirva de referencia sobre el estudio de modulaciones y acceso múltiple para 5G. La utilización de un sustento teórico confiable y la realización de una interfaz gráfica de parámetros editables de cada esquema, dan la pauta necesaria para hacer de este documento un marco de referencia sobre las tecnologías de modulación y acceso múltiple para 5G.

8. CONCLUSIONES

Como parte esencial de las comunicaciones inalámbricas el espectro electromagnético es un recurso limitado del cual se sirven muchas tecnologías para desplegar sus servicios. El aprovechamiento de este recurso es de gran importancia para poder desplegar una nueva generación de comunicaciones móviles, 5G está muy cerca de convertirse en realidad y por ello diferentes formas de onda candidatos se examinaron en este documento. Modulaciones ortogonales y no ortogonales se han barajado para para la transmisión de 5G, OFDM es la base de la mayor parte de sistemas de comunicación en la actualidad debido a su gran robustez y su fácil implementación, sin duda 5G hará uso de este esquema debido a la compatibilidad con antiguos sistemas y a su adaptabilidad con diferentes tecnologías, entre ellas MIMO, además OFDM presenta muchas variantes que pueden trabajar de forma conjunta como es el caso de WOLA un esquema basado en OFDM.

La mejor forma de poder comparar el comportamiento de un determinado sistema (en este caso las tecnologías de modulación candidatos a 5G) son graficas de los parámetros de interés (densidad espectral de potencia, consumo de potencia y retardo), al llevar a cabo las simulaciones mediante el uso del software de programación MATLAB en su versión 2018a fue posible implementar cada esquema y constatar el comportamiento de cada tecnología, de esta manera contrastar los resultados obtenidos. La implementación de cada esquema se logró gracias a la utilización de un programa base realizado por la Universidad de Viena donde se brinda las librerías necesarias para poder realizar simulaciones de espectro potencia y retardo, la creación de un GUIDE (interfaz de usuario) es un gran aporte para la edición de parámetros de simulación debido a su fácil utilización, de esta manera se obtiene distintas graficas que brindan la información necesaria para poder comparar su comportamiento.

Cada esquema de modulación que se ha propuesto para llevar la información a los usuarios de 5G están en la capacidad de poder cumplir con los objetivos que 5G se

propone, las verdaderas diferencias radican en la operación de cada tecnología, es decir, el resultado de cada simulación es relativamente igual, pero el proceso de obtención de resultados es diferente como se puede observar en la parte de resultados, donde el funcionamiento de los esquemas se dividen en dos grupos de comportamientos; uno de los cuales es el que depende de filtros para el aprovechamiento del espectro, encontramos a FBMC, f-OFDM y UFMC, mientras que los esquemas con mejores resultados en la simulación se basan en el enventanamiento y la utilización de prefijo cíclico, en este grupo podemos encontrar a WOLA y OFDM. Los sistemas no ortogonales han sonado fuertemente para adaptarse a los requerimientos de 5G debido a que realiza un gran aprovechamiento del espectro, presenta muchos pros en este sentido ya que es posible enviar más información a un número mayor de usuarios, el contra más grande que posee esta tecnología es su mayor carga de procesamiento en determinados escenarios y además la poca compatibilidad con algunos sistemas actuales, por tal motivo en 5G NOMA podría ser utilizado para aplicaciones o dispositivos específicos como la domótica donde la movilidad es limitada.

Las redes inalámbricas forman ya parte de nuestro entorno, próximamente 5G será una realidad, la densificación de esta red dependerá no solo de un mejor uso del espectro electromagnético sino de diversos factores como: el costo de implementación de la red, también del conflicto social que genera tener una vida totalmente conectada y dependiente del internet ya que es la verdadera finalidad de 5G.

5G se convertirá en una red Global formada por un conjunto de redes Heterogéneas que se adapten a los requerimientos específicos de cada dispositivo. Este proyecto de investigación brinda un marco de referencia sobre la utilización del espectro para redes inalámbricas de quinta generación, se presenta la información necesaria sobre cada esquema que se propone para 5G, basados en los grandes grupos de investigación a nivel mundial tales como METIS, 5GPP, 5GForum, 5GNOW, además de utilizar documentos de investigación realizados por empresas tales como NOKIA, Ericsson, Qualcomm entre otras. Además del sustento teórico se brinda una interfaz de usuario para la simulación de parámetros pertenecientes a cada esquema de modulación

9. RECOMENDACIONES

Al realizar un trabajo de investigación sobre temas en desarrollo, como es el caso de 5G, es de gran importancia utilizar fuentes de información lo más actuales posibles para poder tener un sustento teórico confiable que brinde datos relevantes para el investigador.

Para el desarrollo del software en este caso se utilizó un computador portátil DELL con un procesador core i5 que presenta una velocidad de procesamiento de 1,7Ghz, este computador fue utilizado exclusivamente para el desarrollo de este proyecto, el programa de simulación se lo desarrollo en MATLAB en su versión 2018a con las librerías *LTE toolbox 5G* instaladas, dentro de Matlab se realizó una interfaz de usuario GUIDE ya que facilita el procesamiento de la información y nos brindara más seguridad al momento de realizar pruebas. El computador utilizado en el proyecto a la fecha pertenece a una gama media-baja y la carga que genera el programa hace que este solo pueda realizar esta tarea exclusivamente. El desarrollo de un software de simulación que modula y demodula varios esquemas presenta un gran consumo de procesamiento de la maquina por tal motivo es recomendable el uso de un computador de gama alta con un procesador i 9 de última generación o un i7 de octava generación con por lo menos una memoria RAM de 16GB, que trabajen a velocidad superior a 2,5Ghz, este tipo de equipos podrán cumplir de manera óptima con las exigencias del programa.

Debido a que el desarrollo de un proyecto de tesis comprende parámetros investigativos y de formato es recomendable para el investigador asesorar su proyecto con docentes con más conocimiento y experiencia para que el desarrollo de la investigación sea más llevadero y que los resultados sean el relejo del esfuerzo invertido.

10. BIBLIOGRAFÍA

@article{Vienna5GSLS, title = {Flexible multi-node simulation of cellular mobile communications: the Vienna 5G System Level Simulator}}, author = {Martin Klaus Müller and Fjolla Ademaj and Thomas Dittrich and Agnes Fastenbauer and Blanca Ramos Elbal and Armand Nabavi and Lukas Nagel and Stefan Schwarz and Markus Rupp}, journal = {EURASIP Journal on Wireless Communications and Networking}, year = {2018}, month = sep, volume = {2018}, number = {1}, pages = {17}, doi = {10.1186/s13638-018-1238-7} }

5G Mobile Wireless Technology, [en línea], (2019), disponible en: [<https://www.electronics-notes.com/articles/connectivity/5g-mobile-wireless-cellular/waveform-optimised-ofdm.php>]

Amitabha Ghosh Head of Small Cell Research, Nokia Fellow, IEEE Fellow Nokia Bell Labs, 5G New Radio Design Expanding the human possibilities of technology to make our lives better Fall VTC-2017, Panel September 25th , 2017

Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C., & Zhang, J. C. (2014). What will 5G be?. *IEEE Journal on selected areas in communications*, 32(6), 1065-1082.

Bochechka, G., Tikhvinskiy, V., Vorozhishchev, I., Aitmagambetov, A., & Nurgozhin, B. (2017, June). Comparative analysis of UFMC technology in 5G networks. In 2017 International Siberian Conference on Control and Communications (SIBCON) (pp. 1-6). IEEE.

Cai, Y., Qin, Z., Cui, F., Li, G. Y., & McCann, J. A. (2018). Modulation and multiple access for 5G networks. *IEEE Communications Surveys & Tutorials*, 20(1), 629-646.

Farhang-Boroujeny, B., & Moradi, H. (2016). OFDM inspired waveforms for 5G. *IEEE Communications Surveys & Tutorials*, 18(4), 2474-2492.

Gerzaguet, R., Ktésas, D., Cassiau, N., & Doré, J. B. (2016, January). Comparative study of 5G waveform candidates for below 6 GHz air interface. In *Proc. ETSI Workshop Future Radio Technol. Focusing Air Interface* (pp. 1-16).

Islam, S. M., Zeng, M., & Dobre, O. A. (2017). NOMA in 5G systems: Exciting possibilities for enhancing spectral efficiency. *arXiv preprint arXiv:1706.08215*.

Kasparick, M., Wunder, G., Jung, P., & Maryopi, D. (2014, May). Bi-orthogonal waveforms for 5G random access with short message support. In *European Wireless 2014; 20th European Wireless Conference* (pp. 1-6). VDE.

Larsson, E. G., Edfors, O., Tufvesson, F., & Marzetta, T. L. (2014). Massive MIMO for next generation wireless systems. *IEEE communications magazine*, 52(2), 186-195.

Li, C., Wang, M., Chi, T., Kumar, A., Boenke, M., Cahoon, N., ... & Wang, H. (2017, March). A high-efficiency 5G K/Ka-band stacked power amplifier in 45nm CMOS SOI process supporting 9Gb/s 64-QAM modulation with 22.4% average PAE. In *2017 Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS)* (pp. 1-4). IEEE.

Martinez, E. O., & Agudelo, L. B. (2013). FBMC, una opción para un uso óptimo del espectro electromagnético. *Revista en telecomunicaciones e informática*, 3(5), 39-53.

McCann, J. A., Qin, Z., Cai, Y., Cui, F., & Li, G. Y. (2018). Modulation and multiple access for 5G Networks. Institute of Electrical and Electronics Engineers.

Mitra, R. N., & Agrawal, D. P. (2015). 5G mobile technology: A survey. *ICT Express*, 1(3), 132-137.

Pérez Mantilla Susana, (2019) El sistema de comunicaciones móviles de próxima generación 5G y su caso de uso IoT, Universidad Oberta de Cataluña.

Qualcomm. (2016). Making 5G NR a reality: leading the technology inventions for a unified, more capable 5G air interface. White paper.

Riazul Islam, S. M., Zeng, M., & Dobre, O. A. (2017). NOMA in 5G Systems: Exciting Possibilities for Enhancing Spectral Efficiency. arXiv preprint arXiv:1706.08215.

Rodríguez Fernández, P. (2017). Optimización de la gestión de recursos radio en redes 5G.

Sahin, A., Güven, I., & Arslan, H. (2012). A Survey on Prototype Filter Design for Filter Bank Based Multicarrier Communications. arXiv preprint arXiv:1212.3374

Schaich, F., & Wild, T. (2014, May). Waveform contenders for 5G—OFDM vs. FBMC vs. UFMC. In 2014 6th international symposium on communications, control and signal processing (ISCCSP) (pp. 457-460). IEEE.

Srikanth, S., Pandian, P. M., & Fernando, X. (2012). Orthogonal frequency division multiple access in WiMAX and LTE: a comparison. *IEEE Communications Magazine*, 50(9), 153-161.

Velazquez-Gutierrez, J. M., & Vargas-Rosales, C. (2017). Sequence sets in wireless communication systems: A survey. *IEEE Communications Surveys & Tutorials*, 19(2), 1225-1248.

Wang, B., Wang, K., Lu, Z., Xie, T., & Quan, J. (2015, June). Comparison study of non-orthogonal multiple access schemes for 5G. In *Broadband Multimedia Systems and Broadcasting (BMSB), 2015 IEEE International Symposium on* (pp. 1-5). IEEE.

Zayani, R., Medjahdi, Y., Shaiek, H., & Roviras, D. (2016, December). WOLA-OFDM: A potential candidate for asynchronous 5G. In *2016 IEEE Globecom Workshops (GC Wkshps)* (pp. 1-5). IEEE.

Zhang, X., Jia, M., Chen, L., Ma, J., & Qiu, J. (2015, December). Filtered-OFDM-enabler for flexible waveform in the 5th generation cellular networks. In *2015 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE.

Zyren, J., & McCoy, W. (2017). Overview of the 3GPP long term evolution physical layer. Freescale Semiconductor, Inc., white paper, 7, 2-22.

LISTA DE SIGLAS

AMC (adaptive modulation and coding) modulación y codificación adaptativa	GFDM (Generalized Frequency Division Multiplexing) Multiplexación de división de frecuencia generalizada
BS (base station) estación base	IDFT (Inverse Discrete Fourier Transform) transformada discreta de Fourier inversa
CP (cyclic prefix) Prefijo cíclico	IFFT (Inverse fast Fourier transform) Transformada inversa de Fourier
DFT (discrete Fourier transform) transformada discreta de Fourier	IoT (internet of things), Internet de las cosas
eNB (E-UTRAN Node B) E-UTRAN Node B	LTE (Long Term evolution) Evolución a largo plazo
E-UTRA (Evolved Universal Terrestrial Radio Access) acceso universal de radio terrestre evolucionado.	MCM (multicarrier modulation) modulación multiportadora
FBMC (Filter Bank Multi carrier) Banco de filtro multiportadora	MTC (Machine Type Communications) Comunicaciones tipo maquina
FEC (Forward Error Correction) corrección de errores hacia adelante	ng-eNB (Next generation eNodeB) siguiente generación eNodeB
F-OFDM (filtered-OFDM) OFDM filtrado	NOMA (non orthogonal multiple access) Acceso múltiple no ortogonal

NR (New Radio) Nueva radio	PUCCH (UL control channel) canal al de control
OFDM (Orthogonal frequency división multiplexing) Multiplexación por división de frecuencia ortogonal	PUSCH (UL shared channel) Canal compartido UL
OFDMA (Orthogonal Frequency-Division Multiple Access) Acceso múltiple por división ortogonal de frecuencia	RAN (Radio access network) Red de acceso de radio
PBCH (Broadcast channel) Canal de difusión	RRM (Management of radio resources) administración de recursos de radio
PDCCH (DL control channel) control de canal DL	SC-FDMA (Single Carrier Frequency multiple access) Acceso múltiple por división de frecuencia de portadora única
PDSCH (DL shared channel) canal compartido DL	UE (user equipment) equipo de usuario
PRACH (Random access channel) canal de acceso aleatorio	Wimax (Worldwide Interoperability for Microwave Access) interoperabilidad mundial para acceso por microondas
PRB (Physical resource blocks) bloque de recursos físicos	WOLA (Weighted Overlap and Add) Superposición ponderada

11. ANEXOS

11.1. CÓDIGO DE PROGRAMA GUIDE UI_5G

Para una mejor diferenciación del proceso de programación se a encerrado en cuadros de color amarillo la parte de creación de variables del GUIDE, en estas funciones de Matlab se obtiene los datos de los parámetros de simulación. Se a encerrado de color azul las variables correspondientes a parámetros específicos de cada esquema de modulación y del canal de comunicaciones, el código encerrado en estos cuadros corresponde al aporte del investigador y es complemento al proyecto de la universidad de Viena que comprende el resto del código tomado de su proyecto para 5G.

```
function varargout = UI_5G(varargin)
% UI_5G MATLAB code for UI_5G.fig
%   UI_5G, by itself, creates a new UI_5G or raises the existing
%   singleton*.
%
%   H = UI_5G returns the handle to a new UI_5G or the handle to
%   the existing singleton*.
%
%   UI_5G('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in UI_5G.M with the given input
arguments.
%
%   UI_5G('Property','Value',...) creates a new UI_5G or raises the
%   existing singleton*. Starting from the left, property value
pairs are
%   applied to the GUI before UI_5G_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property
application
%   stop. All inputs are passed to UI_5G_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help UI_5G

% Last Modified by GUIDE v2.5 31-Jan-2019 14:12:05

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @UI_5G_OpeningFcn, ...
                  'gui_OutputFcn',  @UI_5G_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```

end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before UI_5G is made visible.
function UI_5G_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to UI_5G (see VARARGIN)

% limpiamos;
%%clear;
%%cd('..');
%%addpath('./Theory');

addpath('./Theory');
addpath('./+Modulation');
addpath('./+Channel');

%Variables de: Parametros de Simulacion
global Simulation_SNR_OFDM_dB
global Simulation_MonteCarloRepetitions
global Simulation_PlotSignalPowerTheory

%Variables de: Parametros de Canal
global Channel_Velocity_kmh
global Channel_PowerDelayProfile
global Channel_DopplerModel
global Channel_CarrierFrequency

%Variables de: Parametros de Modulacion QAM
global QAM_ModulationOrder

%Variables de: FBMC
global FBMC_NumberOfSubcarriers
global FBMC_NumberOfSymbolsInTime
global FBMC_SubcarrierSpacing
global FBMC_PrototypeFilter
global FBMC_OverlappingFactor

%Variables de: OFDM
global OFDM_NumberOfSubcarriers
global OFDM_NumberOfSymbolsInTime
global OFDM_SubcarrierSpacing
global OFDM_CyclicPrefixLength

```

```

%Variables de: WOLA
global WOLA_NumberOfSubcarriers
global WOLA_NumberOfSymbolsInTime
global WOLA_SubcarrierSpacing
global WOLA_CyclicPrefixLength
global WOLA_WindowLengthTX
global WOLA_WindowLengthRX

%Variables de: FOFDM
global FOFDM_NumberOfSubcarriers
global FOFDM_NumberOfSymbolsInTime
global FOFDM_SubcarrierSpacing
global FOFDM_CyclicPrefixLength
global FOFDM_FilterLengthTX
global FOFDM_FilterLengthRX
global FOFDM_FilterCyclicPrefixLength

%Variables de: UFMC
global UFMC_NumberOfSubcarriers
global UFMC_NumberOfSymbolsInTime
global UFMC_SubcarrierSpacing
global UFMC_CyclicPrefixLength
global UFMC_FilterLengthTX
global UFMC_FilterLengthRX
global UFMC_FilterCyclicPrefixLength
global UFMC_ZeroPaddingInsteadOfCP

%Variables para Graficacion INTERNAS
global SamplingRate
global dt

"SubcarrierSpacing*NumberOfSubcarriers".
SamplingRate      = 15e3*14*14;
dt                = 1/SamplingRate;

% cambiar salidas por default para UI_5G
handles.output = hObject;

% actualizar estructura handles
guidata(hObject, handles);

% UIWAIT makes UI_5G wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = UI_5G_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

```

```

function et_modulacionOrdenQAM_Callback(hObject, eventdata, handles)
% hObject    handle to et_modulacionOrdenQAM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_modulacionOrdenQAM as text
%         str2double(get(hObject,'String')) returns contents of
et_modulacionOrdenQAM as a double

% --- Executes during object creation, after setting all properties.
function et_modulacionOrdenQAM_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_modulacionOrdenQAM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_canalVelocidad_Callback(hObject, eventdata, handles)
% hObject    handle to et_canalVelocidad (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_canalVelocidad
as text
%         str2double(get(hObject,'String')) returns contents of
et_canalVelocidad as a double

% --- Executes during object creation, after setting all properties.
function et_canalVelocidad_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_canalVelocidad (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```



```

function et_canalPerfilRetraso_Callback(hObject, eventdata, handles)
% hObject    handle to et_canalPerfilRetraso (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_canalPerfilRetraso as text
%         str2double(get(hObject,'String')) returns contents of
et_canalPerfilRetraso as a double

% --- Executes during object creation, after setting all properties.
function et_canalPerfilRetraso_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_canalPerfilRetraso (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_canalModDoppler_Callback(hObject, eventdata, handles)
% hObject    handle to et_canalModDoppler (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_canalModDoppler
as text
%         str2double(get(hObject,'String')) returns contents of
et_canalModDoppler as a double

% --- Executes during object creation, after setting all properties.
function et_canalModDoppler_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_canalModDoppler (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_canalFreqCarga_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to et_canalFreqCarga (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_canalFreqCarga
as text
%          str2double(get(hObject,'String')) returns contents of
et_canalFreqCarga as a double

% --- Executes during object creation, after setting all properties.
function et_canalFreqCarga_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_canalFreqCarga (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_simulacionRepMonteCarlo_Callback(hObject, eventdata,
handles)
% hObject    handle to et_simulacionRepMonteCarlo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_simulacionRepMonteCarlo as text
%          str2double(get(hObject,'String')) returns contents of
et_simulacionRepMonteCarlo as a double

% --- Executes during object creation, after setting all properties.
function et_simulacionRepMonteCarlo_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_simulacionRepMonteCarlo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_simulacionSNRlerArg_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to et_simulacionSNR1erArg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_simulacionSNR1erArg as text
%         str2double(get(hObject,'String')) returns contents of
et_simulacionSNR1erArg as a double

% --- Executes during object creation, after setting all properties.
function et_simulacionSNR1erArg_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_simulacionSNR1erArg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_simulacionSNR2doArg_Callback(hObject, eventdata, handles)
% hObject    handle to et_simulacionSNR2doArg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_simulacionSNR2doArg as text
%         str2double(get(hObject,'String')) returns contents of
et_simulacionSNR2doArg as a double

% --- Executes during object creation, after setting all properties.
function et_simulacionSNR2doArg_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_simulacionSNR2doArg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_simulacionSNR3erArg_Callback(hObject, eventdata, handles)
% hObject    handle to et_simulacionSNR3erArg (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_simulacionSNR3erArg as text
%      str2double(get(hObject,'String')) returns contents of
et_simulacionSNR3erArg as a double

% --- Executes during object creation, after setting all properties.
function et_simulacionSNR3erArg_CreateFcn(hObject, eventdata, handles)
% hObject      handle to et_simulacionSNR3erArg (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCSubcargas_Callback(hObject, eventdata, handles)
% hObject      handle to et_UFMCSubcargas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_UFMCSubcargas as
text
%      str2double(get(hObject,'String')) returns contents of
et_UFMCSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCSubcargas_CreateFcn(hObject, eventdata, handles)
% hObject      handle to et_UFMCSubcargas (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCSimTiempo_Callback(hObject, eventdata, handles)
% hObject      handle to et_UFMCSimTiempo (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of et_UFMCSimTiempo as
text
%         str2double(get(hObject,'String')) returns contents of
et_UFMCSimTiempo as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCSimTiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_UFMCSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCEspaciadoSubcargas_Callback(hObject, eventdata,
handles)
% hObject    handle to et_UFMCEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_UFMCEspaciadoSubcargas as text
%         str2double(get(hObject,'String')) returns contents of
et_UFMCEspaciadoSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCEspaciadoSubcargas_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_UFMCEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCPrefCiclico_Callback(hObject, eventdata, handles)
% hObject    handle to et_UFMCPrefCiclico (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of et_UFMCPrefCiclico
as text
%         str2double(get(hObject,'String')) returns contents of
et_UFMCPrefCiclico as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCPrefCiclico_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_UFMCPrefCiclico (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCFiltradoTX_Callback(hObject, eventdata, handles)
% hObject    handle to et_UFMCFiltradoTX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_UFMCFiltradoTX
as text
%         str2double(get(hObject,'String')) returns contents of
et_UFMCFiltradoTX as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCFiltradoTX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_UFMCFiltradoTX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCFiltradoRX_Callback(hObject, eventdata, handles)
% hObject    handle to et_UFMCFiltradoRX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_UFMCFiltradoRX
as text

```

```

%         str2double(get(hObject,'String')) returns contents of
et_UFMCFiltradoRX as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCFiltradoRX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_UFMCFiltradoRX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLASubcargas_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLASubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_WOLASubcargas as
text
%         str2double(get(hObject,'String')) returns contents of
et_WOLASubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_WOLASubcargas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLASubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLASimTiempo_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLASimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_WOLASimTiempo as
text
%         str2double(get(hObject,'String')) returns contents of
et_WOLASimTiempo as a double

```

```

% --- Executes during object creation, after setting all properties.
function et_WOLASimTiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLASimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLAEspaciadoSubcargas_Callback(hObject, eventdata,
handles)
% hObject    handle to et_WOLAEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_WOLAEspaciadoSubcargas as text
%       str2double(get(hObject,'String')) returns contents of
et_WOLAEspaciadoSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_WOLAEspaciadoSubcargas_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_WOLAEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLAPrefCiclico_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLAPrefCiclico (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_WOLAPrefCiclico
as text
%       str2double(get(hObject,'String')) returns contents of
et_WOLAPrefCiclico as a double

```



```

% --- Executes during object creation, after setting all properties.
function et_WOLAPrefCiclico_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLAPrefCiclico (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMFiltradoRX_Callback(hObject, eventdata, handles)
% hObject    handle to et_FOFDMFiltradoRX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FOFDMFiltradoRX
as text
%       str2double(get(hObject,'String')) returns contents of
et_FOFDMFiltradoRX as a double

% --- Executes during object creation, after setting all properties.
function et_FOFDMFiltradoRX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FOFDMFiltradoRX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMSubcargas_Callback(hObject, eventdata, handles)
% hObject    handle to et_FOFDMSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FOFDMSubcargas
as text
%       str2double(get(hObject,'String')) returns contents of
et_FOFDMSubcargas as a double

```

```

% --- Executes during object creation, after setting all properties.
function et_FOFDMSubcargas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FOFDMSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMSimTiempo_Callback(hObject, eventdata, handles)
% hObject    handle to et_FOFDMSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FOFDMSimTiempo
as text
%         str2double(get(hObject,'String')) returns contents of
et_FOFDMSimTiempo as a double

% --- Executes during object creation, after setting all properties.
function et_FOFDMSimTiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FOFDMSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMEspaciadoSubcargas_Callback(hObject, eventdata,
handles)
% hObject    handle to et_FOFDMEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_FOFDMEspaciadoSubcargas as text
%         str2double(get(hObject,'String')) returns contents of
et_FOFDMEspaciadoSubcargas as a double

% --- Executes during object creation, after setting all properties.

```

```

function et_FOFDMEspaciadoSubcargas_CreaterFcn(hObject, eventdata,
handles)
% hObject    handle to et_FOFDMEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMPrefCiclico_Callback(hObject, eventdata, handles)
% hObject    handle to et_FOFDMPrefCiclico (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FOFDMPrefCiclico
as text
%       str2double(get(hObject,'String')) returns contents of
et_FOFDMPrefCiclico as a double

% --- Executes during object creation, after setting all properties.
function et_FOFDMPrefCiclico_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FOFDMPrefCiclico (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMFiltradoTX_Callback(hObject, eventdata, handles)
% hObject    handle to et_FOFDMFiltradoTX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FOFDMFiltradoTX
as text
%       str2double(get(hObject,'String')) returns contents of
et_FOFDMFiltradoTX as a double

% --- Executes during object creation, after setting all properties.
function et_FOFDMFiltradoTX_CreateFcn(hObject, eventdata, handles)

```

```

% hObject    handle to et_FOFDMFiltradoTX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_OFDMSubcargas_Callback(hObject, eventdata, handles)
% hObject    handle to et_OFDMSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_OFDMSubcargas as
text
%         str2double(get(hObject,'String')) returns contents of
et_OFDMSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_OFDMSubcargas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_OFDMSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_OFDMSimTiempo_Callback(hObject, eventdata, handles)
% hObject    handle to et_OFDMSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_OFDMSimTiempo as
text
%         str2double(get(hObject,'String')) returns contents of
et_OFDMSimTiempo as a double

% --- Executes during object creation, after setting all properties.
function et_OFDMSimTiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_OFDMSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_OFDMEspaciadoSubcargas_Callback(hObject, eventdata,
handles)
% hObject    handle to et_OFDMEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_OFDMEspaciadoSubcargas as text
%       str2double(get(hObject,'String')) returns contents of
et_OFDMEspaciadoSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_OFDMEspaciadoSubcargas_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_OFDMEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FBMCSubcargas_Callback(hObject, eventdata, handles)
% hObject    handle to et_FBMCSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FBMCSubcargas as
text
%       str2double(get(hObject,'String')) returns contents of
et_FBMCSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_FBMCSubcargas_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FBMCSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FBMCSimTiempo_Callback(hObject, eventdata, handles)
% hObject    handle to et_FBMCSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FBMCSimTiempo as
text
%          str2double(get(hObject,'String')) returns contents of
et_FBMCSimTiempo as a double

% --- Executes during object creation, after setting all properties.
function et_FBMCSimTiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FBMCSimTiempo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FBMCEspaciadoSubcargas_Callback(hObject, eventdata,
handles)
% hObject    handle to et_FBMCEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_FBMCEspaciadoSubcargas as text
%          str2double(get(hObject,'String')) returns contents of
et_FBMCEspaciadoSubcargas as a double

% --- Executes during object creation, after setting all properties.
function et_FBMCEspaciadoSubcargas_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_FBMCEspaciadoSubcargas (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FBMCProtFiltro_Callback(hObject, eventdata, handles)
% hObject    handle to et_FBMCProtFiltro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_FBMCProtFiltro
as text
%         str2double(get(hObject,'String')) returns contents of
et_FBMCProtFiltro as a double

% --- Executes during object creation, after setting all properties.
function et_FBMCProtFiltro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_FBMCProtFiltro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FBMCFactorSuperposicion_Callback(hObject, eventdata,
handles)
% hObject    handle to et_FBMCFactorSuperposicion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_FBMCFactorSuperposicion as text
%         str2double(get(hObject,'String')) returns contents of
et_FBMCFactorSuperposicion as a double

% --- Executes during object creation, after setting all properties.
function et_FBMCFactorSuperposicion_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_FBMCFactorSuperposicion (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_OFDMPrefCiclicoNum_Callback(hObject, eventdata, handles)
% hObject    handle to et_OFDMPrefCiclicoNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_OFDMPrefCiclicoNum as text
%       str2double(get(hObject,'String')) returns contents of
et_OFDMPrefCiclicoNum as a double

% --- Executes during object creation, after setting all properties.
function et_OFDMPrefCiclicoNum_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_OFDMPrefCiclicoNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_OFDMPrefCiclicoDenom_Callback(hObject, eventdata, handles)
% hObject    handle to et_OFDMPrefCiclicoDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_OFDMPrefCiclicoDenom as text
%       str2double(get(hObject,'String')) returns contents of
et_OFDMPrefCiclicoDenom as a double

% --- Executes during object creation, after setting all properties.
function et_OFDMPrefCiclicoDenom_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_OFDMPrefCiclicoDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```



```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMPrefFiltCiclicoDenom_Callback(hObject, eventdata,
handles)
% hObject    handle to et_FOFDMPrefFiltCiclicoDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_FOFDMPrefFiltCiclicoDenom as text
%       str2double(get(hObject,'String')) returns contents of
et_FOFDMPrefFiltCiclicoDenom as a double

% --- Executes during object creation, after setting all properties.
function et_FOFDMPrefFiltCiclicoDenom_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to et_FOFDMPrefFiltCiclicoDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_FOFDMPrefFiltCiclicoNum_Callback(hObject, eventdata,
handles)
% hObject    handle to et_FOFDMPrefFiltCiclicoNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_FOFDMPrefFiltCiclicoNum as text
%       str2double(get(hObject,'String')) returns contents of
et_FOFDMPrefFiltCiclicoNum as a double

% --- Executes during object creation, after setting all properties.
function et_FOFDMPrefFiltCiclicoNum_CreateFcn(hObject, eventdata,
handles)

```

```

% hObject    handle to et_FOFDMPrefFiltCiclicoNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLAVentanaTXDenom_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaTXDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_WOLAVentanaTXDenom as text
%         str2double(get(hObject,'String')) returns contents of
et_WOLAVentanaTXDenom as a double

% --- Executes during object creation, after setting all properties.
function et_WOLAVentanaTXDenom_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaTXDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLAVentanaTXNum_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaTXNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_WOLAVentanaTXNum
as text
%         str2double(get(hObject,'String')) returns contents of
et_WOLAVentanaTXNum as a double

% --- Executes during object creation, after setting all properties.
function et_WOLAVentanaTXNum_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaTXNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLAVentanaRXDenom_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaRXDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_WOLAVentanaRXDenom as text
%       str2double(get(hObject,'String')) returns contents of
et_WOLAVentanaRXDenom as a double

% --- Executes during object creation, after setting all properties.
function et_WOLAVentanaRXDenom_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaRXDenom (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_WOLAVentanaRXNum_Callback(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaRXNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_WOLAVentanaRXNum
as text
%       str2double(get(hObject,'String')) returns contents of
et_WOLAVentanaRXNum as a double

% --- Executes during object creation, after setting all properties.
function et_WOLAVentanaRXNum_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_WOLAVentanaRXNum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

```

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCPrefFiltCiclicoDenom_Callback(hObject, eventdata,
handles)
% hObject     handle to et_UFMCPrefFiltCiclicoDenom (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_UFMCPrefFiltCiclicoDenom as text
%     str2double(get(hObject,'String')) returns contents of
et_UFMCPrefFiltCiclicoDenom as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCPrefFiltCiclicoDenom_CreateFcn(hObject, eventdata,
handles)
% hObject     handle to et_UFMCPrefFiltCiclicoDenom (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_UFMCPrefFiltCiclicoNum_Callback(hObject, eventdata,
handles)
% hObject     handle to et_UFMCPrefFiltCiclicoNum (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of
et_UFMCPrefFiltCiclicoNum as text
%     str2double(get(hObject,'String')) returns contents of
et_UFMCPrefFiltCiclicoNum as a double

% --- Executes during object creation, after setting all properties.
function et_UFMCPrefFiltCiclicoNum_CreateFcn(hObject, eventdata,
handles)
% hObject     handle to et_UFMCPrefFiltCiclicoNum (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB

```

```

% handles      empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btn_SIMULAR.
function btn_SIMULAR_Callback(hObject, eventdata, handles)
% hObject      handle to btn_SIMULAR (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

%% Obteniendo Valores del GUI

drawnow;
set(handles.txt_estado,'String','ESTADO: Comenzando Simulacion');
set(handles.txt_tiempo,'String','TIEMPO: - min. ');
set(handles.btn_SIMULAR,'Enable','off');
set(handles.btn_GRAFICA,'Enable','off');

%Variables de: Parametros de Simulacion
global Simulation_SNR_OFDM_dB
global Simulation_MonteCarloRepetitions
global Simulation_PlotSignalPowerTheory

SNR_1erArg = str2double(get(handles.et_simulacionSNR1erArg,
'String'));
SNR_2doArg = str2double(get(handles.et_simulacionSNR2doArg,
'String'));
SNR_3erArg = str2double(get(handles.et_simulacionSNR3erArg,
'String'));

Simulation_SNR_OFDM_dB = (SNR_1erArg:SNR_2doArg:SNR_3erArg);
Simulation_MonteCarloRepetitions =
str2double(get(handles.et_simulacionRepMonteCarlo,'String'));
Simulation_PlotSignalPowerTheory =
get(handles.rb_simulacionGrafPoderSi,'Value');

%%if (Simulation_PlotSignalPowerTheory == true)
%%set(handles.txt_estado,'String','SI');
%%else
%%set(handles.txt_estado,'String','NO');
%%end

%Variables de: Parametros de Canal
global Channel_Velocity_kmh
global Channel_PowerDelayProfile
global Channel_DopplerModel
global Channel_CarrierFrequency

```

```

Channel_Velocity_kmh = str2double(get(handles.et_canalVelocidad,
'String'));
Channel_PowerDelayProfile = get(handles.et_canalPerfilRetraso,
'String');
Channel_DopplerModel = get(handles.et_canalModDoppler, 'String');
Channel_CarrierFrequency = str2double(get(handles.et_canalFreqCarga,
'String'));

%Variables de: Parametros de Modulacion QAM
global QAM_ModulationOrder

QAM_ModulationOrder = str2double(get(handles.et_modulacionOrdenQAM,
'String'));

%Variables de: FBMC
global FBMC_NumberOfSubcarriers
global FBMC_NumberOfSymbolsInTime
global FBMC_SubcarrierSpacing
global FBMC_PrototypeFilter
global FBMC_OverlappingFactor

FBMC_NumberOfSubcarriers = str2double(get(handles.et_FBMCSubcargas,
'String'));
FBMC_NumberOfSymbolsInTime = str2double(get(handles.et_FBMCSimTiempo,
'String'));
FBMC_SubcarrierSpacing =
str2double(get(handles.et_FBMCEspaciadoSubcargas, 'String'));
%FBMC_PrototypeFilter = str2double(get(handles.et_FBMCProtFiltro,
'String')); %este
%FBMC_PrototypeFilter = get(handles.pm_FBMCPrototipoFiltro, 'Value');
%este

FBMCPrototipoFiltro = get(handles.pm_FBMCPrototipoFiltro, 'Value');
switch FBMCPrototipoFiltro
    case 1
        FBMC_PrototypeFilter = 'Hermite-OQAM';
    case 2
        FBMC_PrototypeFilter = 'Hermite-QAM';
    case 3
        FBMC_PrototypeFilter = 'Rectangle-QAM';
    case 4
        FBMC_PrototypeFilter = 'RRC-OQAM';
    case 5
        FBMC_PrototypeFilter = 'RRC-QAM';
    case 6
        FBMC_PrototypeFilter = 'PHYDYAS-OQAM';
    case 7
        FBMC_PrototypeFilter = 'PHYDYAS-QAM';
end

FBMC_OverlappingFactor =
str2double(get(handles.et_FBMCFactorSuperposicion, 'String'));

%Variables de: OFDM
global OFDM_NumberOfSubcarriers
global OFDM_NumberOfSymbolsInTime
global OFDM_SubcarrierSpacing

```

```

global OFDM_CyclicPrefixLength

OFDM_NumberOfSubcarriers = str2double(get(handles.et_OFDMSubcargas,
'String'));
OFDM_NumberOfSymbolsInTime = str2double(get(handles.et_OFDMSimTiempo,
'String'));
OFDM_SubcarrierSpacing =
str2double(get(handles.et_OFDMEspaciadoSubcargas, 'String'));

OFDMPrefCiclicoNum = str2double(get(handles.et_OFDMPrefCiclicoNum,
'String'));
OFDMPrefCiclicoDenom = str2double(get(handles.et_OFDMPrefCiclicoDenom,
'String'));

OFDM_CyclicPrefixLength = OFDMPrefCiclicoNum / (OFDMPrefCiclicoDenom
* OFDM_SubcarrierSpacing);

%Variables de: WOLA
global WOLA_NumberOfSubcarriers
global WOLA_NumberOfSymbolsInTime
global WOLA_SubcarrierSpacing
global WOLA_CyclicPrefixLength
global WOLA_WindowLengthTX
global WOLA_WindowLengthRX

WOLA_NumberOfSubcarriers = str2double(get(handles.et_WOLASubcargas,
'String'));
WOLA_NumberOfSymbolsInTime = str2double(get(handles.et_WOLASimTiempo,
'String'));
WOLA_SubcarrierSpacing =
str2double(get(handles.et_WOLAEspaciadoSubcargas, 'String'));
WOLA_CyclicPrefixLength = str2double(get(handles.et_WOLAPrefCiclico,
'String'));

WOLAVentanaTXNum = str2double(get(handles.et_WOLAVentanaTXNum,
'String'));
WOLAVentanaTXDenom = str2double(get(handles.et_WOLAVentanaTXDenom,
'String'));

WOLAVentanaRXNum = str2double(get(handles.et_WOLAVentanaRXNum,
'String'));
WOLAVentanaRXDenom = str2double(get(handles.et_WOLAVentanaRXDenom,
'String'));

WOLA_WindowLengthTX = WOLAVentanaTXNum /
(WOLAVentanaTXDenom*WOLA_SubcarrierSpacing);
WOLA_WindowLengthRX = WOLAVentanaRXNum /
(WOLAVentanaRXDenom*WOLA_SubcarrierSpacing);

%Variables de: FOFDM
global FOFDM_NumberOfSubcarriers
global FOFDM_NumberOfSymbolsInTime
global FOFDM_SubcarrierSpacing
global FOFDM_CyclicPrefixLength
global FOFDM_FilterLengthTX
global FOFDM_FilterLengthRX

```

```

global FOFDM_FilterCyclicPrefixLength

FOFDM_NumberOfSubcarriers = str2double(get(handles.et_FOFDMSubcargas,
'String'));
FOFDM_NumberOfSymbolsInTime =
str2double(get(handles.et_FOFDMSimTiempo, 'String'));
FOFDM_SubcarrierSpacing =
str2double(get(handles.et_FOFDMEspaciadoSubcargas, 'String'));
FOFDM_CyclicPrefixLength = str2double(get(handles.et_FOFDMPrefCiclico,
'String'));

FOFDMFiltradoTX = str2double(get(handles.et_FOFDMFiltradoTX,
'String'));
FOFDMFiltradoRX = str2double(get(handles.et_FOFDMFiltradoRX,
'String'));

FOFDM_FilterLengthTX = (FOFDMFiltradoTX / FOFDM_SubcarrierSpacing);
FOFDM_FilterLengthRX = (FOFDMFiltradoRX / FOFDM_SubcarrierSpacing);

FOFDMPrefFiltCiclicoNum =
str2double(get(handles.et_FOFDMPrefFiltCiclicoNum, 'String'));
FOFDMPrefFiltCiclicoDenom =
str2double(get(handles.et_FOFDMPrefFiltCiclicoDenom, 'String'));

FOFDMDenominador = FOFDMPrefFiltCiclicoDenom *
FOFDM_SubcarrierSpacing;

FOFDM_FilterCyclicPrefixLength = (FOFDMPrefFiltCiclicoNum /
FOFDMDenominador);

%Variables de: UFMC
global UFMC_NumberOfSubcarriers
global UFMC_NumberOfSymbolsInTime
global UFMC_SubcarrierSpacing
global UFMC_CyclicPrefixLength
global UFMC_FilterLengthTX
global UFMC_FilterLengthRX
global UFMC_FilterCyclicPrefixLength
global UFMC_ZeroPaddingInsteadOfCP

UFMC_NumberOfSubcarriers = str2double(get(handles.et_UFMCSubcargas,
'String'));
UFMC_NumberOfSymbolsInTime = str2double(get(handles.et_UFMCsimTiempo,
'String'));
UFMC_SubcarrierSpacing =
str2double(get(handles.et_UFMCEspaciadoSubcargas, 'String'));
UFMC_CyclicPrefixLength = str2double(get(handles.et_UFMCprefCiclico,
'String'));

UFMCFiltradoTX = str2double(get(handles.et_UFMCFiltradoTX, 'String'));
UFMCFiltradoRX = str2double(get(handles.et_UFMCFiltradoRX, 'String'));

UFMC_FilterLengthTX = (UFMCFiltradoTX / FOFDM_SubcarrierSpacing);
UFMC_FilterLengthRX = (UFMCFiltradoRX / FOFDM_SubcarrierSpacing);

```



```

UFMCPrefFiltCiclicoNum =
str2double(get(handles.et_UFMCPrefFiltCiclicoNum, 'String'));
UFMCPrefFiltCiclicoDenom =
str2double(get(handles.et_UFMCPrefFiltCiclicoDenom, 'String'));

UFCMDenominador = UFMCPrefFiltCiclicoDenom * UFMC_SubcarrierSpacing;

UFMC_FilterCyclicPrefixLength = (UFMCPrefFiltCiclicoNum /
UFCMDenominador);

UFMC_ZeroPaddingInsteadOfCP =
get(handles.rb_UFMCRelenoCero_Si, 'Value');

```

```

%Variables para Graficacion INTERNAS

```

```

global SamplingRate
global dt

```

```

% Velocidad de muestreo: debe coincidir aproximadamente con el perfil
de retardo de potencia del
% canal y debe ser mayor que "SubcarrierSpacing *
NumberOfSubcarriers".
SamplingRate          = 15e3*14*14;
dt                    = 1/SamplingRate;

```

```

%% Generate "+Modulation\" Objects
% FBMC Objecto
FBMC = Modulation.FBMC(...
    FBMC_NumberOfSubcarriers,...
% Número de subportadoras
    FBMC_NumberOfSymbolsInTime,...
% Número de símbolos FBMC en el tiempo
    FBMC_SubcarrierSpacing,...
% Espacio entre subportadoras (Hz)
    SamplingRate,...
% Periodo de muestreo(muestras/s)
    0,...
% Frecuencia intermedia de la primer subportadora (Hz). Debe ser un
múltiplo del espaciado de subportadora.
    false,...
% Valor real de la señal Transmita (;se debe cumplir el teorema de
muestreo!)
    FBMC_PrototypeFilter,...
% Filtro prototipo (Hermite, PHYDYAS, RRC) y OQAM o QAM. La tasa de
datos de QAM se reduce en un factor de dos en comparación con O'QAM,
pero aumenta la robustez en canales doblemente selectivos
    FBMC_OverlappingFactor, ...
% factor de Superposición (también determina el exceso de muestreo en
el dominio de la frecuencia)
    0, ...
% Cambio de fase inicial
    true ...
% implementación filtro Polyphase
);

```

```

FBMC_BlockOverlapTime = (FBMC.PrototypeFilter.OverlappingFactor-
1/2)*FBMC.PHY.TimeSpacing;

% OFDM Object
OFDM = Modulation.OFDM(...
    OFDM_NumberOfSubcarriers,...
% Número de subportadoras
    OFDM_NumberOfSymbolsInTime,...
% Número de símbolos OFDM en el tiempo
    OFDM_SubcarrierSpacing,...
% Espacio entre subportadoras (Hz)
    SamplingRate,...
% Periodo de muestreo(muestras/s)
    0,...
% Frecuencia intermedia de la primer subportadora (Hz). Debe ser un
múltiplo del espaciado de subportadora.
    false,...
% Valor real de la señal Transmita (;se debe cumplir el teorema de
muestreo!)
    OFDM_CyclicPrefixLength, ...
% Longitud del prefijo ciclico (s)
    FBMC_BlockOverlapTime ...
% Longitud del tiempo de guarda, es decir, ceros al principio y al
final de la transmisión
    );

% enventanado OFDM (WOLA)
WOLA = Modulation.WOLA(...
    WOLA_NumberOfSubcarriers,...
% Número de subportadoras
    WOLA_NumberOfSymbolsInTime,...
% Número de símbolos WOLA en el tiempo
    WOLA_SubcarrierSpacing,...
% Espacio entre subportadoras (Hz)
    SamplingRate,...
% Periodo de muestreo(muestras/s)
    0,...
% Frecuencia intermedia de la primer subportadora (Hz). Debe ser un
múltiplo del espaciado de subportadora.
    false,...
% Valor real de la señal Transmita (;se debe cumplir el teorema de
muestreo!)
    0, ...
% Longitud del prefijo ciclico (s)
    FBMC_BlockOverlapTime-WOLA_WindowLengthTX/2, ...
% Longitud del tiempo de guarda, es decir, ceros al principio y al
final de la transmisión
    WOLA_WindowLengthTX, ...
% Longitud de la ventana superpuesta (s) en el transmisor
    WOLA_WindowLengthRX ...
% Longitud de la ventana superpuesta (s) en el receptor.
    );

% FOFDM (Filtered OFDM)
FOFDM = Modulation.FOFDM(...
    FOFDM_NumberOfSubcarriers,...
% Número de subportadoras

```

```

    FOFDM_NumberOfSymbolsInTime, ...
% Numero de simbolos FOFDM en el tiempo
    FOFDM_SubcarrierSpacing, ...
% Espacio entre subportadoras (Hz)
    SamplingRate, ...
% Periodo de muestreo (muestras/s)
    0, ...
% Frecuencia intermedia de la primer subportadora (Hz). Debe ser un
múltiplo del espaciado de subportadora.
    false, ...
% Valor real de la señal Transmita (;se debe cumplir el teorema de
muestreo!)
    0, ...
% Longitud del prefijo ciclico (s)
    FBMC_BlockOverlapTime-FOFDM_FilterLengthTX/2, ...
% Longitud del tiempo de guarda, es decir, ceros al principio y al
final de la transmisión
    FOFDM_FilterLengthTX, ...
% Longitud de los filtros de transmisión (s)
    FOFDM_FilterLengthRX, ...
% Longitud de los filtros de recepción(s)
    FOFDM_FilterCyclicPrefixLength ...
% Longitud de los prefijos cíclicos adicionales. Necesario para
combatir ISI e ICI debido al filtrado. Sin embargo, algunos ICI e ISI
pequeños están perfectamente bien.
);

% UFMC (Subband Filtered OFDM)
UFMC = Modulation.UFMC2(...
    UFMC_NumberOfSubcarriers, ...
% Numero de subportadoras
    UFMC_NumberOfSymbolsInTime, ...
% Numero de simbolos UFMC en el tiempo
    UFMC_SubcarrierSpacing, ...
% Espacio entre subportadoras (Hz)
    SamplingRate, ...
% Periodo de muestreo (muestras/s)
    0, ...
% Frecuencia intermedia de la primer subportadora (Hz). Debe ser un
múltiplo del espaciado de subportadora.
    false, ...
% Valor real de la señal Transmita (;se debe cumplir el teorema de
muestreo!)
    0, ...
% Longitud del prefijo ciclico (s). Si se usa relleno cero, esta
longitud representa la longitud de guarda cero en lugar de la longitud
CP
    FBMC_BlockOverlapTime-UFMC_FilterLengthTX/2, ...
% Longitud del tiempo de guarda, es decir, ceros al principio y al
final de la transmisión
    UFMC_FilterLengthTX, ...
% Longitud de los filtros de transmisión (s)
    UFMC_FilterLengthRX, ...
% Longitud de los filtros de recepción(s)
    UFMC_FilterCyclicPrefixLength, ...
% Longitud del prefijo ciclico adicional (o símbolo de protección cero
si se usa ZP) en segundos (s). Necesario para combatir ISI e ICI

```

```

debido al filtrado. Sin embargo, algunos ICI e ISI pequeños están
perfectamente bien.
    UFMC_ZeroPaddingInsteadOfCP ...
% TRUE para Zero Padding (ZP) y FALSE para un prefijo cíclico (CP)
convencional
);

% Number of samples
N_FBMC = FBMC.Nr.SamplesTotal;
N_OFDM = OFDM.Nr.SamplesTotal;
N_WOLA = WOLA.Nr.SamplesTotal;
N_FOFDM = FOFDM.Nr.SamplesTotal;
N_UFMC = UFMC.Nr.SamplesTotal;
N = max([N_FBMC N_OFDM N_WOLA N_FOFDM N_UFMC]);

ChannelModel = Channel.FastFading(...
    SamplingRate,...
% Periodo de muestreo (muestras/s)
    Channel_PowerDelayProfile,...
% Perfil de retardo de alimentación, ya sea cadena o vector: 'Flat',
'AWGN', 'PedestrianA', 'PedestrianB', 'VehicularA', 'VehicularB',
'ExtendedPedestrianA', 'ExtendedPedestrianB', or 'TDL-A_xxns','TDL-
B_xxns','TDL-C_xxns' (with xx the RMS delay spread in ns, e.g. 'TDL-
A_30ns'), or [1 0 0.2] (Perfil de retardo de potencia autodefinido que
depende de la velocidad de muestreo)
    N,...
% Numero total de muestras
    Channel_Velocity_kmh/3.6*Channel_CarrierFrequency/2.998e8,...
% Desplazamiento máximo Doppler: Velocity_kmh / 3.6 * CarrierFrequency
/ 2.998e8
    Channel_DopplerModel,...
% Qué modelo Doppler: 'Jakes', 'Uniforme', 'Discreto-Jakes',
'Discreto-Uniform'. Para "Discrete-", asumimos un espectro Doppler
discreto para mejorar el tiempo de simulación. Esto solo funciona con
precisión si el número de muestras y la velocidad son suficientemente
grandes
    200, ...
% Número de rutas para el proceso WSSUS. Solo relevante para un
espectro Doppler 'Jakes' y 'Uniforme'
    false,...
% correlación del tiempo del canal
    'none',...
% Correlación espacial para canales MIMO
    0,...
% coeficiente de correlación espacial lateral de transmisión
    0,...
% coeficiente de correlación espacial lateral recepción
    1,...
% Número de antenas transmisoras
    1,...
% Número de antenas receptoras
    true,...
% Da una advertencia si las tomas de retardo predefinidas del canal no
se ajustan a la frecuencia de muestreo. Esto generalmente no es un
gran problema si son aproximadamente iguales.
    0,...
% TWDP K parametros

```

```

0 ...
% TWDP delta parametros
);

% PAM y QAM Objeto
PAM =
Modulation.SignalConstellation(sqrt(QAM_ModulationOrder), 'PAM', 0);
QAM = Modulation.SignalConstellation(QAM_ModulationOrder, 'QAM', 0);

% Preasignar potencia
Ps_FBMC = zeros(N_FBMC, 1);
Ps_OFDM = zeros(N_OFDM, 1);
Ps_WOLA = zeros(N_WOLA, 1);
Ps_FOFDM = zeros(N_FOFDM, 1);
Ps_UFMC = zeros(N_UFMC, 1);

% Preasignar densidad espectral de potencia
PSD_FBMC = zeros(N_FBMC, 1);
PSD_OFDM = zeros(N_OFDM, 1);
PSD_WOLA = zeros(N_WOLA, 1);
PSD_FOFDM = zeros(N_FOFDM, 1);
PSD_UFMC = zeros(N_UFMC, 1);

%% SIMULACION
tic

for i_rep = 1:Simulation_MonteCarloRepetitions
% actualizar canal
ChannelModel.NewRealization;

% Datos binarios
BinaryDataStream_FBMC = randi([0
1], FBMC.Nr.Subcarriers*FBMC.Nr.MCSymbols*log2(PAM.ModulationOrder), 1);
BinaryDataStream_OFDM = randi([0
1], OFDM.Nr.Subcarriers*OFDM.Nr.MCSymbols*log2(QAM.ModulationOrder), 1);
BinaryDataStream_WOLA = randi([0
1], WOLA.Nr.Subcarriers*WOLA.Nr.MCSymbols*log2(QAM.ModulationOrder), 1);
BinaryDataStream_FOFDM = randi([0
1], FOFDM.Nr.Subcarriers*FOFDM.Nr.MCSymbols*log2(QAM.ModulationOrder), 1
);
BinaryDataStream_UFMC = randi([0
1], FOFDM.Nr.Subcarriers*FOFDM.Nr.MCSymbols*log2(QAM.ModulationOrder), 1
);

% Símbolos de datos transmitidos (mapear datos binarios a
símbolos)
x_FBMC =
reshape(PAM.Bit2Symbol(BinaryDataStream_FBMC), FBMC.Nr.Subcarriers, FBMC
.Nr.MCSymbols);
x_OFDM =
reshape(QAM.Bit2Symbol(BinaryDataStream_OFDM), OFDM.Nr.Subcarriers, OFDM
.Nr.MCSymbols);
x_WOLA =
reshape(QAM.Bit2Symbol(BinaryDataStream_WOLA), WOLA.Nr.Subcarriers, WOLA
.Nr.MCSymbols);

```

```

x_FOFDM =
reshape(QAM.Bit2Symbol(BinaryDataStream_FOFDM),FOFDM.Nr.Subcarriers,FO
FDM.Nr.MCSymbols);
x_UFMC =
reshape(QAM.Bit2Symbol(BinaryDataStream_UFMC),UFMC.Nr.Subcarriers,UFMC
.Nr.MCSymbols);

% Señal transmitida en el dominio del tiempo.
s_FBMC = FBMC.Modulation(x_FBMC);
s_OFDM = OFDM.Modulation(x_OFDM);
s_WOLA = WOLA.Modulation(x_WOLA);
s_FOFDM = FOFDM.Modulation(x_FOFDM);
s_UFMC = UFMC.Modulation(x_UFMC);

% Convolución de canales
r_FBMC_noNoise = ChannelModel.Convolution(s_FBMC);
r_OFDM_noNoise = ChannelModel.Convolution(s_OFDM);
r_WOLA_noNoise = ChannelModel.Convolution(s_WOLA);
r_FOFDM_noNoise = ChannelModel.Convolution(s_FOFDM);
r_UFMC_noNoise = ChannelModel.Convolution(s_UFMC);

% Canal para el ecualizador de un toque (representa solo una
aproximación porque no se tiene en cuenta la forma del pulso)
h_FBMC =
ChannelModel.GetTransferFunction(FBMC.GetTimeIndexMidPos,FBMC.Implemen
tation.FFTSize,(1:FBMC.Nr.Subcarriers)+FBMC.Implementation.Intermediat
eFrequency);
h_OFDM =
ChannelModel.GetTransferFunction(OFDM.GetTimeIndexMidPos,OFDM.Implemen
tation.FFTSize,(1:OFDM.Nr.Subcarriers)+OFDM.Implementation.Intermediat
eFrequency);
h_WOLA =
ChannelModel.GetTransferFunction(WOLA.GetTimeIndexMidPos,WOLA.Implemen
tation.FFTSize,(1:WOLA.Nr.Subcarriers)+WOLA.Implementation.Intermediat
eFrequency);
h_FOFDM =
ChannelModel.GetTransferFunction(FOFDM.GetTimeIndexMidPos,FOFDM.Implem
entation.FFTSize,(1:FOFDM.Nr.Subcarriers)+FOFDM.Implementation.Interme
diateFrequency);
h_UFMC =
ChannelModel.GetTransferFunction(UFMC.GetTimeIndexMidPos,UFMC.Implemen
tation.FFTSize,(1:UFMC.Nr.Subcarriers)+UFMC.Implementation.Intermediat
eFrequency);

% Calcula la potencia transmitida a lo largo del tiempo.
Ps_FBMC = Ps_FBMC + abs(s_FBMC).^2;
Ps_OFDM = Ps_OFDM + abs(s_OFDM).^2;
Ps_WOLA = Ps_WOLA + abs(s_WOLA).^2;
Ps_FOFDM = Ps_FOFDM + abs(s_FOFDM).^2;
Ps_UFMC = Ps_UFMC + abs(s_UFMC).^2;

% Calcular la densidad espectral de potencia.
PSD_FBMC = PSD_FBMC + abs(fft(s_FBMC)/sqrt(N_FBMC)).^2;
PSD_OFDM = PSD_OFDM + abs(fft(s_OFDM)/sqrt(N_OFDM)).^2;
PSD_WOLA = PSD_WOLA + abs(fft(s_WOLA)/sqrt(N_WOLA)).^2;
PSD_FOFDM = PSD_FOFDM + abs(fft(s_FOFDM)/sqrt(N_FOFDM)).^2;
PSD_UFMC = PSD_UFMC + abs(fft(s_UFMC)/sqrt(N_UFMC)).^2;

```

```

for i_SNR = 1:length(Simulation_SNR_OFDM_dB)
    % Add noise
    SNR_OFDM_dB = Simulation_SNR_OFDM_dB(i_SNR);
    Pn_time     = 1/mean(mean(OFDM.GetSymbolNoisePower(1)))*10^(-
SNR_OFDM_dB/10);
    noise       = sqrt(Pn_time/2)*(randn(N,1)+1j*randn(N,1));

    r_FBMC     = r_FBMC_noNoise + noise(1:N_FBMC);
    r_OFDM     = r_OFDM_noNoise + noise(1:N_OFDM);
    r_WOLA     = r_WOLA_noNoise + noise(1:N_WOLA);
    r_FOFDM    = r_FOFDM_noNoise + noise(1:N_FOFDM);
    r_UFMC     = r_UFMC_noNoise + noise(1:N_UFMC);

    % Demodular señal FBMC
    y_FBMC     = FBMC.Demodulation(r_FBMC);
    y_OFDM     = OFDM.Demodulation(r_OFDM);
    y_WOLA     = WOLA.Demodulation(r_WOLA);
    y_FOFDM    = FOFDM.Demodulation(r_FOFDM);
    y_UFMC     = UFMC.Demodulation(r_UFMC);

    % Ecualizador de un toque
    y_OneTapEqualizer_FBMC = y_FBMC./h_FBMC;
    y_OneTapEqualizer_OFDM = y_OFDM./h_OFDM;
    y_OneTapEqualizer_WOLA = y_WOLA./h_WOLA;
    y_OneTapEqualizer_FOFDM = y_FOFDM./h_FOFDM;
    y_OneTapEqualizer_UFMC = y_UFMC./h_UFMC;

    % Detectar símbolos (cuantificación y demapping a bits).
    DetectedBitStream_OneTapEqualizer_FBMC =
PAM.Symbol2Bit(real(y_OneTapEqualizer_FBMC(:)));
    DetectedBitStream_OneTapEqualizer_OFDM =
QAM.Symbol2Bit(y_OneTapEqualizer_OFDM(:));
    DetectedBitStream_OneTapEqualizer_WOLA =
QAM.Symbol2Bit(y_OneTapEqualizer_WOLA(:));
    DetectedBitStream_OneTapEqualizer_FOFDM =
QAM.Symbol2Bit(y_OneTapEqualizer_FOFDM(:));
    DetectedBitStream_OneTapEqualizer_UFMC =
QAM.Symbol2Bit(y_OneTapEqualizer_UFMC(:));

    % Calcula el BER
    BER_FBMC_OneTapEqualizer(i_SNR,i_rep) =
mean(BinaryDataStream_FBMC~=DetectedBitStream_OneTapEqualizer_FBMC);
    BER_OFDM_OneTapEqualizer(i_SNR,i_rep) =
mean(BinaryDataStream_OFDM~=DetectedBitStream_OneTapEqualizer_OFDM);
    BER_WOLA_OneTapEqualizer(i_SNR,i_rep) =
mean(BinaryDataStream_WOLA~=DetectedBitStream_OneTapEqualizer_WOLA);
    BER_FOFDM_OneTapEqualizer(i_SNR,i_rep) =
mean(BinaryDataStream_FOFDM~=DetectedBitStream_OneTapEqualizer_FOFDM);
    BER_UFMC_OneTapEqualizer(i_SNR,i_rep) =
mean(BinaryDataStream_UFMC~=DetectedBitStream_OneTapEqualizer_UFMC);

end
TimeNeededSoFar = toc;
if mod(i_rep,25)==0

    Inicio1 = 'ESTADO: ';
    Porcentaje = int2str(i_rep/Simulation_MonteCarloRepetitions*100);

```

```

Fin1 = '% Completado';
Mostrar1 = strcat(Inicio1, Porcentaje, Fin1);

Inicio2 = 'TIEMPO: ';
Tiempo =
int2str(TimeNeededSoFar/i_rep*(Simulation_MonteCarloRepetitions-
i_rep)/60);
Fin2 = ' min.';
Mostrar2 = strcat(Inicio2, Tiempo, Fin2);

drawnow;
set(handles.txt_estado,'String',Mostrar1);
set(handles.txt_tiempo, 'String',Mostrar2);

%disp([int2str(i_rep/Simulation_MonteCarloRepetitions*100) '%
Completed! Time Left: approx. '
int2str(TimeNeededSoFar/i_rep*(Simulation_MonteCarloRepetitions-
i_rep)/60) 'min, corresponding to approx. '
int2str(TimeNeededSoFar/i_rep*(Simulation_MonteCarloRepetitions-
i_rep)/3600) 'hour'])
end
end
set(handles.btn_SIMULAR,'Enable','on');
set(handles.btn_GRAFICA, 'Enable','on');

% Take "average"
Ps_FBMC = Ps_FBMC/Simulation_MonteCarloRepetitions;
Ps_OFDM = Ps_OFDM/Simulation_MonteCarloRepetitions;
Ps_WOLA = Ps_WOLA/Simulation_MonteCarloRepetitions;
Ps_FOFDM = Ps_FOFDM/Simulation_MonteCarloRepetitions;
Ps_UFMC = Ps_UFMC/Simulation_MonteCarloRepetitions;
PSD_FBMC = PSD_FBMC/Simulation_MonteCarloRepetitions;
PSD_OFDM = PSD_OFDM/Simulation_MonteCarloRepetitions;
PSD_WOLA = PSD_WOLA/Simulation_MonteCarloRepetitions;
PSD_FOFDM = PSD_FOFDM/Simulation_MonteCarloRepetitions;
PSD_UFMC = PSD_UFMC/Simulation_MonteCarloRepetitions;

% Definir colores para diferentes esquemas de modulación.
ColorFBMC = [0 0 1]*0.5;
ColorOFDM = [1 0 0];
ColorWOLA = [1 0 1];
ColorFOFDM = [1 1 0]*0.7;
ColorUFMC = [0 1 0]*0.5;

% Plot BER
figure(1);
semilogy( Simulation_SNR_OFDM_dB , mean(BER_FBMC_OneTapEqualizer,2)
,'x-','color',ColorFBMC); hold on;
semilogy( Simulation_SNR_OFDM_dB , mean(BER_OFDM_OneTapEqualizer,2)
,'o-','color',ColorOFDM); hold on;
semilogy( Simulation_SNR_OFDM_dB , mean(BER_WOLA_OneTapEqualizer,2)
,'s-','color',ColorWOLA); hold on;

```



```

semilogy( Simulation_SNR_OFDM_dB , mean(BER_FOFDM_OneTapEqualizer,2)
,'d-', 'color',ColorFOFDM); hold on;
semilogy( Simulation_SNR_OFDM_dB , mean(BER_UFMC_OneTapEqualizer,2)
,'*-','color',ColorUFMC); hold on;
ylabel('Relación de error de Bit');
xlabel('Relación de señal a ruido para OFDM [dB]');
% Probabilidad de error de bit teórico
if strcmp(Channel_PowerDelayProfile,'AWGN')
    SNR_morePoints =
    linspace(min(Simulation_SNR_OFDM_dB),max(Simulation_SNR_OFDM_dB),100);
    BitErrorProbability =
    BitErrorProbabilityAWGN(SNR_morePoints,QAM.SymbolMapping,QAM.BitMapping);
    semilogy( SNR_morePoints , BitErrorProbability , 'black'); hold
on;
    ylim([10^-5 1]);
    legend({'FBMC', 'OFDM', 'WOLA', 'f-
OFDM', 'UFMC', 'Theory (OFDM)'}, 'Location', 'SouthWest');
else
    SNR_morePoints =
    linspace(min(Simulation_SNR_OFDM_dB),max(Simulation_SNR_OFDM_dB),100);
    BitErrorProbability =
    BitErrorProbabilityDoublyFlatRayleigh(SNR_morePoints,QAM.SymbolMapping
,QAM.BitMapping);
    semilogy( SNR_morePoints , BitErrorProbability , 'black'); hold
on;
    legend({'FBMC', 'OFDM', 'WOLA', 'f-OFDM', 'UFMC', 'Theory (Doubly-Flat,
OFDM)'}, 'Location', 'SouthWest');
end

% PLOT simulación de poder
figure(2);
time_FBMC = ((0:N_FBMC-1) *dt-FBMC_BlockOverlapTime);
time_OFDM = ((0:N_OFDM-1) *dt-FBMC_BlockOverlapTime);
time_WOLA = ((0:N_WOLA-1) *dt-FBMC_BlockOverlapTime);
time_FOFDM = ((0:N_FOFDM-1)*dt-FBMC_BlockOverlapTime);
time_UFMC = ((0:N_UFMC-1) *dt-FBMC_BlockOverlapTime);
plot( time_FBMC/1e-3 , Ps_FBMC, 'color',ColorFBMC); hold on;
plot( time_OFDM/1e-3 , Ps_OFDM, 'color',ColorOFDM); hold on;
plot( time_WOLA/1e-3 , Ps_WOLA, 'color',ColorWOLA); hold on;
plot( time_FOFDM/1e-3 , Ps_FOFDM, 'color',ColorFOFDM); hold on;
plot( time_UFMC/1e-3 , Ps_UFMC, 'color',ColorUFMC); hold on;
ylabel('Señal Poder');
xlabel('Tiempo [ms]');
legend({'FBMC', 'OFDM', 'WOLA', 'f-OFDM', 'UFMC'});
title('Simulación');
% Teoria

if Simulation_PlotSignalPowerTheory
    figure(5);
    plot( time_FBMC/1e-3 , FBMC.PlotTransmitPower ,
'color',ColorFBMC); hold on;
    plot( time_OFDM/1e-3 , OFDM.PlotTransmitPower ,
'color',ColorOFDM); hold on;
    plot( time_WOLA/1e-3 , WOLA.PlotTransmitPower ,
'color',ColorWOLA); hold on;
    plot( time_FOFDM/1e-3 , FOFDM.PlotTransmitPower,
'color',ColorFOFDM); hold on;

```

```

    plot( time_UFMC/1e-3 , UFMC.PlotTransmitPower ,
'color',ColorUFMC); hold on;
    ylabel('Señal Poder');
    xlabel('Tiempo [ms]');
    legend({'FBMC', 'OFDM', 'WOLA', 'f-OFDM', 'UFMC'});
    title('Teoria');
end

```

```

figure(3);
frequency_FBMC = (0:N_FBMC -1) / (N_FBMC *dt);
frequency_OFDM = (0:N_OFDM -1) / (N_OFDM *dt);
frequency_WOLA = (0:N_WOLA -1) / (N_WOLA *dt);
frequency_FOFDM = (0:N_FOFDM -1) / (N_FOFDM*dt);
frequency_UFMC = (0:N_UFMC -1) / (N_UFMC *dt);
plot( frequency_FBMC /1e6 , 10*log10(PSD_FBMC/max(PSD_FBMC))
, 'color',ColorFBMC); hold on;
plot( frequency_OFDM /1e6 , 10*log10(PSD_OFDM/max(PSD_FBMC))
, 'color',ColorOFDM); hold on;
plot( frequency_WOLA /1e6 , 10*log10(PSD_WOLA/max(PSD_FBMC))
, 'color',ColorWOLA); hold on;
plot( frequency_FOFDM/1e6 ,
10*log10(PSD_FOFDM/max(PSD_FBMC)), 'color',ColorFOFDM); hold on;
plot( frequency_UFMC /1e6 , 10*log10(PSD_UFMC/max(PSD_FBMC))
, 'color',ColorUFMC); hold on;
ylabel('Densidad espectral de potencia [dB]');
xlabel('Frecuencia [MHz]');
ylim([-100 0]);
legend({'FBMC', 'OFDM', 'WOLA', 'f-OFDM', 'UFMC'});
title('Simulación');

```

```

figure(4);
% Plot el perfil de retardo de potencia.
ChannelModel.PlotPowerDelayProfile;

```

```

fprintf('=====\n');
fprintf('====Basic
Settings=====\n');
fprintf('          |(complex)TF-Spacing|  Bandwidth  |      Time
|      BitRate \n');
fprintf('OFDM (with CP)  |%17.2f  |%8.2f MHz  |%8.2f ms  |%8.2f Mbit/s
| \n', OFDM.PHY.TimeSpacing*OFDM.PHY.SubcarrierSpacing
, OFDM.PHY.SubcarrierSpacing*OFDM.Nr.Subcarriers/1e6
, OFDM.PHY.TimeSpacing*OFDM.Nr.MCSymbols/1e-3
, length(DetectedBitStream_OneTapEqualizer_OFDM)/(OFDM.PHY.TimeSpacing*O
FDM.Nr.MCSymbols)/1e6
);
fprintf('FBMC-OQAM      |%17.2f  |%8.2f MHz  |%8.2f ms  |%8.2f Mbit/s
| \n', FBMC.PHY.TimeSpacing*FBMC.PHY.SubcarrierSpacing*2
, FBMC.PHY.SubcarrierSpacing*FBMC.Nr.Subcarriers/1e6
, FBMC.PHY.TimeSpacing*FBMC.Nr.MCSymbols/1e-3
, length(DetectedBitStream_OneTapEqualizer_FBMC)/(FBMC.PHY.TimeSpacing*F
BMC.Nr.MCSymbols)/1e6
);
fprintf('WOLA          |%17.2f  |%8.2f MHz  |%8.2f ms  |%8.2f Mbit/s
| \n', WOLA.PHY.TimeSpacing*WOLA.PHY.SubcarrierSpacing
, WOLA.PHY.SubcarrierSpacing*WOLA.Nr.Subcarriers/1e6
, WOLA.PHY.TimeSpacing*WOLA.Nr.MCSymbols/1e-3
,

```

```

length(DetectedBitStream_OneTapEqualizer_OFDM)/(WOLA.PHY.TimeSpacing*W
OLA.Nr.MCSymbols)/1e6 );
fprintf('FOFDM          |%17.2f |%8.2f MHz |%8.2f ms |%8.2f Mbit/s
| \n', FOFDM.PHY.TimeSpacing*FOFDM.PHY.SubcarrierSpacing ,
FOFDM.PHY.SubcarrierSpacing*FOFDM.Nr.Subcarriers/1e6 ,
FOFDM.PHY.TimeSpacing*FOFDM.Nr.MCSymbols/1e-3 ,
length(DetectedBitStream_OneTapEqualizer_OFDM)/(FOFDM.PHY.TimeSpacing*
FOFDM.Nr.MCSymbols)/1e6 );
fprintf('UFMC          |%17.2f |%8.2f MHz |%8.2f ms |%8.2f Mbit/s
| \n', UFMC.PHY.TimeSpacing*UFMC.PHY.SubcarrierSpacing ,
UFMC.PHY.SubcarrierSpacing*UFMC.Nr.Subcarriers/1e6 ,
UFMC.PHY.TimeSpacing*UFMC.Nr.MCSymbols/1e-3 ,
length(DetectedBitStream_OneTapEqualizer_OFDM)/(UFMC.PHY.TimeSpacing*U
FMC.Nr.MCSymbols)/1e6 );
fprintf('=====
===== \n');
fprintf('=====
===== \n');

fprintf('===== Transmitted Power
===== \n');
fprintf('          | (Sim.) Trans. Energy |      Time      | (Sim.)
Av. Power | SNR rel. to OFDM | \n');
fprintf('OFDM (with CP) | %17.2f mJ |%8.2f ms |%14.2f W |%14.2f dB |
\n', sum(Ps_OFDM)*dt/1e-3 ,
OFDM.PHY.TimeSpacing*OFDM.Nr.MCSymbols/1e-3 ,
sum(Ps_OFDM)*dt/(OFDM.PHY.TimeSpacing*OFDM.Nr.MCSymbols) ,
10*log10( mean(mean(OFDM.GetSymbolNoisePower(1))) /
mean(mean(OFDM.GetSymbolNoisePower(1))) ));
fprintf('FBMC-OQAM      | %17.2f mJ |%8.2f ms |%14.2f W |%14.2f dB |
\n', sum(Ps_FBMC)*dt/1e-3 ,
FBMC.PHY.TimeSpacing*FBMC.Nr.MCSymbols/1e-3 ,
sum(Ps_FBMC)*dt/(FBMC.PHY.TimeSpacing*FBMC.Nr.MCSymbols) ,
10*log10( mean(mean(OFDM.GetSymbolNoisePower(1))) /
mean(mean(FBMC.GetSymbolNoisePower(1))*2 )));
fprintf('WOLA          | %17.2f mJ |%8.2f ms |%14.2f W |%14.2f dB |
\n', sum(Ps_WOLA)*dt/1e-3 ,
WOLA.PHY.TimeSpacing*WOLA.Nr.MCSymbols/1e-3 ,
sum(Ps_WOLA)*dt/(WOLA.PHY.TimeSpacing*WOLA.Nr.MCSymbols) ,
10*log10( mean(mean(OFDM.GetSymbolNoisePower(1))) /
mean(mean(WOLA.GetSymbolNoisePower(1))) ));
fprintf('FOFDM          | %17.2f mJ |%8.2f ms |%14.2f W |%14.2f dB |
\n', sum(Ps_FOFDM)*dt/1e-3 ,
FOFDM.PHY.TimeSpacing*FOFDM.Nr.MCSymbols/1e-3 ,
sum(Ps_FOFDM)*dt/(FOFDM.PHY.TimeSpacing*FOFDM.Nr.MCSymbols) ,
10*log10( mean(mean(OFDM.GetSymbolNoisePower(1))) /
mean(mean(FOFDM.GetSymbolNoisePower(1))) ));
fprintf('UFMC          | %17.2f mJ |%8.2f ms |%14.2f W |%14.2f dB |
\n', sum(Ps_UFMC)*dt/1e-3 ,
UFMC.PHY.TimeSpacing*UFMC.Nr.MCSymbols/1e-3 ,
sum(Ps_UFMC)*dt/(UFMC.PHY.TimeSpacing*UFMC.Nr.MCSymbols) ,
10*log10( mean(mean(OFDM.GetSymbolNoisePower(1))) /
mean(mean(UFMC.GetSymbolNoisePower(1))) ));
fprintf('=====
===== \n');
fprintf('=====
===== \n');

```

```

% --- Executes on selection change in pm_FBMCPrototipoFiltro.
function pm_FBMCPrototipoFiltro_Callback(hObject, eventdata, handles)
% hObject    handle to pm_FBMCPrototipoFiltro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
pm_FBMCPrototipoFiltro contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
pm_FBMCPrototipoFiltro

% --- Executes during object creation, after setting all properties.
function pm_FBMCPrototipoFiltro_CreateFcn(hObject, eventdata, handles)
% hObject    handle to pm_FBMCPrototipoFiltro (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_NOMA_P_Callback(hObject, eventdata, handles)
% hObject    handle to et_NOMA_P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_NOMA_P as text
%       str2double(get(hObject,'String')) returns contents of
et_NOMA_P as a double

% --- Executes during object creation, after setting all properties.
function et_NOMA_P_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_NOMA_P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function et_NOMA_G1_Callback(hObject, eventdata, handles)
% hObject    handle to et_NOMA_G1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_NOMA_G1 as text
%        str2double(get(hObject,'String')) returns contents of
et_NOMA_G1 as a double

% --- Executes during object creation, after setting all properties.
function et_NOMA_G1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_NOMA_G1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_NOMA_G2_Callback(hObject, eventdata, handles)
% hObject    handle to et_NOMA_G2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_NOMA_G2 as text
%        str2double(get(hObject,'String')) returns contents of
et_NOMA_G2 as a double

% --- Executes during object creation, after setting all properties.
function et_NOMA_G2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to et_NOMA_G2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function et_NOMA_Conteo_Callback(hObject, eventdata, handles)
% hObject    handle to et_NOMA_Conteo (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of et_NOMA_Conteo as
text
% str2double(get(hObject,'String')) returns contents of
et_NOMA_Conteo as a double

% --- Executes during object creation, after setting all properties.
function et_NOMA_Conteo_CreateFcn(hObject, eventdata, handles)
% hObject handle to et_NOMA_Conteo (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in btn_GRAFICA.
function btn_GRAFICA_Callback(hObject, eventdata, handles)
% hObject handle to btn_GRAFICA (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

set(handles.btn_SIMULAR,'Enable','off');
set(handles.btn_GRAFICA,'Enable','off');

%%% NOMA parametros

P = str2double(get(handles.et_NOMA_P,'String'));

G1 = str2double(get(handles.et_NOMA_G1,'String'));

G2 = str2double(get(handles.et_NOMA_G2,'String'));

count = str2double(get(handles.et_NOMA_Conteo,'String'));

for alpha = 0:0.01:1 %POTENCIA spl

P1 = P*alpha;
P2 = P - P1;
R1(count) = log2(1 + P1*G1);
R2(count) = log2(1 + P2*G2/(P1*G2 + 1));

count = count + 1;

end

```

```

figure (1)
plot (R1,R2, 'r');
hold on;
grid on;
count = 1;

for alpha = 0:0.01:1 %

P1 = P/2;
P2 = P/2;
R1(count) = alpha*log2(1 + P1*G1/alpha);
R2(count) = (1-alpha)*log2(1 + P2*G2/(1-alpha));
count = count + 1;

end

```

```

figure (1)
plot(R1,R2, 'k');
hold on;
grid on;
box on;
xlabel('Tasa de usuario 1 (bps/Hz)');
ylabel('Tasa de usuario 2 (bps/Hz)');

legend('NOMA', 'OFDMA');

```

```

B = 5*10^6; %ancho de banda Hz
N0 = 10^-21; %-150 dBw/Hz
N = N0*B; % dBW
G1 = 10^-12; %-120 dB
G2 = 10^-14; %-140 dB
Pcircuit = 100; %watt

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%% NOMA

```

```

count = 1;

```

```

for p = 1:1:100 %W

```

```

P1 = p*0.1; %asignar menos potencia a UE1
P2 = p - P1;
R1 = B*log2(1 + P1*G1/N);
R2 = B*log2(1 + P2*G2/(P1*G2 + N));
R = R1 + R2;
SE(count) = R/B; % bit/sec/Hz
EE(count) = (R/(Pcircuit + p)); % bit/watt.sec
count = count + 1;

```

```

end

```

```

figure (2)

```

```

plot(SE,EE,'k');
hold on;
grid on;
xlabel('SE (bit/sec/Hz)');
ylabel('EE (bit/joule)');

% OFDMA

count = 1;
greenpoint = 0;
maxEE = -1000;

for p = 1:1:100 %Watt

P1 = p/2;
P2 = p/2;
R1 = (B/2)*log2(1 + P1*G1/(N0*B/2));
R2 = (B/2)*log2(1 + P2*G2/(N0*B/2));
R = R1 + R2;

SE_line(count) = R/B; % bit/sec/Hz
EE_line(count) = (R/(Pcircuit + p)); % bit/watt.sec = Mbit/joule

count = count + 1;

end

figure (2)
plot(SE_line,EE_line,'g-');
hold on;
grid on;
xlabel('Eficiencia espectral (bit/sec/Hz)');
ylabel('Eficiencia de Energía (bit/joule)');
legend('NOMA','OFDMA');

set(handles.btn_SIMULAR,'Enable','on');
set(handles.btn_GRAFICA,'Enable','on');

```