



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TELECOMUNICACIONES

**“DESARROLLO DE UN SIMULADOR CON INTERFAZ
GRÁFICA DE LAS TÉCNICAS OMA (CDMA, OFDMA) Y
NOMA (SCMA, OPDMA) PARA EL LABORATORIO DE
COMUNICACIONES DELA FIE.”**

Trabajo de Integración Curricular

Tipo: Proyecto de Investigación

Presentado para optar al grado académico de:

INGENIERO EN TELECOMUNICACIONES

AUTOR:

ROGER JOEL ARMAS BASTIDAS

Riobamba – Ecuador

2022



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TELECOMUNICACIONES

**“DESARROLLO DE UN SIMULADOR CON INTERFAZ
GRÁFICA DE LAS TÉCNICAS OMA (CDMA, OFDMA) Y
NOMA (SCMA, OPDMA) PARA EL LABORATORIO DE
COMUNICACIONES DELA FIE.”**

Trabajo de Integración Curricular

Tipo: Proyecto de Investigación

Presentado para optar al grado académico de:

INGENIERO EN TELECOMUNICACIONES

AUTOR: ROGER JOEL ARMAS BASTIDAS

DIRECTOR: ING. JEFFERSON ALEXANDER RIBADENEIRA RAMÍREZ PhD.

Riobamba – Ecuador

2022

© 2022, Roger Joel Armas Bastidas

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Roger Joel Armas Bastidas, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 4 de marzo del 2022



Roger Joel Armas Bastidas

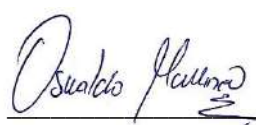
C.I. 0504257130

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA TELECOMUNICACIONES

El Tribunal del Trabajo de Integración Curricular certifica que: El trabajo de Integración Curricular; tipo: Proyecto de Investigación “**DESARROLLO DE UN SIMULADOR CON INTERFAZ GRÁFICA DE LAS TÉCNICAS OMA (CDMA, OFDMA) Y NOMA (SCMA, OPDMA) PARA EL LABORATORIO DE COMUNICACIONES DE LA FIE**”, realizado por el señor: **ROGER JOEL ARMAS BASTIDAS**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de Integración Curricular, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Alberto Leopoldo Arellano Aucancela Mag. PRESIDENTE DEL TRIBUNAL		<u>11/05/2022</u>
Ing. Jefferson Alexander Ribadeneira Ramírez PhD. DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR		<u>11/05/2022</u>
Ing. Oswaldo Geovanny Martínez Guashima Mtr. MIEMBRO DEL TRIBUNAL		<u>11/05/2022</u>

DEDICATORIA

El presente trabajo de titulación se lo dedico principalmente a mis padres, Edison y Verónica, quienes supieron brindarme su cariño y apoyo incondicional mediante su esfuerzo en todos estos años, a mis abuelitos Luis y Rosa por siempre preocuparse por mí a la distancia y que tengo la fortuna de tenerla con vida, a mis hermanas Sheyla y Jelixa quienes fueron una fuente de motivación para seguir cada día. A mis amigos Pedro, Katherine y Alejandro quienes estuvieron siempre en los momentos más felices y difíciles de la carrera. Finalmente dedico esto a Evelyn, una persona muy especial para mí quien me demostró su cariño y enseñó que puedo mejorar cada día.

Roger

AGRADECIMIENTO

Primero agradezco a Dios por darme salud y sabiduría para terminar la carrera de manera satisfactoria. Agradezco a mis padres por darme la oportunidad de ser un profesional y por enseñarme que debo seguir adelante cumpliendo todo lo que me proponga, esto es por ellos y para ellos. De igual manera a mis hermanas quienes son un pilar fundamental para mí y fuente de motivación. Agradezco a mis tíos, Rober, Nanci, Mario y Milton quienes me brindaron su ayuda, a mis compañeros y amigos por los momentos de camaradería entre exámenes y trabajos. A Evelyn por brindarme su compañía y por enseñarme cada día a sonreír un poquito más. Agradezco a cada uno de los docentes que supieron brindarme su conocimiento el cual supe aprovechar para lograr cumplir este objetivo.

Finalmente quiero agradecer al Ing. Jefferson Ribadeneira quien, mediante su conocimiento, experiencia y sus consejos, fue de gran ayuda en la realización de este trabajo.

Roger

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE ECUACIONES	xvi
ÍNDICE DE ABREVIATURAS	xvii
ÍNDICE DE ANEXOS	xix
RESUMEN	xx
SUMMARY	xxi
INTRODUCCIÓN	1

CAPITULO I

1. MARCO TEÓRICO REFERENCIAL	5
1.1. Antecedentes de la investigación	5
1.2. Técnicas de Acceso Múltiple Ortogonales (OMA)	6
<i>1.2.1. Acceso múltiple por división de código (CDMA)</i>	<i>7</i>
<i>1.2.1.1. Principio de funcionamiento</i>	<i>7</i>
<i>1.2.1.2. Generador de códigos ortogonales.....</i>	<i>9</i>
1.2.2. Acceso Múltiple por División de Frecuencias Ortogonales (OFDMA).....	10
<i>1.2.2.1. Principio de funcionamiento</i>	<i>11</i>
<i>1.2.2.2. Mapeo de usuarios</i>	<i>12</i>
<i>1.2.2.3. Inserción de prefijo cíclico CP.....</i>	<i>12</i>
<i>1.2.2.4. Estimación de canal</i>	<i>13</i>
1.3. Técnicas de Acceso Múltiple No Ortogonales (NOMA).....	13
1.3.1. Conceptos básicos de NOMA	14
<i>1.3.1.1. Codificación de superposición (SC).....</i>	<i>14</i>
<i>1.3.1.2. Cancelación sucesiva de interferencias (SIC)</i>	<i>15</i>
1.3.2. Clasificación	16
<i>1.3.2.1. Dominio de Potencia</i>	<i>16</i>
<i>1.3.2.2. Dominio de código</i>	<i>17</i>
1.3.3. Acceso Múltiple por Códigos Dispersos (SCMA).....	18
<i>1.3.3.1. Principio de funcionamiento del codificador SCMA</i>	<i>19</i>
<i>1.3.3.2. Diseño de libro de códigos</i>	<i>20</i>

1.3.3.3.	<i>Algoritmo de Paso de Mensajes (MPA)</i>	22
1.3.4.	Acceso Múltiple por División de Potencia Ortogonal (OPDMA)	23
1.3.4.1.	<i>Funcionamiento de PDM</i>	23
1.3.4.2.	<i>Estructura de OPDMA</i>	25
1.3.4.3.	<i>División de Segmentos de potencia y estrategia de localización</i>	26
1.4.	Canal de comunicación	27
1.4.1.	<i>Canal AWGN</i>	27
1.4.2.	<i>Canal Rice</i>	27
1.4.3.	<i>Canal Rayleigh</i>	28
1.5.	Parámetros de evaluación	28
1.5.1.	<i>Bit Error Rate (BER)</i>	28
1.5.2.	<i>Signal-to-Noise Ratio (SNR)</i>	28
1.5.3.	<i>Energy per-Bit to Noise Power Density Ratio (Eb/No)</i>	28

CAPITULO II

2.	MARCO METODOLÓGICO	29
2.1.	Diseño de la investigación	29
2.1.1.	<i>Tipo de investigación</i>	29
2.1.2.	<i>Métodos de investigación</i>	29
2.1.3.	<i>Técnicas de Investigación</i>	30
2.2.	Determinación de los procesos del simulador	30
2.2.1.	<i>Diagrama de flujo de la interfaz gráfica</i>	31
2.2.2.	<i>Diagrama de flujo de los procesos de las técnicas OMA y NOMA</i>	34
2.2.3.	Simulación de CDMA	36
2.2.3.1.	<i>Transmisor</i>	37
2.2.3.2.	<i>Receptor</i>	39
2.2.4.	Simulación OFDMA	41
2.2.4.1.	<i>Transmisor</i>	41
2.2.4.2.	<i>Receptor</i>	45
2.2.5.	Simulación SCMA	48
2.2.5.1.	<i>Transmisor</i>	49
2.2.5.2.	<i>Receptor</i>	50
2.2.6.	Simulación OPDMA	55
2.2.6.1.	<i>Transmisor</i>	56
2.2.6.2.	<i>Receptor</i>	58
2.2.7.	Canales inalámbricos	60

CAPITULO III

3.	MARCO DE RESULTADOS Y DISCUSIÓN DE LOS RESULTADOS	61
3.1.	Validación de las curvas BER.....	61
3.1.1.	<i>Comparación de resultados CDMA.....</i>	<i>61</i>
3.1.2.	<i>Comparación de resultados OFDMA.....</i>	<i>65</i>
3.1.3.	<i>Comparación de resultados SCMA</i>	<i>69</i>
3.1.4.	<i>Comparación de resultados OPDMA.....</i>	<i>72</i>
3.1.4.1.	<i>Segmentos de potencia con el algoritmo original.....</i>	<i>74</i>
3.1.4.2.	<i>Segmentos de potencia con el algoritmo modificado.....</i>	<i>75</i>
3.1.4.3.	<i>Curvas BER de OPDMA</i>	<i>77</i>
3.2.	Análisis de resultados del simulador	80
3.2.1.	<i>Escenarios de simulación.....</i>	<i>80</i>
3.2.2.	<i>Curvas BER.....</i>	<i>82</i>
3.2.2.1.	<i>CDMA</i>	<i>82</i>
3.2.2.2.	<i>OFDMA.....</i>	<i>85</i>
3.2.2.3.	<i>SCMA.....</i>	<i>89</i>
3.2.2.4.	<i>OPDMA.....</i>	<i>92</i>
3.2.3.	<i>Comparación de las técnicas según el canal.....</i>	<i>95</i>
3.3.	Implementación del simulador	98
	CONCLUSIONES.....	100
	RECOMENDACIONES.....	102
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1:	Algoritmo del proceso de división de PS y asignación al usuario.	26
Tabla 1-2:	Parámetros de entrada para la simulación de CDMA.....	36
Tabla 2-2:	Parámetros de entrada para simulación de OFDM.	41
Tabla 3-2:	Parámetros iniciales del sistema SCMA.	48
Tabla 4-2:	Parámetros de entrada para la función scmadec.	50
Tabla 5-2:	Parámetros de entrada requeridos al inicio de la simulación de OPDMA.	56
Tabla 1-3:	Segmentos de potencia obtenidos con el algoritmo original.	75
Tabla 2-3:	Tasas de transmisión requeridos para los usuarios en base al número de segmentos de potencia.	76
Tabla 3-3:	Segmentos de potencia obtenidos con el algoritmo modificado.....	76
Tabla 4-3:	Tasas de transmisión en función del factor de ensanchamiento para una tasa de chips de 3.84 Mcps.	80
Tabla 5-3:	Tamaño de IFFT y FFT para downlink.	81
Tabla 6-3:	Resumen de SNR requeridos para varias tasas BER de CDMA con canal AWGN.	83
Tabla 7-3:	Resumen de SNR requeridos para varias tasas BER de CDMA con canal Rice.	84
Tabla 8-3:	Resumen de SNR requeridos para varias tasas BER de CDMA con canal Rayleigh.....	85
Tabla 9-3:	Resumen de SNR requeridos para varias tasas BER de OFDMA con canal AWGN.....	86
Tabla 10-3:	Resumen de SNR requeridos para varias tasas BER de OFDMA con canal Rice.	87

ÍNDICE DE FIGURAS

Figura 1-1:	Asignación de recursos en CDMA.	7
Figura 2-1:	Modelo de sistema de comunicaciones de Espectro Ensanchado.	8
Figura 3-1:	Diagrama General de un generador de secuencias m.	9
Figura 4-1:	Generador de secuencias Gold.	10
Figura 5-1:	Asignación de portadoras en OFDMA.	10
Figura 6-1:	Conversión de serie a paralelo en OFDM.	11
Figura 7-1:	Diagrama de bloques de un transmisor y receptor OFDMA.	12
Figura 8-1:	Inserción de prefijo cíclico en OFDMA.	13
Figura 9-1:	Transmisor implementado con SC.	15
Figura 10-1:	Diagrama de bloques de un receptor con SIC.	15
Figura 11-1:	Clasificación de las técnicas NOMA.	16
Figura 12-1:	Transmisiones con NOMA en el dominio de potencia.	17
Figura 13-1:	Estructura general de un transceptor NOMA en el dominio de código.	18
Figura 14-1:	Diagrama de bloques de un sistema SCMA para Downlink.	19
Figura 15-1:	Gráfico de factores para SCMA con $J = 6$ y $K = 4$	20
Figura 16-1:	Multiplexación de las palabras de código de cada usuario.	20
Figura 17-1:	Diagrama de generación del libro de código.	21
Figura 18-1:	Pasos del algoritmo de Mensaje de Paso.	23
Figura 19-1:	TDM, FDM, Y PDM en el dominio de bit.	24
Figura 20-1:	Enlace ascendente con OPDMA.	25
Figura 21-1:	Diagrama de canal de ruido aditivo.	27
Figura 1-2:	Diagrama de flujo del simulador.	32
Figura 2-2:	Ventana de inicio del simulador.	33
Figura 3-2:	Interfaz de simulación de la técnica CDMA.	33
Figura 4-2:	Mensaje de aviso en caso de espera y cuando la simulación está realizada.	34
Figura 5-2:	Mensaje de error cuando no se ingresa un parámetro de simulación.	34
Figura 6-2:	Diagrama de flujo de la simulación de cada técnica.	35
Figura 7-2:	Sistema de transmisión CDMA.	36
Figura 8-2:	Diagrama de bloques para la simulación del transmisor OFDMA.	41
Figura 9-2:	Diagrama de bloques de la simulación SCMA.	48
Figura 10-2:	Traspaso de mensajes del nodo g al nodo v utilizando la gráfica de factores. ...	52
Figura 11-2:	Envío de actualizaciones entre nodos g mediante el traspaso de mensajes con el nodo v.	53

Figura 12-2:	Diagrama de bloques del sistema OPDMA simulado.....	55
Figura 1-3:	Proceso de instalación del simulador en el Laboratorio de Comunicaciones.....	99
Figura 2-3:	Ventana inicial del simulador.....	100

INDICE DE GRÁFICOS

Gráfico 1-3:	BER vs E_b/N_0 para DS-CDMA con modulaciones BPSK y QPSK en un canal AWGN y Rayleigh.....	62
Gráfico 2-3:	BER vs SNR para CDMA con modulación BPSK y canal Rician.	62
Gráfico 3-3:	Comparación del BER en función del SNR de CDMA con modulaciones BPSK y QPSK en canales AWGN y Rayleigh.	63
Gráfico 4-3:	Comparación del BER en función del SNR de CDMA con modulación BPSK sobre canal Rice con factor $K = 1$	64
Gráfico 5-3:	BER vs SNR de OFDMA para BPSK, QPSK, 16-QAM y 64-QAM sobre canal AWGN.....	65
Gráfico 7-3:	BER vs SNR de OFDM para modulación BPSK y QPSK sobre canal Rice. ...	66
Gráfico 8-3:	Comparación del BER en función del SNR de OFDMA para canal AWGN con distintas modulaciones.	67
Gráfico 9-3:	Comparación del BER en función del SNR de OFDMA con modulaciones BPSK y QPSK sobre canal Rice.	68
Gráfico 10-3:	Comparación del BER en función del SNR de OFDMA con modulaciones BPSK y QPSK sobre canal Rayleigh.	69
Gráfico 11-3:	BER vs SNR de SCMA para un canal AWGN.	70
Gráfico 12-3:	BER vs SNR de SCMA para canal Rayleigh.	71
Gráfico 13-3:	Comparación del BER en función del SNR de SCMA para canal AWGN.	72
Gráfico 14-3:	Comparación del BER en función del SNR de SCMA para canal Rayleigh. ...	73
Gráfico 15-3:	BER vs SNR de OPDMA con modulación QPSK en canal AWGN.....	74
Gráfico 16-3:	BER vs SNR de OPDMA con modulación 4-QAM en canal AWGN para 2 usuarios y 2 segmentos de potencia.	74
Gráfico 17-3:	BER vs SNR de OPDMA con modulación 4-QAM en canal AWGN para 5 usuarios y L segmentos de potencia.	75
Gráfico 18-3:	Comparación del BER en función del SNR de OPDMA con modulación QPSK y canal AWGN.	78
Gráfico 19-3:	Comparación del BER en función del SNR de OPDMA con modulación 4QAM y canal AWGN.	79
Gráfico 20-3:	BER en función del SNR obtenido del simulador OPDMA con modulación 4-QAM para un número de 5 usuarios y canal AWGN.....	80
Gráfico 21-3:	Gráfica desplegada en la interfaz gráfica de la simulación CDMA con canal AWGN.....	83

Gráfico 22-3:	Gráfica desplegada en la interfaz gráfica de la simulación CDMA con canal Rice ($K = 6$ dB).	85
Gráfico 23-3:	Gráfica desplegada en la interfaz gráfica de la simulación CDMA con canal Rayleigh.....	86
Gráfico 24-3:	Gráfica desplegada en la interfaz gráfica de la simulación OFDMA con canal AWGN.....	87
Gráfico 25-3:	Gráfica desplegada en la interfaz gráfica de la simulación OFDMA con canal Rice ($K=2$).	88
Gráfico 26-3:	Gráfica desplegada en la interfaz gráfica de la simulación OFDMA con canal Rayleigh.....	89
Gráfico 27-3:	Gráfica desplegada en la interfaz gráfica de la simulación SCMA con canal AWGN.....	90
Gráfico 28-3:	Gráfica desplegada en la interfaz gráfica de la simulación SCMA con canal Rice.....	91
Gráfico 29-3:	Gráfica desplegada en la interfaz gráfica de la simulación SCMA con canal Rayleigh.....	92
Gráfico 30-3:	Gráfica desplegada en la interfaz gráfica de la simulación OPDMA con canal AWGN.....	93
Gráfico 31-3:	Gráfica desplegada en la interfaz gráfica de la simulación OPDMA con canal Rice.....	95
Gráfico 32-3:	Gráfica desplegada en la interfaz gráfica de la simulación OPDMA con canal Rayleigh.....	96
Gráfico 33-3:	Curvas BER en función del SNR de las técnicas OMA y NOMA para un canal AWGN.....	97
Gráfico 34-3:	Curvas BER en función del SNR de las técnicas OMA y NOMA para un canal Rice.....	97
Gráfico 35-3:	Curvas BER en función del SNR de las técnicas OMA y NOMA para un canal Rayleigh.....	98

ÍNDICE DE ECUACIONES

Ecuación 1-1:	Ancho de banda de la señal encancheda.....	8
Ecuación 2-1:	Señal generada por el transmisor SCMA.....	19
Ecuación 3-1:	Matriz de factores F para SCMA con $J = 6$ y $K = 4$	20
Ecuación 4-1:	Operador de rotación de fase	22
Ecuación 5-1:	Relación entre PS y flujo de información en el dominio de bit	24
Ecuación 6-1:	Potencia transmitida en el dominio de bit.....	24
Ecuación 7-1:	Señal para transmitir con OPDMA.....	25
Ecuación 1-2:	Ecuación para el cálculo de LLR.....	54
Ecuación 1-3:	Segmentos de potencia, algoritmo original.	75
Ecuación 2-3:	Segmentos de potencia, algoritmo modificado.....	76

ÍNDICE DE ABREVIATURAS

AWGN	Ruido gaussiano blanco aditivo (<i>Additive White Gaussian Noise</i>)
BFG	Gráfico de factores básico (<i>Basic Factor Graph</i>)
BPSK	Modulación por desplazamiento de fase binaria (<i>Binary-Phase Shift Keying</i>)
CDMA	Acceso múltiple por división de código (<i>Code Division Multiple Access</i>)
EFG	Gráfico de factores extendido (<i>Extended Factor Graph</i>)
LFSR	Registro de desplazamiento con retroalimentación lineal (<i>Linear Feedback Shift Register</i>)
MPA	Algoritmo de paso de mensajes (<i>Message Passing Algorithm</i>)
OFDM	Multiplexación por división de frecuencias ortogonales (<i>Orthogonal frequency-division multiplexing</i>)
OFDMA	Acceso múltiple por división de frecuencias ortogonales (<i>Orthogonal Frequency-Division Multiple Access</i>)
OPDMA	Acceso múltiple por división de potencia ortogonal (<i>Orthogonal Power Division Multiple Access</i>)
PD-NOMA	Acceso múltiple no ortogonal en el dominio de potencia (<i>Power Domain Non-Orthogonal Multiple Access</i>)
PS	Segmentos de potencia (<i>Power Segment</i>)
QAM	Modulación de amplitud en cuadratura (<i>Quadrature Amplitude Modulation</i>)

QPSK	Modulación por desplazamiento de fase cuaternaria (<i>Quaternary-Phase Shift Keying</i>)
RB	Bloque de recursos (<i>Resource Block</i>)
RN	Nodo de recursos (<i>Resource Node</i>)
SC	Codificación por superposición (<i>Superposition Coding</i>)
SCMA	Acceso múltiple por código disperso (<i>Sparse Code Multiple Access</i>)
SIC	Cancelación de interferencia sucesiva (<i>Successive Interference Cancellation</i>)
UE	Equipo de usuario (<i>User Equipment</i>)
VN	Nodo Variable (<i>Variable Node</i>)

ÍNDICE DE ANEXOS

ANEXO A: Código de programación de las simulaciones.

ANEXO B: Libros de código utilizado para la simulación de SCMA.

ANEXO C: Instalación del simulador de interfaz gráfica.

ANEXO D: Manual de simulación para cada técnica.

ANEXO E: Implementación del simulador en el Laboratorio de comunicaciones de la FIE y comprobación del funcionamiento.

RESUMEN

El objetivo de estudio del presente trabajo fue desarrollar un simulador de interfaz gráfica de las técnicas de acceso múltiple ortogonales (CDMA, OFDMA), y no ortogonales (SCMA, OPDMA), mediante el análisis de su funcionamiento con el fin de programar en Matlab las etapas de transmisión y recepción en canales AWGN, Rice y Rayleigh. Debido a la robustez del lenguaje de programación de Matlab, se desarrolló los scripts que contienen las líneas de código necesarias para llevar a cabo la simulación de las técnicas y la interfaz gráfica. En las etapas de transmisión y recepción de las técnicas se realizó el procesamiento de las señales en base a sus principios de funcionamiento. Los canales inalámbricos se implementaron con funciones nativas del software y el cálculo matemático de la ganancia de canal. Con un bucle repetitivo, se realizó el cálculo la tasa de errores de bit (BER) en función de la Relación Señal a Ruido (SNR). Para el análisis y validación de los resultados obtenidos, se realizó simulaciones configurando parámetros de entrada a fin de comparar con resultados presentados de artículos científicos, dando como resultado una similitud significativa con el artículo de referencia donde se obtuvo una diferencia menor 1 dB en el requerimiento de SNR en CDMA, 3 dB en OFDMA, 2,9 dB en SCMA y 3,2 en OPDMA, sin embargo, en los canales de desvanecimiento las curvas difieren en todas las técnicas. Se concluyó que las diferencias presentadas en los resultados obtenidos se deben a la metodología aplicada en el presente trabajo, no obstante, las similitudes en algunas técnicas proporcionan un elevado nivel de confiabilidad para posteriores implementaciones. Se recomienda implementar las etapas de codificación para la corrección de errores en el transmisor y decodificador en el receptor para analizar el impacto en el BER.

Palabras clave: <SIMULADOR DE INTERFAZ GRÁFICA> <COMUNICACIONES INALÁMBRICAS> <ACCESO MÚLTIPLE NO ORTOGONAL (NOMA)> <ACCESO MÚLTIPLE POR DIVISIÓN DE CÓDIGO (CDMA)> <TÉCNICAS DE ACCESO MÚLTIPLE>.

0966-DBRA-UPT-2022



SUMMARY

The objective of this work was to develop a graphical interface simulator for orthogonal (CDMA, OFDMA), and non-orthogonal (SCMA, OPDMA) multiple access techniques, by analyzing its operation in order to program Matlab the stages of transmission and reception on AWGN, Rice and Rayleigh channels. Due to the robustness of the Matlab programming language, the scripts containing the lines of code necessary to carry out the simulation of the techniques and the graphical interface were developed. In the stages of transmission and reception of the techniques, the processing of the signals was carried out based on their operating principles. Wireless channels were increased with native software functions and mathematical calculation of channel gain. With a repetitive loop, the calculation of the bit error rate (BER) was performed as a function of the Signal to Noise Ratio (SNR). For the analysis and validation of the results obtained, simulations were carried out configuring input parameters in order to compare with results presented in scientific articles, resulting in a significant similarity with the reference article where the difference of less than 1 dB in the requirement was obtained of SNR in CDMA, 3 dB in OFDMA, 2.9 in SCMA and 3.2 in OPDMA, however, in the fading channels the curves differ in all techniques. It was concluded that the differences presented in the results obtained are due to the methodology applied in this work, however, the similarities in some techniques provide a high level of reliability for later implementation. It is recommended to implement the coding stages for error correction in the transmitter and decoder in the receiver to analyze the impact on the BER.

Keywords: <GRAPHICAL INTERFACE SIMULATOR> <WIRELESS COMMUNICATIONS> <NON-ORTHOGONAL MULTIPLE ACCESS (NOMA)> <CODE DIVISION MULTIPLE ACCESS (CDMA)> <MULTIPLE ACCESS TECHNIQUES>.



MSc. Wilson G. Rojas

C.I. 0602361842

INTRODUCCIÓN

El aumento de personas que necesitan acceder a una red de información es un hecho palpable en la actualidad debido a los problemas que acarrean situaciones como la pandemia de la COVID-19, la misma que mantuvo a la sociedad en confinamiento realizando sus actividades laborales o académicas con medios telemáticos. Cada persona transmite grandes flujos de bits de información por un canal en base a la tecnología de comunicación inalámbrica o alámbrica utilizada. Dichas tecnologías definen técnicas de acceso múltiple para asegurar la transmisión y recepción correcta de los datos de varios usuarios conectados simultáneamente para lo cual recurren a la ortogonalidad de los recursos compartidos como la frecuencia, el tiempo o códigos, para disminuir la interferencia o la distorsión de las señales.

Las técnicas de acceso múltiple ortogonales (OMA) como CDMA y OFDMA son las más utilizadas en las comunicaciones debido a la compartición de recursos de manera que las señales de los usuarios cumplen con la definición de ortogonalidad. Sin embargo, esta propiedad se ve comprometida por el aumento exponencial del flujo de información en diferentes aplicaciones saturando de cierta forma el espectro radioeléctrico, para lo cual se ha desarrollado varios estudios experimentales y proyectos con el fin de abrir paso a la utilización de frecuencias dentro del rango de las ondas milimétricas (mmWaves). Por otro lado, también se han planteado diferentes técnicas de acceso múltiple no ortogonales (NOMA) para superar los retos que conllevan las aplicaciones del futuro.

La mayoría de las técnicas NOMA se basan en modificaciones de las técnicas OMA como es el caso del Acceso Múltiple de Código Disperso (SCMA) o el Acceso Múltiple por División de Potencia Ortogonal (OPDMA). El estudio estas técnicas y su factibilidad en las comunicaciones inalámbricas se ve desplegado en numerosos documentos en internet, no obstante, al ser un tema novedoso, no suele obtener un gran impacto en la comunidad estudiantil debido a la complejidad de su funcionamiento. Para lo cual, simulaciones de las técnicas NOMA puede ser una herramienta que permita expandir el conocimiento y despertar el interés por estos avances científicos.

El desarrollo del presente trabajo de titulación está destinado a brindar un instrumento de aprendizaje como un simulador de interfaz gráfica sobre las técnicas de acceso múltiple OMA y NOMA para así poder establecer comparaciones de su rendimiento y proponer las nuevas técnicas en futuras implementaciones de redes inalámbricas. El simulador será desarrollado con la utilización de un software matemático a través de los entornos de programación que tenga dicho

software con la finalidad de transmitir datos por canales como AWGN, Rice y Rayleigh, y determinar la calidad de recepción mediante el cálculo de parámetros como el BER.

FORMULACIÓN DEL PROBLEMA

¿Existe una herramienta de simulación con interfaz gráfica de las técnicas de acceso múltiple ortogonales y no ortogonales en el laboratorio de comunicaciones de la FIE?

SISTEMATIZACIÓN DEL PROBLEMA

- ¿Cuál es el principio de funcionamiento de las técnicas OMA y NOMA planteadas?
- ¿Cómo diseñar y programar un simulador de las técnicas de acceso al medio OMA y NOMA?
- ¿Cómo se realizarán las simulaciones de las técnicas en canales AWGN, Rice y Rayleigh?
- ¿Cómo se incorporará el simulador en el laboratorio de comunicaciones y cómo será su utilización?

JUSTIFICACIÓN TEÓRICA

El aumento de la necesidad de conectividad a internet hace necesario altas tasas de transmisión para poder satisfacer la demanda de tráfico producido por los múltiples usuarios. Cerca del 55% de la población mundial es usuario de internet y en Ecuador, el 81% de los habitantes poseen un teléfono inteligente y cerca del 88% tiene acceso a internet en sus hogares lo cual demuestra el constante crecimiento de usuarios que acceden a una red alámbrica o inalámbrica (INEC, 2021, p. 19). Nuevas tecnologías inalámbricas o la innovación de una han sido las respuestas por parte del mundo científico y comercial para cubrir los requerimientos de conectividad como las técnicas a utilizar para la transmisión de información de múltiples usuarios y la forma de acceso al medio.

Las técnicas de acceso no ortogonales (Non-Orthogonal Multiple Access - NOMA) han sido objeto de investigaciones en el ámbito académico con el fin de determinar una tecnología habilitadora a futuras redes inalámbricas mediante el acceso al medio para conectividades masivas. La simulación de las técnicas de acceso múltiple como metodología para comprender su funcionamiento es muy utilizado debido a la facilidad de comparar resultados experimentales obtenidos de pruebas de simulación con diferentes parámetros y canales de comunicación. En el país han realizado simulaciones que se limitan en mostrar el rendimiento de las técnicas en el

dominio de la potencia NOMA (PD-NOMA) solo en canales AWGN, por lo tanto, el análisis de estas técnicas en base a otros modelos de canales inalámbricos resulta interesante para el aprendizaje de las nuevas técnicas en el área académica.

El laboratorio de comunicaciones de la FIE no dispone de un simulador de estas nuevas técnicas de acceso al medio NOMA donde se pueda visualizar su rendimiento en base a las características del sistema de comunicaciones especificadas por el usuario, y además establecer una comparativa con las técnicas de acceso OMA para determinar la técnica más adecuada para futuras implementaciones de redes inalámbricas. Con este simulador se pretende proveer una herramienta de soporte adicional en el aprendizaje de las técnicas de acceso al medio que será utilizado por docentes y estudiantes de las carreras afines.

JUSTIFICACIÓN APLICATIVA

El presente trabajo de titulación propone desarrollar un simulador con interfaz gráfica de las técnicas de acceso al medio ortogonales CDMA/OFDMA y no ortogonales SCMA/OPDMA para la transmisión de datos en canales AWGN, Rice y Rayleigh con la finalidad de proporcionar una herramienta de simulación de las técnicas planteadas donde en base a parámetros de rendimiento como SNR, BER, EB/NO; evaluar las técnicas de acceso múltiple y determinar cuál es la más adecuada para futuras implementaciones.

El simulador será incorporado en el laboratorio de comunicaciones de la Facultad de Informática y Electrónica donde estará disponible para su utilización en el campo académico de las carreras de la facultad a fines.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un simulador con interfaz gráfica de las técnicas OMA (CDMA, OFDMA) y NOMA (SCMA, OPDMA) para el laboratorio de comunicaciones de la FIE.

OBJETIVOS ESPECÍFICOS

- Analizar los principios de funcionamiento de las técnicas de acceso múltiple OMA y NOMA planteadas.
- Diseñar y programar el simulador de las técnicas de acceso múltiple con el uso de software matemático.
- Realizar simulaciones de rendimiento de las técnicas OMA y NOMA planteadas para canales AWGN, Rice y Rayleigh.
- Implementar el simulador en el laboratorio de comunicaciones de la FIE para su utilización en el área académica de las carreras afines.

CAPITULO I

1. MARCO TEÓRICO REFERENCIAL

1.1. Antecedentes de la investigación

El desarrollo de las comunicaciones inalámbricas es una variable que aumenta en base a la tecnología y la saturación de los recursos radioeléctricos, puesto que las futuras implementaciones de redes inalámbricas deben admitir una conectividad masiva de usuarios y/o dispositivos para satisfacer la demanda de dispositivos de bajo costo, baja latencia y diversos tipos de servicios. Con el fin de satisfacer estas necesidades se han propuesto candidatos potenciales para abordar los desafíos planteados en tecnología como 5G, MIMO masivo, comunicaciones de ondas milimétricas, redes ultradensas y el acceso múltiple no ortogonal (NOMA) (Dai, et.al, 2015, p. 74).

En base a los análisis estudiados sobre la estructura de una comunicación inalámbrica y su arquitectura tanto en capa física como en la capa de acceso al medio MAC se puede establecer una perspectiva de cómo han ido cambiando las tecnologías para ofrecer un mejor servicio. La evolución de varios sistemas de comunicaciones inalámbricas abarca el uso de diferentes esquemas de acceso múltiple. Las técnicas de acceso al medio tradicionales como FDMA, TDMA, CDMA, han sido utilizados como esquemas principales de acceso múltiple (MA) en los sistemas de comunicaciones móviles, tecnologías inalámbricas basados en los estándares de la IEEE, los cuales pertenecen a una clase de acceso múltiple ortogonal (OMA) (Anwar, Seet, Hasan, y Li, 2019, p. 2).

Las técnicas de acceso utilizadas en las recientes tecnologías inalámbricas son CDMA y OFDMA debido a su combinación con la tecnología MIMO. Gracias a esto se han realizado estudios sobre el rendimiento de estos esquemas de acceso múltiple para canales con niveles altos de transmisión de datos, donde OFDMA tiene una gran ventaja sobre CDMA en cuanto al rendimiento en situaciones de carga elevada del sistema o cuando existe una conexión de múltiples usuarios. Sin embargo, OFDMA proporciona un alto impacto de interferencia multiusuario cuando las secuencias de propagación de otros usuarios están disponibles en los receptores de los demás usuarios (Phasouliotis y K. C. So, 2009, p. 1).

Las técnicas de acceso múltiple no ortogonales se han popularizado en comunicaciones inalámbricas emergentes como la comunicación por luz visible VLC donde el esquema basado en acceso múltiple de código disperso (SCMA) el cual es parte del dominio de código de NOMA,

es particularmente viable debido a la ganancia de codificación y en su mayoría, una propagación de error mínima o nula en comparación con las técnicas NOMA en el dominio de la potencia (PD-NOMA) (Mitra, Sharma, Kaddoum, y Bhatia, 2020, p. 1).

Por otro lado, la reducción en el consumo de recursos energéticos es uno de los parámetros que también son investigados para el desarrollo de nuevas técnicas de acceso múltiple. En 2016 surge la terminología comunicaciones ecológicas haciendo referencia al ahorro de energía en los dispositivos, para lo cual, una técnica de acceso por división de potencia PDMA cuyo principio es la división en múltiples segmentos de potencia regulares (PS) para transmitir información simultáneamente es estudiada. Esto conlleva a la aparición de un protocolo PDMA ortogonal (OPDMA) para utilizar multiplexación y ganancias de un canal degradado con fines ahorrativos (Han, et.al, 2016, p. 2).

Debido a estos avances de las técnicas de acceso en comunicaciones inalámbricas y como futuros profesionales de la carrera de Ingeniería en Telecomunicaciones, es necesario comprender el funcionamiento y rendimiento de las técnicas que van a la vanguardia con las tecnologías inalámbricas de futuras generaciones para lo cual se destina al laboratorio de comunicaciones una herramienta que permita simular las técnicas NOMA planteadas.

1.2. Técnicas de Acceso Múltiple Ortogonales (OMA)

Las técnicas de acceso múltiple son relevantes para sistemas de comunicaciones multiusuarios, en el que un gran número de usuarios comparten un canal de comunicación común para transmitir información a un receptor (Proakis & Salehi, 2007, p. 1028), para lo cual utilizan técnicas acordes al sistema. Por lo tanto, uno de los métodos más utilizados es la multiplexación para la transmisión de información de diferentes usuarios compartiendo los mismos recursos como el tiempo, frecuencia, o códigos.

Las técnicas de acceso múltiple ortogonales hacen referencia a las técnicas convencionales utilizadas en los sistemas de comunicaciones inalámbricas actuales donde las señales de diferentes usuarios son ortogonales entre sí, es decir, su correlación cruzada es cero (Wang, Xiao, & Ping, 2006, p. 4). Esto permite que los recursos sean repartidos equitativamente entre los usuarios en base a parámetros establecidos para asegurar que no exista interferencia entre canales adyacentes. Generalmente, las técnicas OMA se dividen según la forma en el que dividen o comparten los recursos del canal donde se transmite la información. De acuerdo con Escudero y Céspedes (2021, p. 1), las técnicas de acceso múltiple ortogonal son:

- Acceso múltiple por división de frecuencia (FDMA)
- Acceso múltiple por división de tiempo (TDMA)

- Acceso múltiple por división de código (CDMA)
- Acceso múltiple por división de frecuencia ortogonal (OFDMA)

1.2.1. Acceso múltiple por división de código (CDMA)

Es una técnica de acceso múltiple donde las señales de los usuarios ocupan todo el ancho de banda de un canal como se muestra en la figura 1-1. En CDMA, a cada usuario se le asigna una secuencia de firma distinta (o forma de onda), que el usuario emplea para modular y difundir la señal portadora de información. Las secuencias de distinción también permiten al receptor demodular el mensaje transmitido por múltiples usuarios del canal, que transmiten simultáneamente y, en general, de forma asincrónica (Proakis & Salehi, 2007, p. 1036).

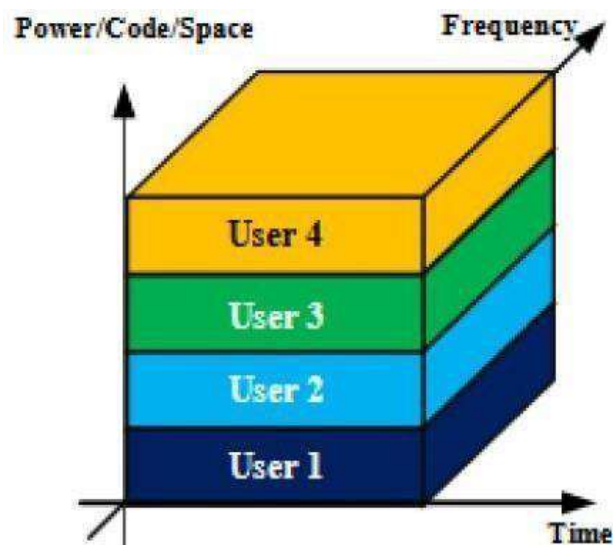


Figura 1-1: Asignación de recursos en CDMA.

Fuente: Chen et. al, 2018, p. 20.

1.2.1.1. Principio de funcionamiento

La técnica CDMA utiliza el principio de espectro ensanchado cuyas señales son utilizadas para la transmisión de información digital y se distinguen por la característica de que su ancho de banda W es mucho mayor que la velocidad de información R en bits/s (Proakis y Salehi, 2007, p. 762). Esta característica proporciona de una gran robustez ante los distintos tipos de interferencias que se encuentran en un sistema de información. El diagrama de un sistema de comunicación basado en

espectro ensanchado se visualiza en la figura 2-1, el cual es similar al diagrama de CDMA, a diferencia de la generación de los códigos pseudo aleatorios PN.

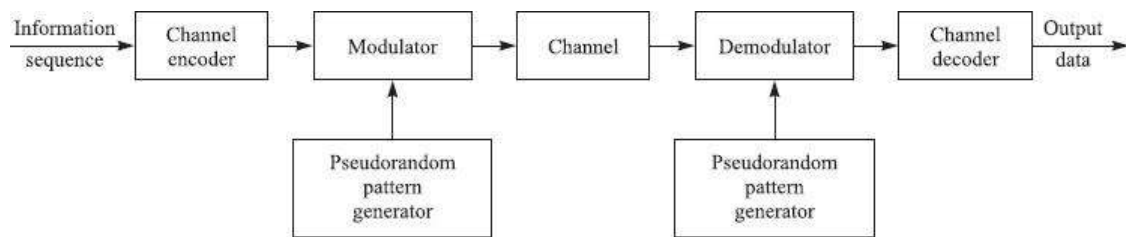


Figura 2-1: Modelo de sistema de comunicaciones de Espectro Ensanchado.

Fuente: Proakis & Salehi, 2007, p. 764.

El tipo de espectro ensanchado se determina mediante el método de generación del patrón pseudoaleatorio. En CDMA, al utilizarse modulaciones digitales PSK, el espectro ensanchado se denomina Secuencia Directa (DS), el cual, mediante un factor de ensanchamiento, los bits generados son ensanchados y sumados con una secuencia pseudoaleatoria. Desde el punto de vista del sistema de comunicación, para utilizar todo el ancho de banda del canal disponible, la fase de la portadora se desplaza pseudoaleatoriamente de acuerdo con el patrón del generador de PN a una velocidad W veces/s. El recíproco de W , denotado por T_c , define la duración de un pulso, que se llama chip y T_c se denomina intervalo de chip. El pulso es el elemento básico en una señal de espectro ensanchado DS (Proakis y Salehi, 2007, p. 765).

El ancho de banda de la señal ensanchada viene dado mediante la siguiente expresión:

$$B_e = \frac{T_b}{T_c}$$

Ecuación 1-1:

Ancho de banda de la señal encanchada.

Donde:

B_e : Ancho de banda

T_b : Tiempo de bit

T_c : Tiempo o Intervalo de pulso (Chip)

1.2.1.2. Generador de códigos ortogonales

Las señales generadas pseudoaleatoriamente se les conoce como pulsos o chips y su velocidad de transmisión se le conoce como tasa de chip, la cual es superior a la tasa de transmisión de la señal en banda base. Según Escudero y Céspedes “a través de esta señal de chip es posible asignar un código ortogonal a cada usuario para evitar la interferencia entre ellos” (2021, p. 5). Las secuencias pseudo aleatorias tienen la propiedad de tener una correlación cercana a cero por lo que se puede obtener la ortogonalidad. Las secuencias más utilizadas son: las secuencias-m y las secuencias Gold.

- Secuencias-m

Una secuencias-m generada pseudoaleatoriamente es una secuencia binaria (no idénticamente cero) que satisface una recurrencia lineal cuyo polinomio característico es primitivo (McEliece, 1987, p. 151). Estas secuencias son generadas con registros de desplazamiento lineales (LFSR), donde la semilla es implantada en base al polinomio primitivo, cuyo resultado tiene un periodo de $2^m - 1$ siendo m el número de registros en el LFSR.

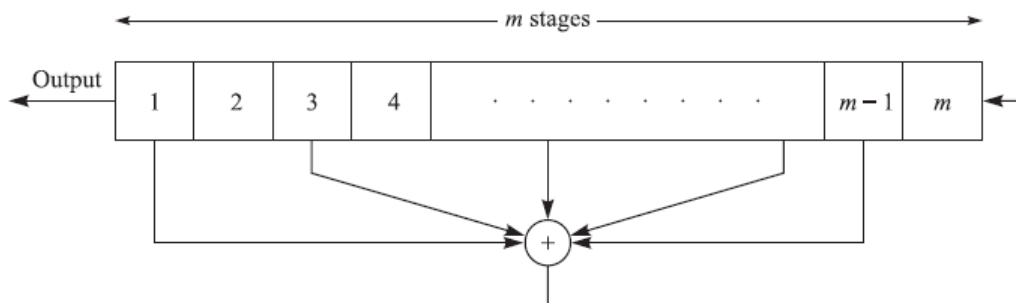


Figura 3-1: Diagrama General de un generador de secuencias m.

Fuente: Proakis y Salehi, 2007, p. 797.

- Secuencias Gold

Las secuencias Gold provienen de secuencias-m, sin embargo, proporcionan mejores propiedades de correlación cruzada periódica. Se obtienen a partir de dos secuencias m de longitud n con una función de correlación cruzada periódica que toma los valores posibles $\{-1, -t(m), t(m) - 2\}$, las cuales se denominan secuencias preferidas (Proakis y Salehi, 2007, p. 799). Cada secuencia-m es generada por un polinomio primitivo, mismo que puede variar según el diseño del sistema o en base a la configuración de una tecnología de comunicación. Uno de los polinomios comunes se pueden visualizar en la figura 4-1.

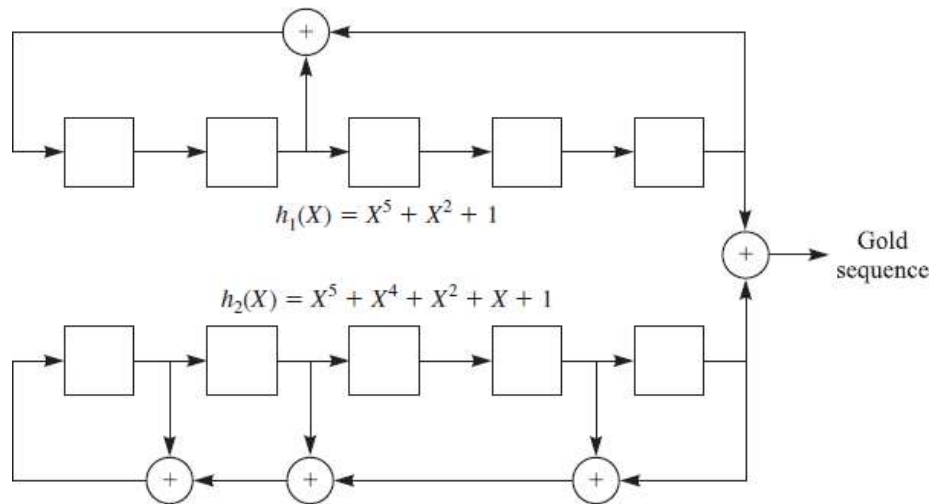


Figura 4-1: Generador de secuencias Gold.

Fuente: Proakis & Salehi, 2007, p. 800.

1.2.2. Acceso Múltiple por División de Frecuencias Ortogonales (OFDMA)

OFDMA es una técnica de acceso múltiple que surge de la variación de la técnica de multiplexación OFDM, donde se hace uso de un gran número de subportadoras ortogonales muy próximas entre sí, que se transmiten en paralelo y se modulan con un esquema de modulación convencional (por ejemplo, QPSK, 16 QAM o 64 QAM) a una cierta velocidad de símbolo baja. La combinación de estas subportadoras permite alcanzar altas velocidades de datos (Huidrobo, 2014, p 267). La asignación de recursos en OFDMA se realiza de manera que a todo usuario se le proporciona un subconjunto de subportadoras, haciendo a esta técnica una versión multiusuario de OFDM.

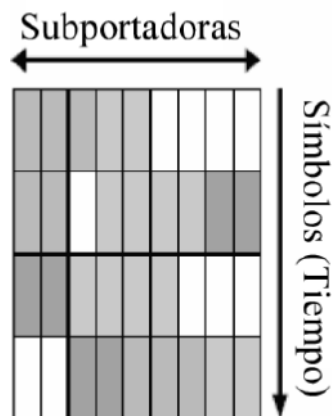


Figura 5-1: Asignación de portadoras en OFDMA.

Fuente: Huidrobo, 2014, p 267.

1.2.2.1. Principio de funcionamiento

OFDMA utiliza el principio de funcionamiento de OFDM donde el flujo de datos de entrada se divide en varios subflujos paralelos con una velocidad de datos reducida (por lo tanto, mayor duración del símbolo) y cada subflujo se transmite en una subportadora ortogonal separada (Yin y Alamouti, 2006, p. 1). Para este proceso es necesaria la conversión de serie a paralelo visualizada en la figura 6-1, donde los símbolos obtenidos de la modulación binaria son separados en las subportadoras con un espacio de 15 KHz entre ellas y una duración de símbolo de 66,7 μ s.

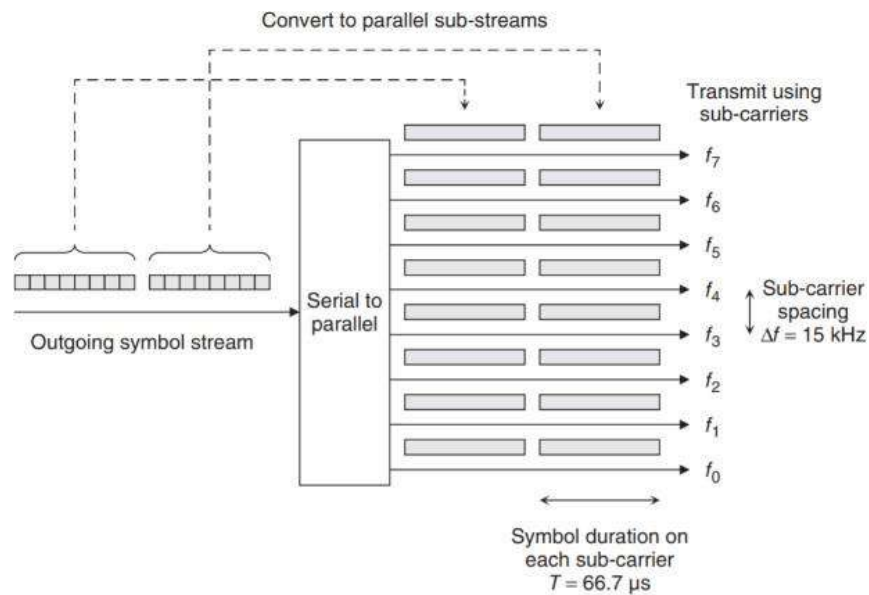


Figura 6-1: Conversión de serie a paralelo en OFDM.

Fuente: Cox, 2014, p. 68.

En la figura 7-1, se puede apreciar el diagrama de bloques de los procesos llevados a cabo en el transmisor y receptor para un enlace descendente donde el transmisor es una estación base y el receptor es un equipo de usuario. Los procesos son similares a OFDM a excepción del mapeo de los elementos de recursos RE, la inserción de prefijo cíclico y la estimación del canal.

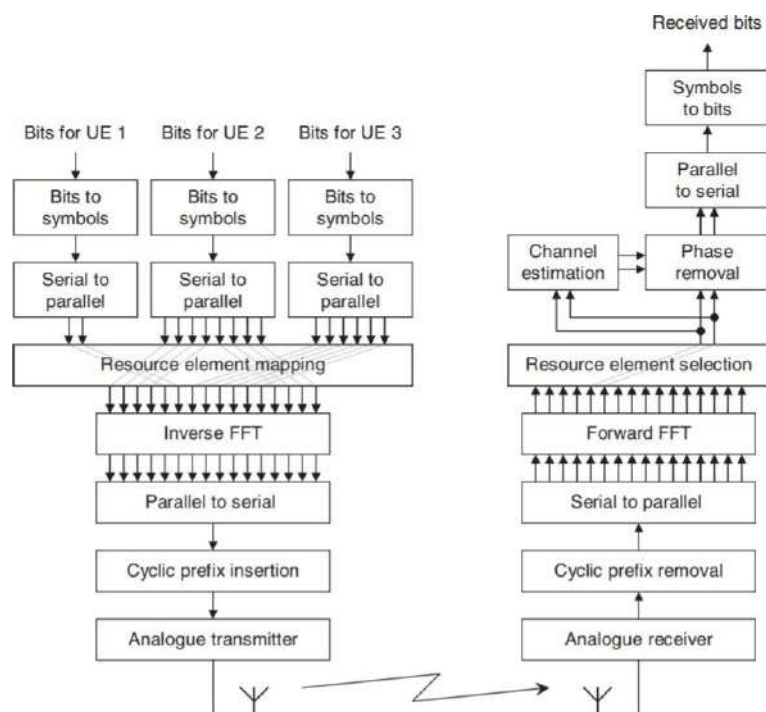


Figura 7-1: Diagrama de bloques de un transmisor y receptor OFDMA.

Fuente: Cox, 2014, p. 73.

1.2.2.2. Mapeo de usuarios

El proceso de mapeo de los símbolos de los usuarios se lleva a cabo en el Resource Element Mapping el cual toma los subflujos individuales y elige las subportadoras para transmitirlos. Las subportadoras destinadas a la transmisión de un usuario pueden estar en un bloque contiguo o pueden estar divididos, es decir, se distribuyen de manera dinámica en base a los requerimientos de la comunicación. La información resultante es la amplitud y la fase de cada subportadora en función de la frecuencia (Cox, 2014, p. 72). Luego de este proceso, los símbolos están listos para ser pasados al dominio del tiempo mediante la transformada inversa rápida de Fourier IFFT.

1.2.2.3. Inserción de prefijo cíclico CP

La inserción de un prefijo cíclico surge de la necesidad de proveer un intervalo de guarda en la transmisión con el objetivo de contrarrestar la interferencia entre símbolos ISI producida por la suma de las contribuciones que pueden llegar al receptor con retardo. Si el período de guarda es más largo que la propagación del retardo, entonces el receptor puede estar seguro de leer la información de un solo símbolo a la vez, sin ninguna superposición con los símbolos que preceden

o siguen (Cox, 2014, p. 72). En la figura 8-1, se puede visualizar la inserción de los prefijos cíclicos como intervalo de guarda.

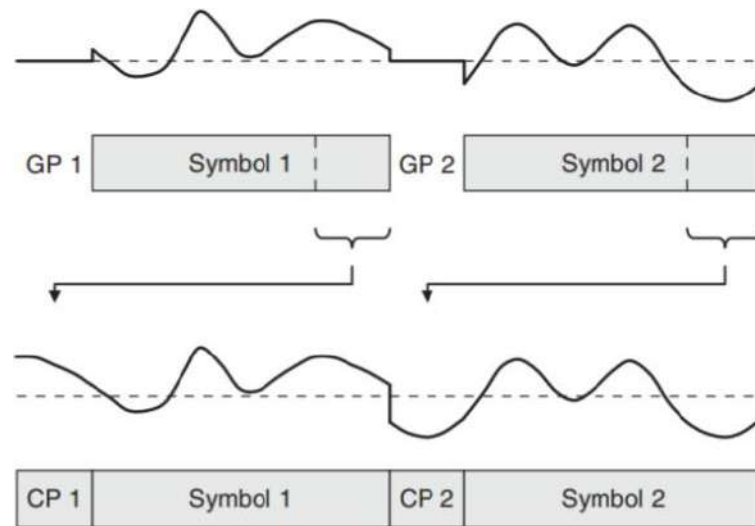


Figura 8-1: Inserción de prefijo cíclico en OFDMA.

Fuente: Cox, 2014, p. 79.

En algunas tecnologías de comunicación inalámbrica que utilizan OFDMA como LTE, el transmisor comienza insertando un período de guarda antes de cada símbolo. Sin embargo, luego copia los datos del final del símbolo siguiente, para completar el período de guarda con una duración que depende de los parámetros del sistema (Cox, 2014, p. 79).

1.2.2.4. Estimación de canal

Debido a las múltiples trayectorias que puede generar un canal de comunicaciones, la interferencia entre símbolos ISI puede resultar un problema significativo en la calidad de la transmisión de datos. En adición del prefijo cíclico, la inserción de símbolos de referencia es una técnica utilizada en el transmisor con el objetivo que el receptor mida dichos símbolos entrantes, los compare con los que definieron en las especificaciones del sistema y utilice el resultado para eliminar el desplazamiento de fase de la señal entrante (Cox, 2014, p. 79).

1.3. Técnicas de Acceso Múltiple No Ortogonales (NOMA)

El término NOMA se refiere a una o varias técnicas de acceso al medio que permiten a múltiples UEs se programen conjuntamente y compartan los mismos recursos de radio en tiempo, frecuencia y/o código (Makki, Chitti, Behravan, y Alouini, 2020, p. 179). De la misma manera, Liu, et al. concuerda en que la idea clave de NOMA es “garantizar que se pueda atender a múltiples usuarios

dentro de un intervalo de recursos determinado, mediante la aplicación de cancelación de interferencias sucesivas (SIC), que es fundamentalmente diferente de las técnicas OMA convencionales” (2016, p. 2).

Dentro de las ideas principales de NOMA, inicialmente solo se considera transmitir información de múltiples usuarios sobre el mismo recurso de frecuencia al mismo tiempo con diferentes niveles de potencia denominado dominio de potencia (Liu, 2016, p. ii), logrando un aprovechamiento óptimo de todo el ancho de banda de un canal.

A diferencia de OMA, NOMA permite relajar la restricción de la asignación de recursos de radio ortogonales, problemas como la programación restringida de un usuario por un tiempo limitado y la asignación de recursos de ancho de banda ya no es una selección binaria, sino es una optimización de la potencia conjunta, la firma de código y el diseño del receptor en un sistema (Chen et. al, 2018, p. 20).

1.3.1. Conceptos básicos de NOMA

1.3.1.1. Codificación de superposición (SC)

La codificación de superposición es una técnica de comunicación simultánea de información a varios receptores mediante una sola fuente, en otras palabras, permite que el transmisor transmita la información de varios usuarios al mismo tiempo (Islam, Avazov, Dobre, y Kwak, 2017, p. 725). Algunos autores han propuesto distintos métodos de diseño de transmisores con este concepto para aplicaciones como transmisiones de televisión. Adicionalmente, la técnica SC puede considerarse como un esquema de modulación multicapa en el que cada capa corresponde a la señal de mensaje modulada y codificada de diferentes usuarios (Anwar, Seet, Hasan, y Li, 2019, p. 5). En la figura 9-1 se puede apreciar el diagrama de bloques de un transmisor con codificación de superposición, el mismo que superpone las señales de mensaje de los usuarios independientes añadiéndolas para constituir una señal compuesta antes de transmitirla al canal (Li, et.al , 2007, p. 330).

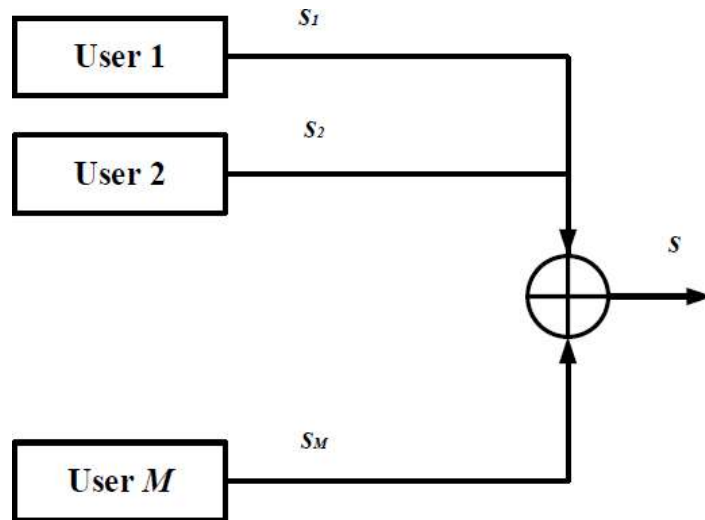


Figura 9-1: Transmisor implementado con SC.

Fuente: Anwar, Seet, Hasan, & Li, 2019, p. 6.

1.3.1.2. Cancelación sucesiva de interferencias (SIC)

Para decodificar la señal superpuesta por el transmisor en la etapa de recepción, la técnica SIC se utiliza, cuya función es decodificar sucesivamente las señales de los usuarios. Después de decodificar la señal de un usuario, se resta de la señal combinada antes de decodificar la señal del siguiente usuario, las cuales son tratadas como una interferencia, pero luego se decodifica esta última con el beneficio de que la señal de la primera ya ha sido eliminada (Islam, Avazov, Dobre, y Kwak, 2017, p. 727). El orden de la decodificación depende del nivel de intensidad de la señal con la que llega al receptor y con esto se decodifica las señales más fuertes y aislar las señales débiles que pueden ser consideradas como interferencias. En la figura 10-1 se puede observar un esquema de un receptor con esta técnica aplicada.

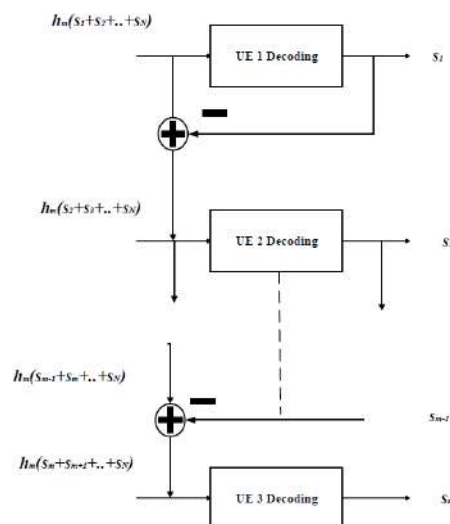


Figura 10-1: Diagrama de bloques de un receptor con SIC.

Fuente: Anwar, Seet, Hasan, & Li, 2019, p. 7.

1.3.2. Clasificación

A raíz de múltiples estudios y experimentaciones, se genera la diversificación de los tipos de técnicas en base al dominio en el que sus recursos son compartidos entre los usuarios. Según Wang et al., (2018, p. 11) establecen que las técnicas de acceso múltiple no ortogonales se pueden clasificar en dos dominios como lo muestra la figura 11-1.

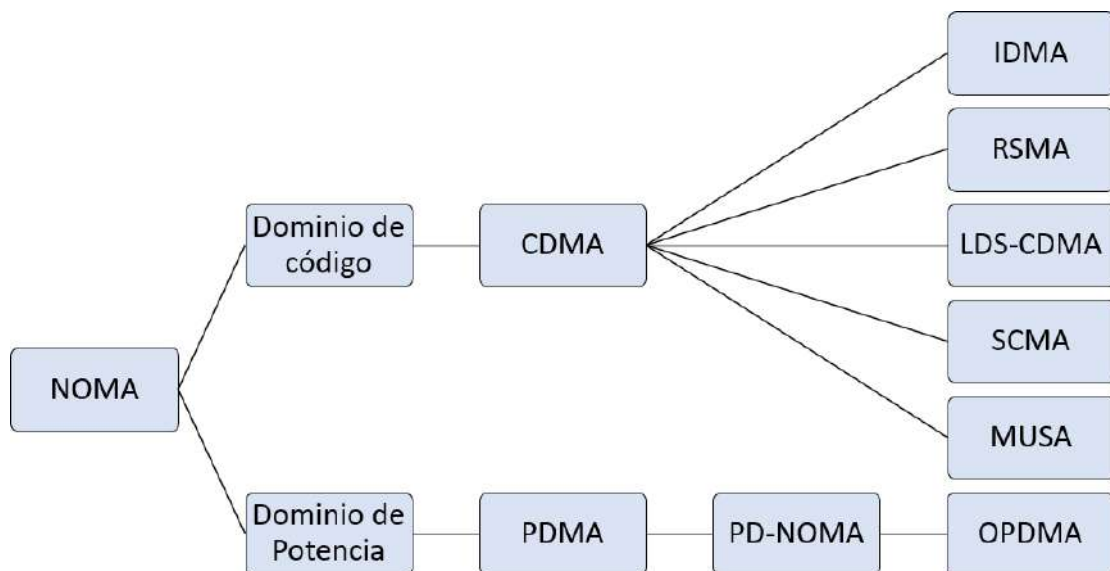


Figura 11-1: Clasificación de las técnicas NOMA.

Fuente: Wang, Zhang, Yang, & Hanzo, 2018, p. 11.

1.3.2.1. Dominio de Potencia

En el dominio de la potencia los usuarios se superponen en un mismo recurso (tiempo/frecuencia/código) mediante la asignación de diferentes niveles de potencia (Power Allocation-PA) como se observa en la figura 12-1. Por tanto, varios usuarios acceden al canal de forma no ortogonal aplicando el principio de codificación de superposición (SC) (Anwar, et al., 2019, p. 4), mejorando significativamente la capacidad de transmisión reduciendo interferencias y, además, el ahorro de recursos energéticos de los transmisores y receptores.

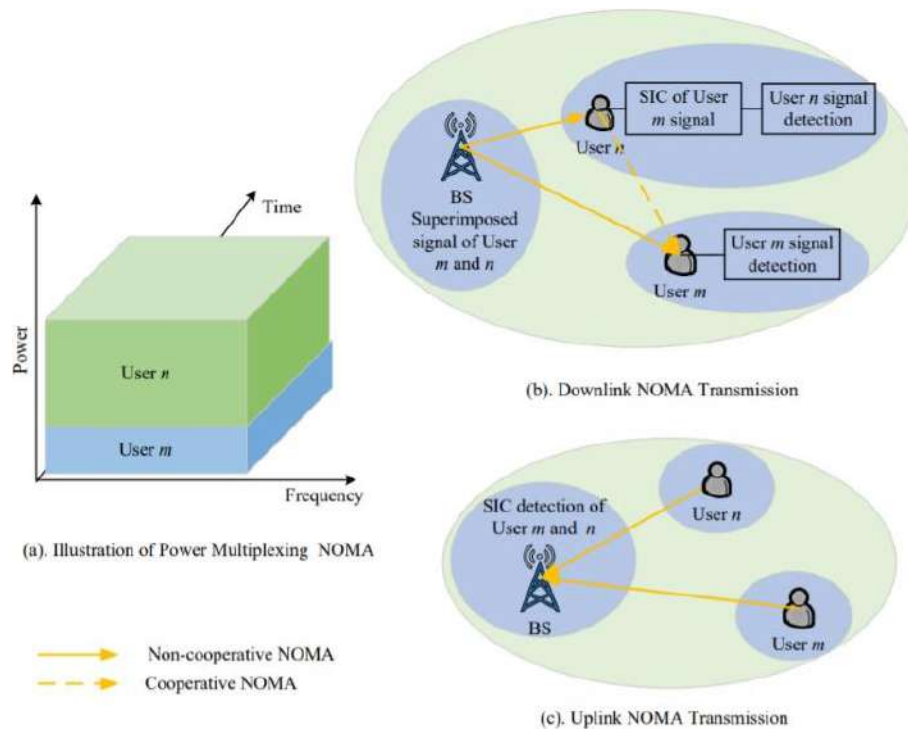


Figura 12-1: Transmisiones con NOMA en el dominio de potencia.

Fuente: Liu, 2016, p. 22.

1.3.2.2. Dominio de código

NOMA en el dominio de código se basa en el principio de funcionamiento de la técnica CDMA clásico, es decir, se distingue a varios usuarios con la ayuda de secuencias de difusión únicas y específicas del usuario logrando una compartición de los recursos para la transmisión de información a través de la ortogonalización de las señales, sin embargo, esto no se logra en la práctica, lo que conlleva en la utilización de secuencias de ensanchamiento pseudoaleatorias no ortogonales, por lo que CDMA constituye una técnica NOMA clásica, que conduce a las diversas soluciones NOMA de dominio de código (Wang, et al., 2018, p. 12).

En la figura 13-1 se puede apreciar el diagrama de bloques general de un transceptor NOMA en el dominio de código.

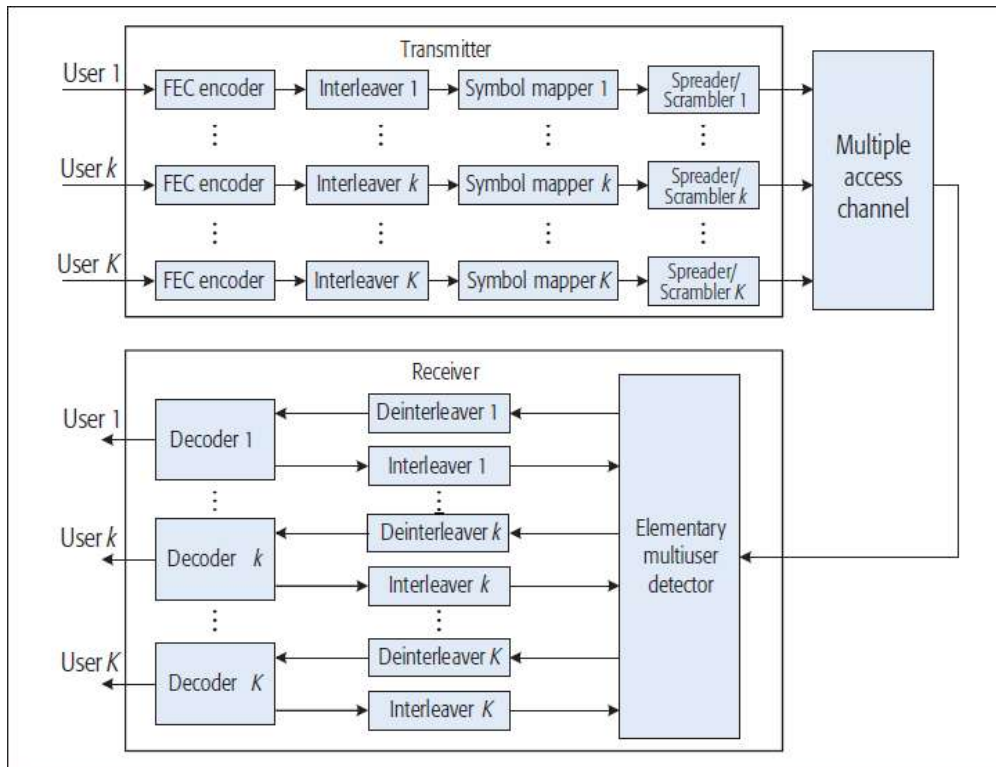


Figura 13-1: Estructura general de un transceptor NOMA en el dominio de código.

Fuente: Wang, et al., 2018, p. 13.

1.3.3. Acceso Múltiple por Códigos Dispersos (SCMA)

El Acceso múltiple de código disperso SCMA es una técnica que se basa en LDS-CDMA (CDMA asistido por esparcimiento de baja densidad), donde las secuencias LDS se utilizan para la separación de usuarios. Sin embargo, a diferencia de LDS-CDMA, SCMA mapea directamente los bits transmitidos a palabras de código multidimensionales de valor complejo para cada usuario (Wang, et al., 2018, p. 12).

En la figura 14-1 se puede apreciar el diagrama de un sistema de comunicación con SCMA para un enlace descendente. Donde b_j representa el flujo de bits a la entrada de los codificadores SCMA y x_j hace referencia a los símbolos obtenidos a través del libro de códigos, los cuales son sumados o mapeados para la transmisión por el canal de comunicación. En el receptor destaca el detector de usuarios a través del algoritmo de paso de mensajes MPA.

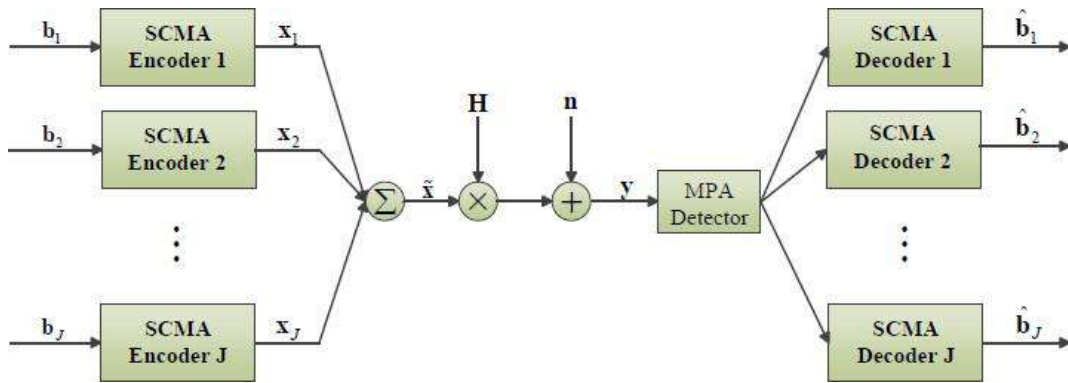


Figura 14-1: Diagrama de bloques de un sistema SCMA para Downlink.

Fuente: Yu et al., 2018, p. 3

Matemáticamente la señal generada a la salida del transmisor se puede expresar como:

$$y = \sum_{j=1}^J x_j \mathbf{H} + n$$

Ecuación 2 - 1:

Señal generada por el transmisor SCMA.

donde x_j son las palabras de código multiplexados de cada usuario, \mathbf{H} es la ganancia de canal, y n es el ruido blanco gaussiano producido por el sistema.

1.3.3.1. Principio de funcionamiento del codificador SCMA

La codificación en SCMA se basa en un mapeo que transforma $\log_2 M$ bits a valores complejos obtenidos de un libro de códigos de K dimensiones y de tamaño M , en otras palabras, $\log_2 M$ bits se codifican en una palabra de código de dimensión K del libro de códigos de usuario predefinido con tamaño M (Lou, et al., 2018, p. 2). El libro de códigos este compuesto por palabras de código las cuales están representados como vectores esparcidos con entradas distintas a cero $N < K$. Todos los libros de códigos poseen la misma cantidad de ceros en $K-N$ dimensiones. Una matriz de mapeo \mathbf{V} mapea las N dimensiones distintas de cero al dominio complejo de K dimensiones (Taherzadeh, et al., 2014, p. 1).

Consideramos un sistema SCMA con J usuarios y K recursos físicos ortogonales, donde el factor de sobrecarga se define como $\lambda = J/K$, este factor de sobre carga en SCMA es mayor a 1. El funcionamiento de la codificación de SCMA se puede representar como un gráfico de factores, como se muestra en la Figura 15-1, donde los R_N y V_N representan a Nodos de Recursos y Nodos Variables, respectivamente (Lou, et al., 2018, p. 2).

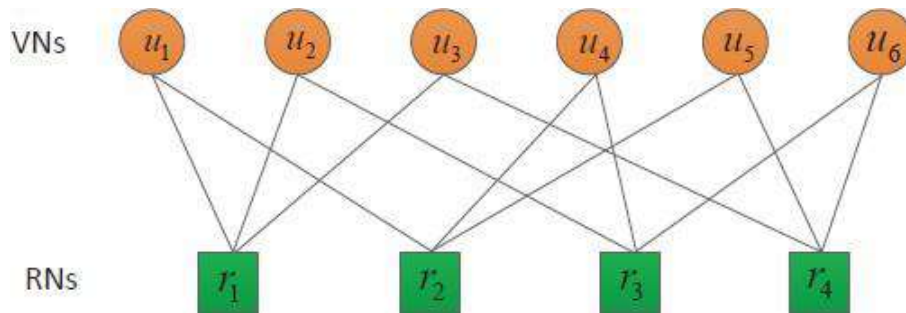


Figura 15-1: Gráfico de factores para SCMA con J = 6 y K = 4.

Fuente: Lou, et al., 2018, p. 2.

El gráfico de factores se puede expresar como una matriz F compuesta de j usuarios y k recursos ortogonales como columnas y filas respectivamente.

$$F = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Ecuación 3-1:

Matriz de factores F para SCMA con J = 6 y K = 4

Cada columna \$F_j\$ de esta matriz representa el mapeo a cada usuario expresado como la matriz diagonal de la columna, es decir \$V_j = \text{diag}(F_j)\$. Posterior al proceso de mapeo, las palabras de código SCMA de J usuarios se pueden multiplexar de manera síncrona, definida como multiplexación SCMA (Lou, et al., 2018, p. 2). En la figura 16-1 se muestra el proceso de multiplexación de los libros de códigos.

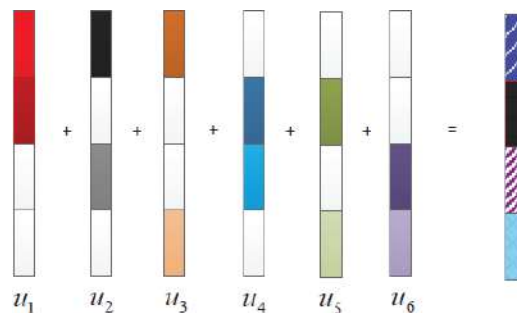


Figura 16-1: Multiplexación de las palabras de código de cada usuario.

Fuente: Lou, et al., 2018, p. 2.

1.3.3.2. Diseño de libro de códigos

Los libros de códigos contienen valores complejos obtenidos mediante métodos que permiten detectar a los usuarios en el receptor. El diseño del libro de códigos SCMA es complicado ya que se multiplexan varias capas con diferentes libros de códigos (Tahezadeh, et al., 2014, p. 1), por lo

tanto, se plantean técnicas para su diseño las cuales involucran operaciones especiales como conjugación compleja, rotación de fase y permutación vectorial. El diseño puede ser una combinación de estas operaciones dependiendo del rendimiento de SCMA en la mejora de BER (Aamer, et.al, 2021, p. 322).

El diseño de los libros de códigos varía acorde al sistema de comunicaciones general y el rendimiento que se desea de los mismos. Sin embargo, los métodos más utilizados se pueden clasificar en tres categorías: optimización basada en celosía, rotación de constelaciones con permutación y optimización de gráficos de factores.

- Optimización basada en celosía

En este método propuesto por Zhang, et al. (2019, pp. 169-173), se optimiza las distancias euclidianas entre los puntos de una constelación a través de una ganancia máxima de codificación la cual se obtiene de la estructura matemática de celosía expresada como una matriz generadora G , donde su determinante es necesaria para el cálculo de esta ganancia. Los elementos de esta matriz se utilizan para calcular la parte real de los puntos de constelación que formaran parte de una constelación madre. Finalmente, para obtener el libro de código se multiplica la constelación madre con los valores de la matriz de factor de sobrecarga F_j como se puede observar en la figura 17-1.

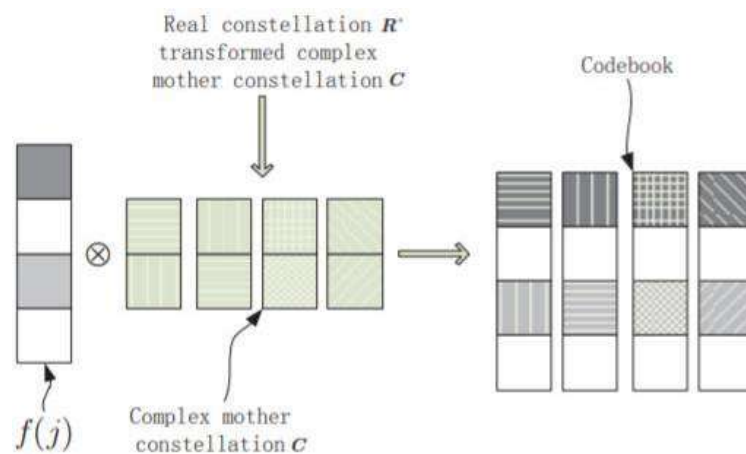


Figura 17-1: Diagrama de generación del libro de código.

Fuente: Zhang, et al., 2019, p. 172.

- Rotación de constelaciones

La rotación de constelaciones es un método para aumentar la distancia euclidiana mínima de las palabras de código de las constelaciones producidas con un bloque de modulación m-QAM. El diseño del libro de códigos planteado por Cai, et al. (2016, p. 2) involucra el seguimiento de 4 pasos:

- 1) Se define un subconjunto S_1 de la red bidimensional Z^2 para la primera dimensión de la constelación madre.
- 2) Cada punto del subconjunto S_1 puede ser etiquetado mediante codificación gray.
- 3) Se multiplica a cada elemento del subconjunto S_1 por un elemento U_N el cual contiene la matriz de rotaciones en fase para obtener la matriz de constelaciones madre.
- 4) Los elementos de dimensiones pares (filas) de la constelación base son reordenados (Interleaving).

Una vez construido la constelación madre, los libros de códigos para cada usuario, se multiplica la matriz que representa la gráfica de factores del codificador SCMA con el operador de rotación del ángulo de fase φ_u que viene dado por:

$$\varphi_u = (u - 1) \frac{2\pi}{Md_f} + e_u \frac{2\pi}{M} \quad \forall u = 1, \dots, df$$

Ecuación 4-1:
Operador de rotación de fase

Donde

M : Número de puntos en la constelación,

e_u : Miembro arbitrario de Z ,

d_f : Número de usuarios conectados a un mismo recurso.

- Optimización del gráfico de factores

La optimización del gráfico de factores se realiza mediante una extensión de este denominado gráfico de factor extendido (EFG). Para su construcción se realizan las operaciones de copiar y permutar en el gráfico de factor básico (BFG), donde BFG denota el gráfico de factor SCMA normal. Cada VN en EFG está conectado a RN que pertenecen a diferentes RB y el mensaje correspondiente se transmite a través de recursos físicos ortogonales en múltiples bloques de recursos (Zhong, et al., 2018, p. 1).

1.3.3.3. Algoritmo de Paso de Mensajes (MPA)

El algoritmo de paso de mensajes MPA es un método de decodificación SCMA para detectar los datos de los usuarios y reducir la interferencia entre las capas de la técnica de acceso al medio. En términos generales, el algoritmo descompone el procesamiento de señales complejas en pasos iterativos los cuales contienen subprocesos relativamente simples (Min y Sun, 2019, p. 325).

El algoritmo obtiene la palabra de código perteneciente a un flujo de información mediante un mensaje de paso iterativo el cual proporciona valores de confiabilidad de la palabra de código SCMA transmitida por cada usuario a lo largo de los bordes del gráfico de factores. En cuanto a los bordes, estos vinculan el nodo de recursos (RN) y el nodo variable (VN), que representan a los recursos físicos ortogonales y a los usuarios respectivamente (Min & Sun, 2019, p. 325). Sea \mathcal{U} el conjunto de símbolos potenciales a transmitir por cada usuario con $|\mathcal{U}| = M$, $V(k)$ es el conjunto de todos los nodos variables a los recursos físicos (Bayesteh, Nikopour, Taherzadeh, Baligh, & Ma, 2015, p. 2). Los pasos iterativos se componen de la siguiente manera:

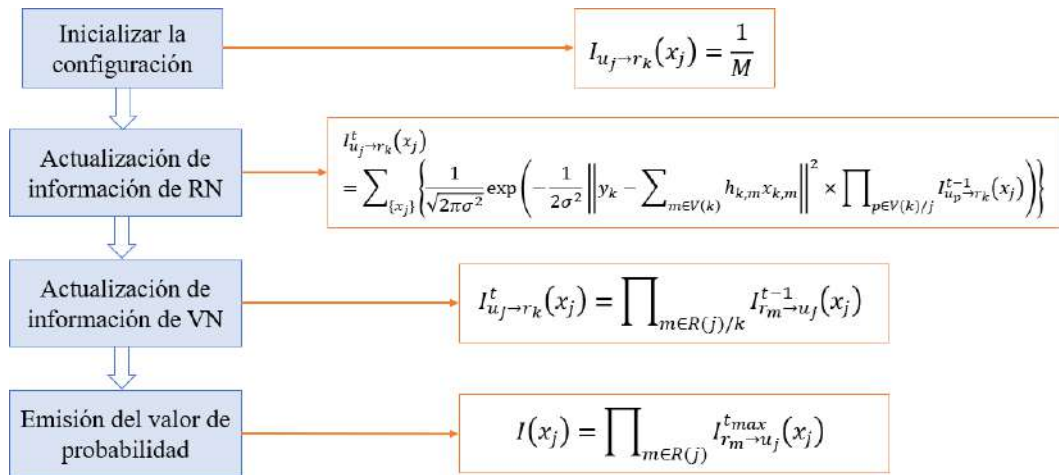


Figura 18-1: Pasos del algoritmo de Mensaje de Paso.

Fuente: Min & Sun, 2019, p. 325.

1.3.4. Acceso Múltiple por División de Potencia Ortogonal (OPDMA)

OPDMA se plantea como un protocolo destinado a proporcionar ortogonalidad y equidad de bits para usuarios de acceso múltiple mediante una extensión de la Multiplexación por División de Potencia (PDM) en el dominio de bit. Este protocolo realiza la asignación de sus recursos por medio de segmentos de potencia (PS) ortogonales. El término “ortogonal” en esta técnica NOMA hace referencia a que los PS aseguran que los flujos de información se puedan entregar de forma independiente con diferentes ganancias de canal (Han, et.al, 2016, p. 5). Para comprender el protocolo planteado por Han, es necesario conceptualizar el funcionamiento de la extensión de PDM propuesto por el mismo.

1.3.4.1. Funcionamiento de PDM

La técnica de multiplexación por división de potencia o PDM, a diferencia de las técnicas OMA convencionales como TDM y FDM, asigna segmentos de potencia PS que se utilizan

independientemente para flujos de información paralelos completamente diferentes sin las divisiones de tiempo y frecuencia (Han y Ma, 2016, p. 3). Este método de PDM propuesto se somete a una transformación lineal del dominio de la potencia a dominio de bit para ser comparado con las técnicas de multiplexación ortogonales. En la figura 19-1 se visualiza la multiplexación de TDM, FDM y PDM en el dominio de bit.

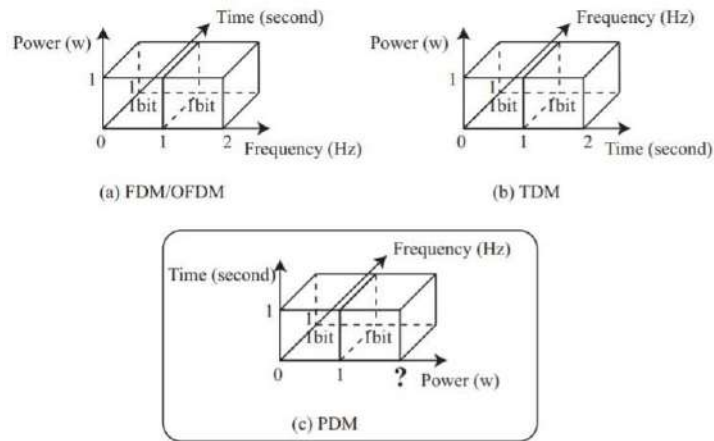


Figura 19-1: TDM, FDM, y PDM en el dominio de bit.

Fuente: Han, W. & Ma, X. 2016. p. 3.

Para demostrar la ortogonalidad de PDM, se realiza una transformación lineal de manera que se pueda enfocar en la asignación de diferentes segmentos de potencia. Considerando que B es el ancho de banda del canal y t es el tiempo, se puede establecer como s_1 y s_2 a los segmentos de potencia para dos flujos de información a través de la siguiente relación establecida por Han y Ma (2016).

$$c_1 = \log(1 + s_1) \Rightarrow s_1 = 2^{c_1} - 1$$

Ecuación 5-1:

Relación entre PS y flujo de información en el dominio de bit

Para asegurar la ortogonalidad entre los bits de diferentes flujos de información, la potencia transmitida a entregar es:

$$s_1 + s_2 = 2^{c_1+c_2} - 1$$

Ecuación 6-1:

Potencia transmitida en el dominio de bit

Los segmentos de potencia PS s_1, s_2 deben cubrir la necesidad de transmitir múltiples flujos de información en varias secuencias PS donde en cada secuencia se transmite n cantidad de bits. Según Han y Ma (2016, p.4), las secuencias PS cuentan con la propiedad de aditividad demostrada mediante una proposición y un corolario con el propósito de combinar dinámicamente múltiples PS para los flujos de información.

1.3.4.2. Estructura de OPDMA

El sistema de transmisión con OPDMA se basa en PDM, a diferencia que se consideran diferentes ganancias de canal h_M para $m = 1, 2, \dots, M$ usuarios en el proceso de división de segmentos de potencia y localización (Han, et.al, 2016, p. 5). El diagrama de bloques de un enlace ascendente con OPDMA se puede apreciar en la figura 20-1.

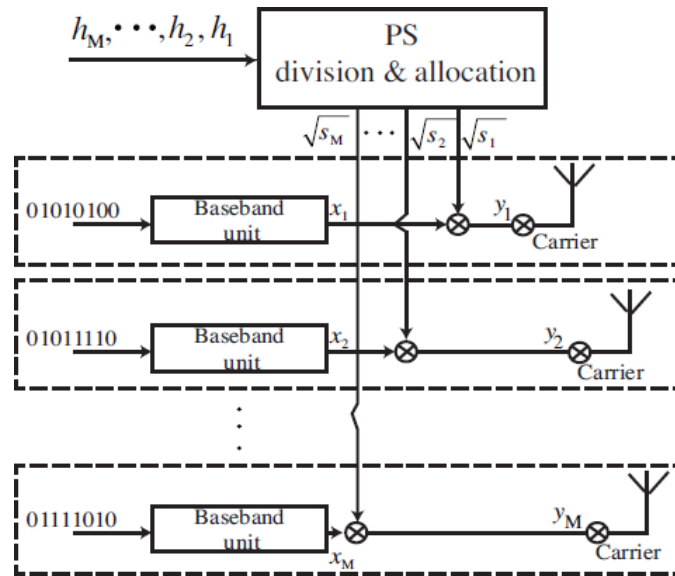


Figura 20-1: Enlace ascendente con OPDMA.

Fuente: Han, et.al, 2016, p. 6.

Los símbolos producto de la modulación de cada usuario x_1, x_2, \dots, x_M son multiplexados con diferentes segmentos de potencia denotado por $\sqrt{(s_1)}, \sqrt{(s_2)}, \dots, \sqrt{(s_M)}$, que están determinados por la unidad de división y asignación de PS. De manera general, se puede expresar matemáticamente la señal de salida del transmisor con la siguiente ecuación.

$$y = \sum_{m=1}^M \sqrt{s_m} x_m$$

Ecuación 7-1:

Señal para transmitir con OPDMA

Para la recepción, se puede incorporar directamente un cancelador de interferencias sucesivas SIC para la detección de los diferentes flujos de información.

1.3.4.3. División de Segmentos de potencia y estrategia de localización

Debido a la aleatoriedad de un canal inalámbrico, es posible que la estación base no conozca las ganancias de canal precisas. Bajo esta situación, se requiere que los PS relacionados se establezcan más grandes para garantizar la velocidad de transmisión requerida por el usuario (Han, et.al, 2016, p. 8). Se añade un factor de acumulación β_m a la ganancia de canal h_m , el cual se ajusta entre 0 y 1 para incrementar o disminuir el valor del segmento de potencia PS. En base a esta metodología, se propone una estrategia óptima de división y asignación de PS (OPS) de OPDMA, cuyos procedimientos se presentan en forma de algoritmo.

Tabla 1-1: Algoritmo del proceso de división de PS y asignación al usuario.

Procedimiento de la estrategia OPS	
Paso i)	Clasificar las ganancias de M canales en orden descendente para la transmisión en Downlink. $h_1 \geq h_2 \geq \dots \geq h_M$ Clasificar las ganancias de M canales en orden ascendente para la transmisión en Uplink. $h_1 \leq h_2 \leq \dots \leq h_M$ Establecemos $h[1:M] = \beta_{1:M} h_{1:M}$
Paso ii)	For $m = 1:1:M$ $s[m] = \left(2^{\frac{c}{B}} - 1\right) \left(\frac{\sigma^2}{h[m]} + s _1\right)$ If $ s _1 > S_{max}$ $s[l] = 0,$ Regresar al paso 2, end end
Paso iii)	Asignar los PS para acceder a los usuarios de acuerdo con la s.

Fuente: Han, et.al, 2016, p. 8

Realizado por: Armas, Roger, 2021.

En el paso i) la clasificación de las ganancias es necesaria para asegurar que la estrategia propuesta pueda utilizar suficientemente el canal degradado. En el paso ii) se calcula el segmento de potencia y en base al ancho de banda B, la cantidad de bits de información a transmitir c, potencia de ruido σ^2 , la ganancia del canal h y el PS para el primer usuario. Por último, la sentencia if se usa para controlar los PS calculados que satisfacen la restricción de potencia de transmisión (Han, et.al, 2016, p. 8).

1.4. Canal de comunicación

En las simulaciones de las técnicas de acceso al medio para transmitir información a través de canales físicos es importante determinar el medio por el cual viajarán las señales de información, ya sea alámbrica o inalámbricamente. La naturaleza de cada modelo de canal de comunicaciones, en este caso, inalámbricas, afecta al desempeño de la transmisión de datos introduciendo ruido o interferencias generando errores en la información. Debido a que estas limitaciones dependen del canal, se procede a analizar el canal AWGN (Additive White Gaussian Noise), cuyo ruido generado bajo este concepto se modela la mayoría de los casos de comunicación en un canal real (Arrieta, 2020, p. 24).

1.4.1. Canal AWGN

La comunicación por medio de un canal con ruido aditivo tiene un modelo matemático muy simple y fácil de comprender. En la figura 21-1 se observa el proceso donde se corrompe una señal transmitida $s(t)$ añadiendo el ruido $n(t)$. Dicho puede surgir de componentes electrónicos y amplificadores en el receptor del sistema de comunicación o de interferencias encontradas en la transmisión (como en el caso de la transmisión de señales de radio). Al introducido principalmente por componentes electrónicos y amplificadores en el receptor se denomina ruido térmico. Este tipo de ruido se caracteriza estadísticamente como un proceso de ruido gaussiano. Por lo tanto, el modelo matemático resultante para el canal generalmente se denomina canal de ruido gaussiano aditivo AWGN (Proakis y Salehi, 2007, p. 160).

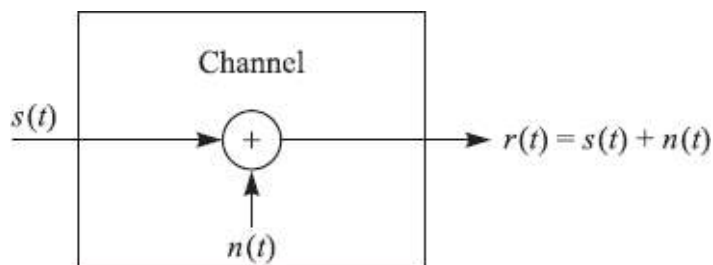


Figura 21-1: Diagrama de canal de ruido aditivo

Fuente: Proakis y Salehi, 2007, p. 161,

1.4.2. Canal Rice

Es un modelo de canal con distribución estadística Rice que viene caracterizada por una línea de vista entre el transmisor y receptor con un nivel de potencia del trayecto directo superior al de los

ecos producidos por el efecto multitrayecto. Usualmente se utiliza para escenarios en exteriores donde existe LOS.

1.4.3. Canal Rayleigh

Se denomina modelo de canal Rayleigh al canal donde las componentes de cada trayectoria de la señal transmitida no se pueden descomponer en elementos más simples y además tienen una distribución normal con media nula lo cual asegura el comportamiento de la amplitud de la señal como una distribución de Rayleigh. Es implementado en interiores debido a que no existe una componente principal de línea directa.

1.5. Parámetros de evaluación

1.5.1. Bit Error Rate (BER)

El BER es una medida que se desprende del número de bits errados con respecto al total de datos enviados por el transmisor en un intervalo de tiempo. Este parámetro permite determinar y evaluar el rendimiento de un esquema de modulación o de un canal en el sistema de comunicación.

1.5.2. Signal-to-Noise Ratio (SNR)

El SNR se define como la relación de potencia de señal útil respecto a la potencia de ruido añadido en un canal de comunicación. Se considera como una medida de rendimiento en dB del sistema el cual permite realizar comparaciones con sistemas de diferentes componentes.

1.5.3. Energy per-Bit to Noise Power Density Ratio (Eb/No)

El Eb/No es un parámetro que se define como la relación entre la energía de bit y la densidad de potencia de Ruido, el cual es utilizado para indicar qué tan robusto es un esquema de modulación o sistema bajo el efecto de un canal de comunicación a través de comparaciones con la tasa de errores de Bit o BER.

CAPITULO II

2. MARCO METODOLÓGICO

El desarrollo de un simulador de interfaz gráfica de las técnicas OMA y NOMA constituye un proyecto relevante para el estudio de las nuevas técnicas de acceso al medio que son propuestas para su implementación en tecnologías de comunicación inalámbrica de nueva generación. El simulador tiene como propósito servir de herramienta para el aprendizaje de las nuevas técnicas con el fin de comprender el funcionamiento y rendimiento de estas a comparación con las técnicas convencionales. El simulador fue instalado en los ordenadores del laboratorio de comunicaciones de la FIE y se proporcionó una aplicación ejecutable para ser instalada por los estudiantes y docentes.

2.1. Diseño de la investigación

2.1.1. Tipo de investigación

Para el presente trabajo de titulación se empleó el tipo de investigación aplicada enfocada al área científica debido a que tiene por objetivo replicar los conocimientos adquiridos durante la carrera de ingeniería en Telecomunicaciones en cuanto a las técnicas de acceso al medio ortogonales e implementarlas en un simulador junto a las técnicas NOMA las cuales fueron objeto de investigación.

2.1.2. Métodos de investigación

Los métodos de investigación aplicados en el trabajo de titulación fueron los métodos analítico-sintético e inductivo-deductivo los cuales infieren de la siguiente manera:

- Método analítico-sintético

Se utilizó el análisis en el método analítico-sintético para descomponer mentalmente un todo en sus partes y cualidades, en sus múltiples relaciones, propiedades y componentes (Rodríguez & Pérez, 2017, p. 186) los cuales son necesarios para replicar el comportamiento de las técnicas de acceso al medio ortogonales y no ortogonales en el simulador experimentalmente.

Por otro lado, la síntesis establece mentalmente la unión o combinación de las partes previamente analizadas y posibilita descubrir relaciones y características generales entre los elementos de la realidad (Rodríguez & Pérez, 2017, p. 186). Esta síntesis se realizó en la recopilación de información necesaria para desarrollar el simulador de las técnicas de acceso al medio ortogonales y no ortogonales uniendo cada componente de las mismas.

- Método inductivo-deductivo

Este método se basa en dos procesos análogos. La inducción es una forma de razonamiento en la que se pasa del conocimiento de casos particulares a un conocimiento más general (Rodríguez & Pérez, 2017, p. 187), lo cual ayudó a establecer los procesos y parámetros a comparar entre las técnicas OMA y NOMA con el fin de obtener las conclusiones.

Por otro lado, el razonamiento deductivo permite pasar de un conocimiento general a otro de menor nivel de generalidad, es decir, un conocimiento particular (Rodríguez & Pérez, 2017, p. 187), por lo que ayudó a generar las conclusiones y recomendaciones del presente trabajo de titulación.

2.1.3. Técnicas de Investigación

Para el desarrollo del trabajo de titulación se tomó en cuenta las técnicas de investigación de observación y la experimentación con el fin de recopilar la información necesaria para comprender el funcionamiento de las técnicas y simularlas.

- Mediante la observación se puede ampliar el panorama sobre el principio de funcionamiento de cada componente que conforma una técnica de acceso al medio ortogonal y no ortogonal.
- El proceso de simular las técnicas OMA y NOMA planteadas requieren de realización de pruebas y mejoramiento de los errores para lo cual, la experimentación es útil para la optimización de las simulaciones y sus resultados en ambientes controlados.

La información recopilada es obtenida de la revisión de fuentes primarias y secundarias como artículos de revistas científicas.

2.2. Determinación de los procesos del simulador

En esta sección se explica la simulación mediante interfaz gráfica de las técnicas de acceso múltiple ortogonales CDMA, OFDMA y no ortogonales SCMA y OPDMA, las cuales se

desarrollaron mediante líneas de códigos de MATLAB. A la vez se explica las indicaciones de uso del simulador para que el usuario del programa pueda hacer uso de este.

El simulador mediante su interfaz gráfica ofrece el ingreso de los parámetros para cada técnica de acceso y la presentación de los resultados. Dentro de los parámetros generales que se ingresan está el número de usuarios, la cantidad de datos transmitidos y el tipo de canal sobre el cual se transmitirá la información multiplexada. Por otro lado, en la presentación de resultados el usuario puede especificar los parámetros para la visualización de las curvas de BER en función del SNR.

Para cada técnica de acceso se determinaron los procesos acordes a la teoría planteada en las fuentes bibliográficas, cuya implementación en el simulador se detalla en diagramas de flujo. Las curvas visualizadas en las simulaciones se obtuvieron con el fin de analizar y comparar el comportamiento de las técnicas de acceso al medio con diferentes canales de transmisión y esquemas de modulación.

2.2.1. Diagrama de flujo de la interfaz gráfica

En la figura 1-2 se puede apreciar el diagrama de la interfaz gráfica. Al iniciar el programa se despliega la venta inicial en la cual se visualiza parte de la portada del trabajo de titulación e indica el autor de este como se puede visualizar en la figura 2-2. El diagrama de flujo refleja el procedimiento que debe realizar el usuario para llevar a cabo las simulaciones de las técnicas de acceso múltiple ortogonales (OMA) y no ortogonales (NOMA).

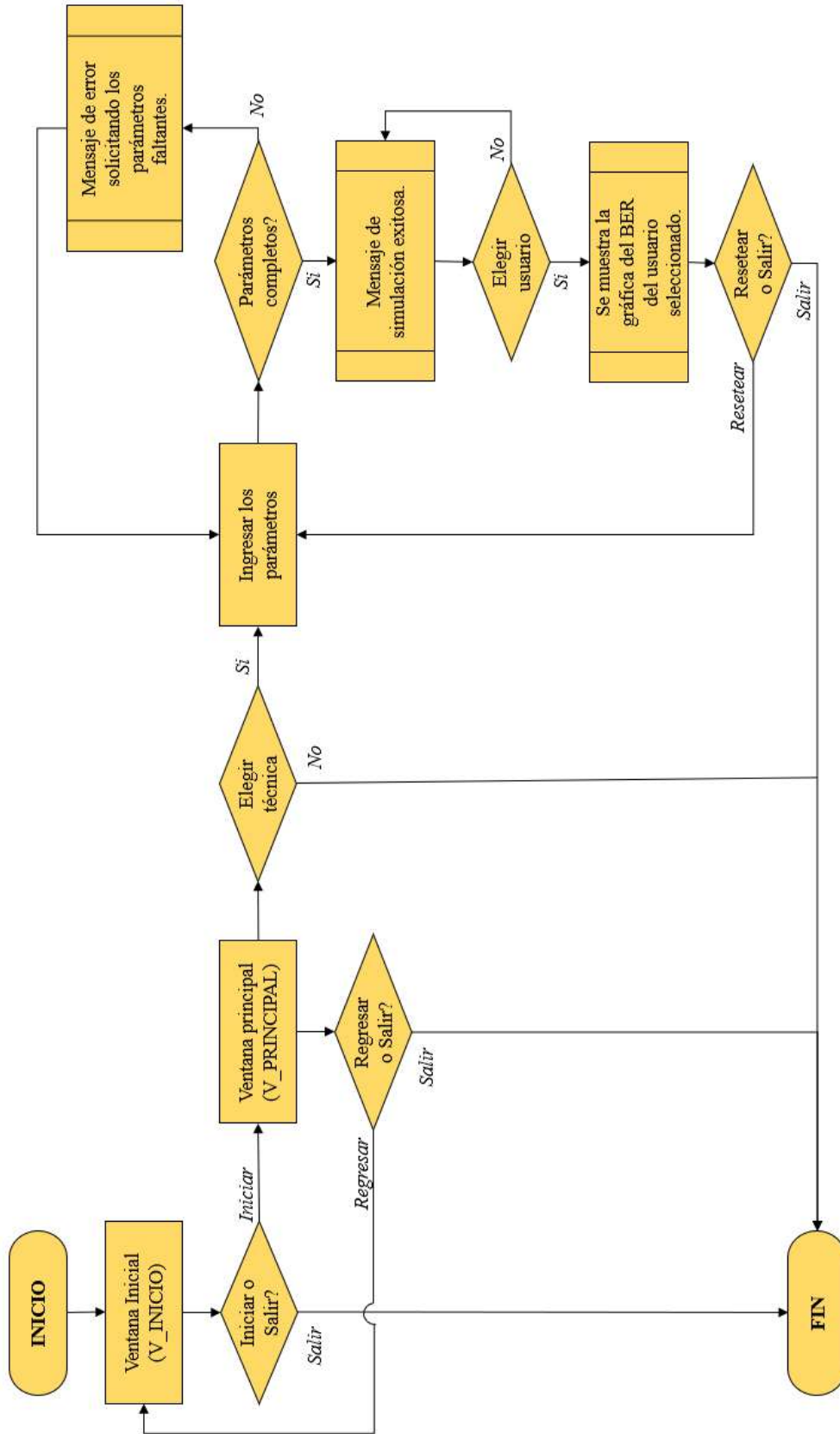


Figura 1-2: Diagrama de flujo del simulador.

Realizado por: Armas, Roger, 2021.



Figura 2-2: Ventana de inicio del simulador.

Realizado por: Armas, Roger, 2021.

En la ventana de inicio el usuario puede seleccionar si desea iniciar el programa para la simulación de las técnicas o salir del simulador presionando el botón cerrar de la esquina superior derecha. Si se selecciona el botón Iniciar se desplegará la ventana principal la cual se muestra en la figura 3-2.

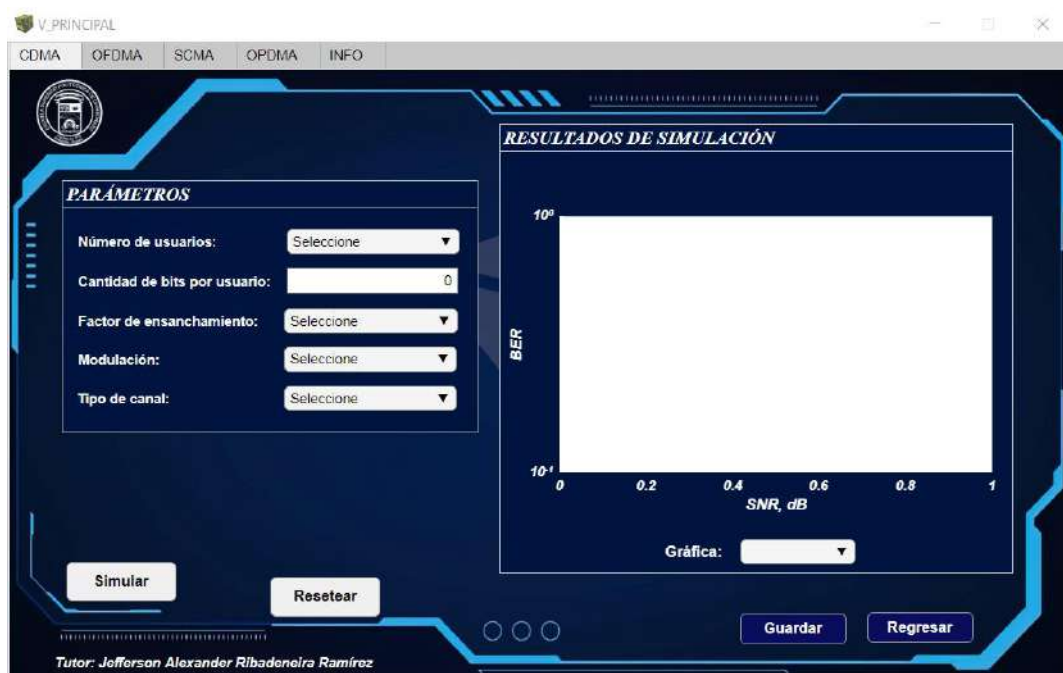


Figura 3-2: Interfaz de simulación de la técnica CDMA.

Realizado por: Armas, Roger, 2021.

En la ventana principal se encuentran las técnicas de acceso al medio separadas por pestañas para un mejor despliegue de sus parámetros. En cada Pestaña se despliegan dos paneles los cuales corresponden a los parámetros de simulación que el usuario debe ingresar y el otro a los resultados de la simulación, es decir, la gráfica de la curva BER en función del SNR. Al oprimir el botón de Simular, inicia los cálculos para posteriormente mostrar la gráfica en el panel de resultados. Por otro lado, cuando se oprime el botón Resetear los parámetros vuelven a los valores por defecto que fueron configurados. La primera técnica en mostrarse es CDMA, donde se deben ingresar los parámetros mostrados en la figura 3-2. Una vez ingresados se presiona el botón simular y posteriormente se despliega un mensaje de espera mientras se realizan los cálculos y por último se muestra un mensaje cuando la simulación se ha realizado con éxito los cuales se pueden visualizar en la figura 4-2.

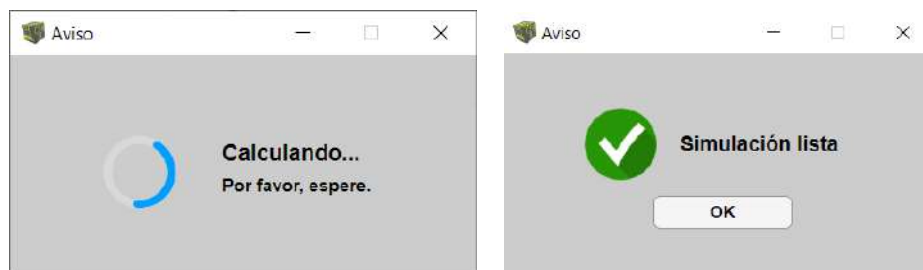


Figura 4-2: Mensaje de aviso en caso de espera y cuando la simulación está realizada.

Realizado por: Armas, Roger, 2021.

Si no se configura los parámetros y se presiona el botón Simular, se despliega automáticamente un mensaje de error con el aviso de seleccionar los parámetros faltantes.



Figura 5-2: Mensaje de error cuando no se ingresa un parámetro de simulación.

Realizado por: Armas, Roger, 2021.

2.2.2. Diagrama de flujo de los procesos de las técnicas OMA y NOMA

El procesamiento general de las señales para las técnicas OMA y NOMA simuladas se realizó como se observa en la figura 6-2. Cabe recalcar que el diagrama representa la transmisión de datos en un enlace ascendente. Los datos correspondientes a los usuarios fueron generados aleatoriamente para luego procesarlos en el transmisor y se sometieron a los efectos del canal

inalámbrico. Mediante iteraciones se pudo obtener el comportamiento de las técnicas mediante diferentes niveles de SNR para calcular la tasa de errores de bits. Finalmente se grafica las curvas BER en escala logarítmica.

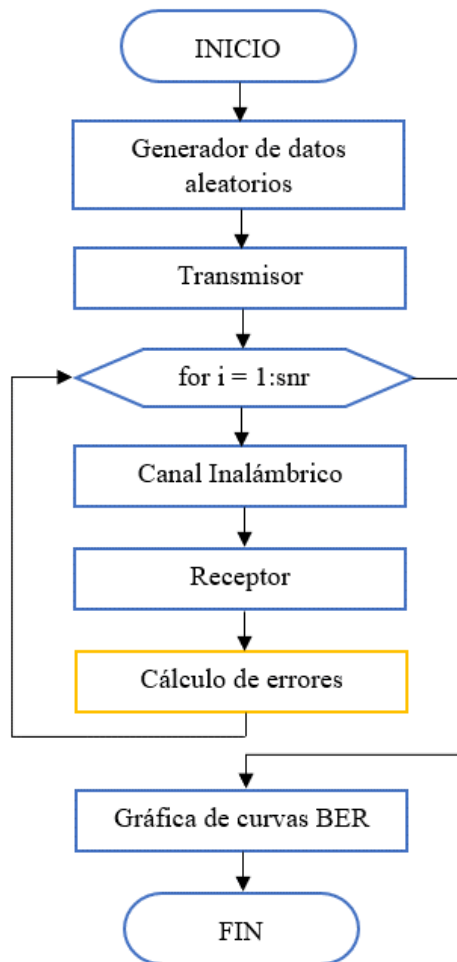


Figura 6-2: Diagrama de flujo de la simulación de cada técnica.

Realizado por: Armas, Roger, 2021.

A continuación, se explican los procesos de cada etapa de la simulación de las técnicas. Los códigos completos correspondiente a cada simulación se encuentran en el anexo A.

2.2.3. Simulación de CDMA

En la técnica de acceso al medio ortogonal CDMA, fueron definidos diferentes bloques de procesamiento de las señales como se muestra en la figura 6-2, donde el sistema se encarga de generar aleatoriamente los bits de información de acuerdo con el número de usuarios para luego proceder al ensanchamiento mediante secuencia directa, modulaciones MPSK o MQAM y suma de las señales. En esta simulación no se consideró métodos de corrección de errores como codificadores con el objetivo de mostrar el rendimiento únicamente de la técnica de espectro ensanchado respecto a las demás técnicas OMA y NOMA. La implementación de CDMA en MATLAB se desarrolló de forma secuencial sin la utilización de funciones individuales a fin de que se pueda observar cada proceso de manera eficiente. La etapa de transmisión se muestra en la figura 7-2. El proceso de recepción es el inverso del mismo.

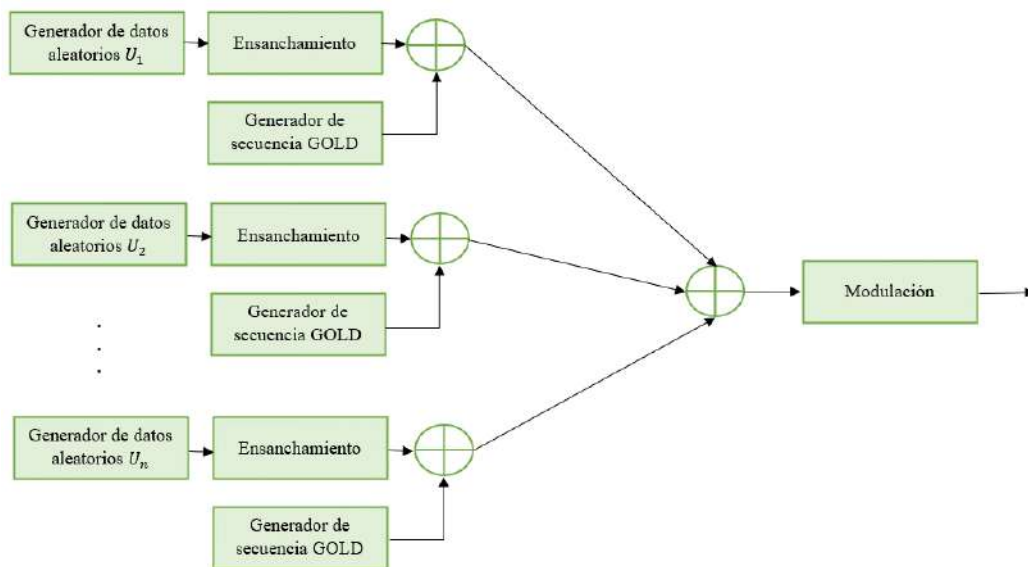


Figura 7-2: Sistema de transmisión CDMA.

Realizado por: Armas, Roger, 2021.

Los parámetros que a ingresar mediante la interfaz gráfica se observan en la tabla 1-2.

Tabla 1-2: Parámetros de entrada para la simulación de CDMA.

Parámetro de entrada	Descripción
NU	Número de usuarios
Nbits	Cantidad de bits generados aleatoriamente para cada usuario.
SF	Factor de ensanchamiento para cada usuario.

Mod	Modulación de las señales (QPSK, BPSK, 16-QAM y 64-QAM)
-----	---

Realizado por: Armas, Roger, 2021.

2.2.3.1. Transmisor

Para la simulación del transmisor CDMA, se consideró los siguientes procesos:

- Generador de bits aleatorios

Los datos son generados con la función `randn` misma que permite obtener valores aleatorios enteros entre un intervalo definido (0 y 1 para obtener bits). Los bits de cada usuario son guardados en un vector con dimensiones `Nbits` filas y `NU` columnas como se muestra a continuación:

```

%% Generador de las señales de
usuariosNU = 6; % Número de usuarios
Nbits = 20000; % Cantidad de bits por usuario
data = randi([0 1], Nbits, NU); % Generador de datos aleatorios

```

- Ensanchamiento

Para el proceso de ensanchamiento se tomó en cuenta el valor de factor de ensanchamiento `SF` para cada usuario y mediante una estructura `FOR` y el manejo de los índices de los vectores, se expandió cada bit acorde al valor de `SF`.

```

for u = 1:NU
    for i = 0:length(data)-1
        if data(i+1,u) == 1
            dataE(i*SF(u)+1:SF(u)*(i+1),u) = 1; % Ensanchamiento si el
bit es 1
        elseif data(i+1,u) == 0
            dataE(i*SF(u)+1:SF(u)*(i+1),u) = 0; % Ensanchamiento si el
bit es
        end
    end
end

```

- Generador de secuencias GOLD

Las secuencias GOLD se pueden generar mediante un objeto que forma parte del toolbox de comunicaciones de MATLAB `comm.GoldSequence` cuyos argumentos de entrada son los polinomios primitivos para cada LFSR, la palabra semilla, el índice `ind` y la cantidad de bits de secuencias acorde al factor de ensanchamiento `SF`.

```

for u = 1:NU
    PNseq = comm.GoldSequence('FirstPolynomial','x^25+x^3+1',... %
Polinomio 1

```

```

        'FirstInitialConditions', [1 1 0 0 zeros(1, 20) 1], ... %
Secuencia semilla 1
        'SecondPolynomial', 'x^25+x^3+x^2+x+1',... % Polinomio 2
        'SecondInitialConditions', [1 1 0 0 zeros(1, 20) 1], ...
% Secuencia semilla 2
        'Index', ind(u), 'SamplesPerFrame', SF(u)); % Indice de la
secuencia
        spreadseq(1:SF(u), u) = PNseq(); % Obtención de la secuencia PN
end

```

- Multiplicación de la secuencia GOLD con los datos ensanchados

Mediante la función xor se pudo aplicar las secuencias GOLD para cada intervalo SF de los datos ensanchados lo cual es controlado mediante la estructura FOR.

```

%% XOR
for u = 1:NU
    for i=0:length(data)-1
        dataspread(i*SF(u)+1:SF(u)*(i+1), u) =
double(xor(dataE(i*SF(u)+1:SF(u)*(i+1), u), spreadseq(1:SF(u), u)));
    end
end

```

- Suma de señales

La suma de las señales se realizó a nivel de bit donde se coloca un bit de cada usuario alternadamente, generando un solo vector de bits la cual simula a la adición de todos datos generados por cada usuario.

```

%% SUMA DE SEÑALES
for i=0: length(dataspread)-1
    for u = 1:NU
        SS(i*NU+u,1)=dataspread(i+1, u);
    end
end

```

- Modulación

Se aplicó la modulación al vector de salida del proceso anterior y se elige el esquema de modulación a aplicarse acorde al valor del parámetro Mod. Para la modulación BPSK y QPSK se utilizó objetos como comm.BPSKModulator y comm.QPSKModulator donde se configuró a la entrada tipo bits en ambos esquemas y un desfase de 90 y 45 grados respectivamente. En el objeto de modulator QPSK el mapeo de bits para las constelaciones se configuró por defecto con códigos gray. Por otro lado, para las modulaciones QAM se aplicó la función qammod donde se debe ingresar los argumentos de entrada como los datos de usuario, el orden de modulación (16 y 64) y el tipo de entrada.

```

%% Modulación
switch Mod

```

```

case 1
    bpskModulator = comm.BPSKModulator;
    bpskModulator.PhaseOffset = pi/2;
    SSmod = bpskModulator(SS);
case 2
    QpskModulator = comm.QPSKModulator;
    QpskModulator.PhaseOffset = pi/4;
    QpskModulator.BitInput = 1;
    SSmod = QpskModulator(SS);
case 3
    SSmod = qammod(SS,16,'InputType','bit');
case 4
    SSmod = qammod(SS,64,'InputType','bit');
End

```

2.2.3.2. Receptor

En la etapa de recepción se aplicó el proceso inverso a la transmisión de los datos. La señal luego de pasar por el canal inalámbrico es demodulada directamente y luego procesada para obtener los datos generados por los usuarios. El procedimiento de recepción de los datos se realizó tomando en cuenta los siguientes pasos:

- Ecuación

La señal recibida se encuentra multiplicada por la ganancia del canal inalámbrico, por lo tanto, el proceso de ecualización consta en dividir la señal afectada por el canal con la ganancia del canal h la cual es de conocimiento en el receptor.

```

% Ecuación
s_rx = sigcanal./h';

```

- Demodulación

De la misma manera que en la modulación, se utilizó los objetos `comm.BPSKDemodulator` y `comm.QPSKDemodulator` con las mismas especificaciones en los argumentos que en la modulación para que no existan errores en los bits. Por otro lado, para la demodulación QAM se utilizó la función `qamdemod` con parámetros de entrada la señal del canal, el orden de la demodulación y el tipo de salida de datos.

```

% DEMODULACIÓN
switch Mod
case 1
    bpskDemodulator = comm.BPSKDemodulator; % Objeto para
demodulador BPSK
    bpskDemodulator.PhaseOffset = pi/2; %Desfase de 90 grados
    SSdemod = bpskDemodulator(s_rx); % Aplica el demodulador en la
señal del canal
case 2
    QpskDemodulator = comm.QPSKDemodulator('BitOutput',
true,'PhaseOffset', pi/4, 'SymbolMapping', 'Gray');% Objeto para
demodulador QPSK

```

```

        SSdemod = QpskDemodulator(s_rx); % Aplica el demodulador en la
señal del canal
        case 3
            SSdemod = qamdemod(s_rx,16, 'OutputType', 'bit'); % Demodula la
señal con salida tipo bit
        case 4
            SSdemod = qamdemod(s_rx,64, 'OutputType', 'bit'); % Demodula la
señal con salida tipo bit
        end
end

```

- Separación de señales

Este proceso permite separar los bits de cada usuario obteniendo su señal ensanchada. De la misma manera que en la suma de señales, el mapeo de los bits y separación se implementó mediante estructuras for para la navegación entre los índices de los vectores.

```

for j =0: length(dataspread)-1
    for u = 1:NU % Se repite para cada usuario
        dataspreadrx(j+1,u)=SSdemod(j*NU+u,1); % Se separa cada bit de
la señal demodulada en un vector para cada usuario
    end
end
end

```

- Desensanchamiento

Finalmente se realizó el desensanchamiento de los flujos de datos mediante un reconocimiento de un intervalo de elementos del vector en valores de 1 o 0, obteniéndose como resultado un vector de recepción de los datos mediante el uso de estructuras de repetición y condicionales.

```

datarx=zeros(Nbits,NU);
for u = 1:NU % Se repite para cada usuario
for j = 0:length(data)-1
    if datadespread(j*SF(u)+1,u) == ones(1,SF(u)) % Reconoce a un
intervalo de bits como 1
        datarx(j+1,u) = 1; % Obtención del bit 1 desensanchado
    elseif datadespread(j*SF(u)+1,u) == zeros(1,SF(u)) % Reconoce a un
intervalo de bits como 1
        datarx(j+1,u) = 0; % Obtención del bit 0 desensanchado
    end
end
end
end

```

2.2.4. Simulación OFDMA

Para el proceso de simulación de la técnica OFDMA se consideró las etapas de transmisión y recepción mostrados en la figura 7-1. El diagrama de bloques del transmisor OFDMA propuesto para su simulación se presenta en la figura 8-2, donde se omitió la sección de conversión de serie a paralelo debido a la facilidad que ofrece MATLAB para trabajar con los datos en vectores columna lo cual evita el proceso de conversión mencionado anteriormente. En el receptor los procesos ocurren de manera inversa a la del transmisor a excepción del conteo de símbolos y preprocesamiento de trama.

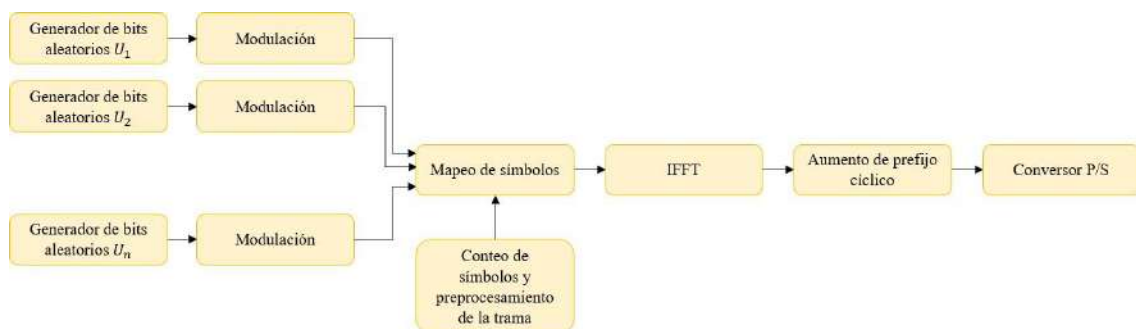


Figura 8-2: Diagrama de bloques para la simulación del transmisor OFDMA.

Realizado por: Armas, Roger, 2021.

Los parámetros de entrada necesarios para la simulación se muestran en la tabla 2-2

Tabla 2-2: Parámetros de entrada para simulación de OFDM.

Parámetro de entrada	Descripción
NU	Número de usuarios
Nbits	Cantidad de bits generados aleatoriamente para cada usuario.
Mod	Tipo de modulación
t_fft	Tamaño de la FFT
t_pc	Duración del prefijo cíclico

Realizado por: Armas, Roger, 2021.

2.2.4.1. Transmisor

En el bloque de transmisor OFDMA se consideran los siguientes procesos:

- Generador de bits aleatorios

De la misma manera que en CDMA, se utilizó la función randn para la obtención de números enteros aleatorios dentro de un rango establecido. Para generar datos binarios el rango es de 0 y

1. Se estableció el número de usuarios y la cantidad de bits a generar.

```
% Generador de bits
aleatorios NU = 6; % Número
de usuarios Nbits = 20000;
% Número de bits

data = randi([0 1],
Nbits,NU);
```

- Modulación

Los bits generados aleatoriamente fueron modulados directamente con el fin de obtener los símbolos que serán mapeados en el proceso posterior. De igual manera que en CDMA, la modulación BPSK y QPSK son generadas mediante los objetos comm.BPSKModulator y comm.QPSKModulator donde se configuró a la entrada tipo bits en ambos esquemas y se configura un desfase de 90 y 45 grados respectivamente. En el objeto de modulator QPSK el mapeo de bits para las constelaciones está configurado por defecto con códigos gray. Por otro lado, para las modulaciones QAM se aplicó la función qammod donde se ingresa los argumentos de entrada como los datos de usuario, el orden de modulación (16 y 64) y el tipo de entrada que en el caso de este trabajo es de tipo bit.

```
%% Modulación
switch Mod
case 1
for u = 1:NU
bpskModulator = comm.BPSKModulator;
bpskModulator.PhaseOffset = pi/2;
Smod(:,u) = bpskModulator(data(:,u));
end
case 2
for u = 1:NU
QpskModulator = comm.QPSKModulator;
QpskModulator.BitInput = 1;
Smod(:,u) = QpskModulator(data(:,u));
end
case 3
for u = 1:NU
Smod(:,u) = qammod(data(:,u),16,'InputType','bit');
end
case 4
for u = 1:NU
Smod(:,u) = qammod(data(:,u),64,'InputType','bit');
end
end
```

En cada caso de la estructura Switch se realizó la modulación en base a una estructura for la cual tiene como función modular los datos para cada usuario del vector data.

- Conteo de símbolos y preprocesamiento de la trama

El conteo de símbolos es esencial para el cálculo de los números de bloques de recursos (RB) necesarios para cada usuario. Mediante la función length se pudo encontrar el número de símbolos totales correspondiente a cada usuario y se guardó en un vector N_simb.

```
%% Calculo de los numeros de simbolos por usuario
for u = 1:NU
    N_simb(u) = length(Smod);
end
```

El preprocesamiento de la trama consistió en la creación de una trama inicial que consta de N subportadoras y 140 símbolos agrupados en 20 slots. Luego se procedió a calcular la cantidad de RBs necesarias para cada usuario dividiendo el número total de símbolos entre 84 cuyo valor corresponde al número total de símbolos que contiene un RB. La función ceil de Matlab permitió redondear el valor al inmediato superior. Finalmente se obtuvo el total de bloques de recursos mediante una sumatoria de los bloques de cada usuario y se realizó un aumento de trama si este valor supera al máximo especificado Nref.

```
%% Preparación de la trama
Rgrid = zeros(subp,140);
% Conteo de RB para cada usuario
for u = 1:NU
    RB(u) = ceil(N_simb(u)/84);
end
NT = sum(RB); % Suma de los RBs
if NT <= Nref % Si el número de RBs es menor a Nref, se transmite solo
una trama
    trama=Rgrid;
    Num = 1;
else % Si el número de RBs es mayor a 500, se transmite Num
tramas
    Num=ceil(NT/Nref); % Obtención del número de tramas
    trama = Rgrid;
    for i = 2:1:Num
        trama = [trama Rgrid];
    end
end
[f,c] = size(trama); % Se obtiene las dimensiones de la trama
```

- Mapeo de símbolos

En OFDMA los recursos fueron asignados en base a la demanda de tráfico que se genera en un sistema de comunicación. Para obtener esta asignación se ha recurrido a la aleatoriedad de asignación mediante la generación de una secuencia de posiciones correspondientes a los RBs en la matriz trama. La función randperm permitió generar un vector que contiene la permutación de

números enteros que van desde 1 hasta N sin repetir elementos. Luego se obtuvo los RBs aleatorios para cada usuario mediante una estructura for.

```
% Generacion de # RB aleatorio para asignación
RBsA = randperm(Nref*Num);
cont = 1;
for u = 1:NU % Asignación de los RBs Aleatorios a cada usuario
    RBAu(:,u) = RBsA(cont:RB(u)*u);
    cont = cont + RB(u);
end
```

El mapeo de usuarios se realizó mediante estructuras for e if anidadas con el objetivo de desplazarnos fácilmente en la matriz trama y asignar los símbolos de cada usuario en cada RB. Para la localización de los RBs se aplicó una ecuación obtenida empíricamente la cual en base al RB aleatorio generado aleatoriamente RBAu, el número total de símbolos de la trama c, y el número total de trama Num. La variable k es un contador de símbolos de cada usuario.

```
% Mapeo de usuarios en la trama
for u = 1:NU % Desplazamiento entre usuarios
    k=1;
    for I = 1:RB(u) % Desplazamiento entre elementos del vector RB
        for i = 1:12:subp % Desplazamiento entre portadoras
            if i == (ceil(RBAu(I,u)/(c/7))-1)*12+1 %Localización de la
subportadora
                for j = 1:7:c % Desplazamiento entre simbolos
                    if j == ((RBAu(I,u) - (ceil(RBAu(I,u)/(c/7))-
1)*(20*Num))-1)*7+1 % Localización del slot
                        for l = 0:1:11 % Desplazamiento entre portadoras
                            for h = 0:1:6 % Desplazamiento entre slots
                                if k <= N_simb(u) % Control de asignación
de símbolos
                                    trama(i+1,j+h) = Smod(k,u);
                                %Asignación del simbolo en la trama
                                    k = k+1;
                                else
                                    break
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
end
end
end
end
end
end
end
```

- IFFT

En este bloque se calculó la IFFT mediante la función ifft de Matlab donde se ingresó las subportadoras correspondientes a cada símbolo de la trama. En la función se especifica el tamaño de la ifft mediante la variable t_ifft la cual es un parámetro inicial de la simulación. Posterior a eso se realizó la normalización respectiva acorde al número de portadoras.

```

for i=1:numel(trama)/subp
    s_ifft(i,:)=ifft(trama(:,i),t_fft); % Calculo de la Ifft cada 140
portadoras
end
s_ifft=sqrt(subp)*s_ifft; % Normalización de la IFFT

```

- Adición del prefijo cíclico

Para simular este proceso de una manera generalizada, se calculó la duración del prefijo cíclico pc como una fracción del símbolo y es añadida al principio de cada uno de los mismos.

```

pc = ceil(t_fft*t_pc);
for l=1:1:((numel(trama))/subp) %Agrega CP de 40 unidades de tiempo a
cada slot
    s_cp(l,:)=[s_ifft(l,t_fft-(pc-1):t_fft) s_ifft(l,1:t_fft)];
end
[n1,n2] = size(s_cp);

```

- Conversor de Paralelo a Serie

El proceso de conversión se realizó mediante una estructura for la cual se utilizó para localizar cada símbolo y elemento de tiempo obtenido en la IFFT.

```

for i=0:1:((numel(trama))/subp)-1
    s_ps(i*n2+1:n2*(i+1))= s_cp(i+1,:);
end

```

2.2.4.2. Receptor

El receptor OFDMA contiene la cadena de procesos inversa a la del transmisor. La señal recibida afectada por el canal inalámbrico y fue procesada siguiendo los siguientes pasos:

- Ecuación

La señal recibida se encuentra multiplicada por la ganancia del canal inalámbrico, por lo tanto, el proceso de ecuación consta en dividir la señal afectada por el canal con la ganancia del canal h la cual es de conocimiento en el receptor.

```

% Ecuación
s_rx = sigcanal./h';

```

- Conversor de Serie a Paralelo

De manera similar al conversor de Paralelo a Serie, se utilizó estructuras for para crear la matriz donde se ubican los elementos de la IFFT en función de cada símbolo.

```

for i=0:1:((numel(trama))/subp)-1
    r_sp(i+1,:) = r_rx(i*n2+1:n2*(i+1));
end

```

- Remoción del prefijo cíclico

Para remover el prefijo cíclico se utilizó una estructura for para el desplazamiento de los símbolos y mediante el manejo de los índices de la matriz, se asignó los elementos del símbolo OFDM únicamente, descartando así el prefijo cíclico.

```

for l=1:1:((numel(trama))/subp) %Remueve el CP
de los símbolos
    r_cp(l,:)=r_sp(l,pc+1:n2);
end

```

- FFT

Matlab proporciona la función fft la cual permitió obtener la transformada rápida de Fourier de manera automática y para obtener una mejor precisión se utilizó la función roundn para redondear los símbolos afectados por el canal. Luego se dimensionó los valores obtenidos en una matriz con las mismas dimensiones que la trama inicial. Por último, se realizó la desnormalización de la trama.

```

%% Transformada FFT
for i=1:1:((numel(trama))/subp) %Calcula la FFT
    r_fft(:,i)=fft(r_cp(i,:));
end
r_fft=roundn(r_fft,-15); %Redondea a 15 decimales
for l=1:1:((numel(trama))/subp)
    for i=1:1:subp
        tramaR(i,l)=r_fft(i,l);%Adquiere RGrid con 300 subportadoras
    end
end
tramaR=(1/sqrt(subp)).*tramaR; % Desnormalización

```

- Demapeo de símbolos

La etapa de demapeo de símbolos es similar al mapeo ya que se utilizó las mismas estructuras for para la localización de los bloques de recursos y desplazamientos entre los símbolos. La matriz d_mod es la encargada de almacenar los símbolos correspondientes a cada usuario. Cabe recalcar que el proceso es iterativo en función del número de usuarios.

```

%% Demapeo
for u = 1:NU % Desplazamiento entre usuarios
k=1;
for I = 1:RB(u) % Desplazamiento entre elementos del vector RB
    for i = 1:12:subp % Desplazamiento entre portadoras
        if i == (ceil(RBAu(I,u)/(c/7))-1)*12+1 %Localización de la
subportadora

```


2.2.5. Simulación SCMA

La simulación de esta técnica NOMA se basa en los modelos propuestos en los artículos científicos citados en este proyecto el cual se puede visualizar en la figura 14-1. En las etapas de transmisión y recepción se estableció una codificación y decodificación SCMA, el cual sirve como mapeador de los bits para la asignación de las palabras de códigos. El presente trabajo de integración curricular no realizó el diseño de libros de códigos debido al nivel de dificultad del proceso, esto limitó la simulación de la técnica para una cantidad fija de 6 usuarios, los libros de códigos utilizados para cada usuario se pueden visualizar en el anexo B. El diagrama de bloques de la simulación de la técnica se muestra en la figura 9-2.

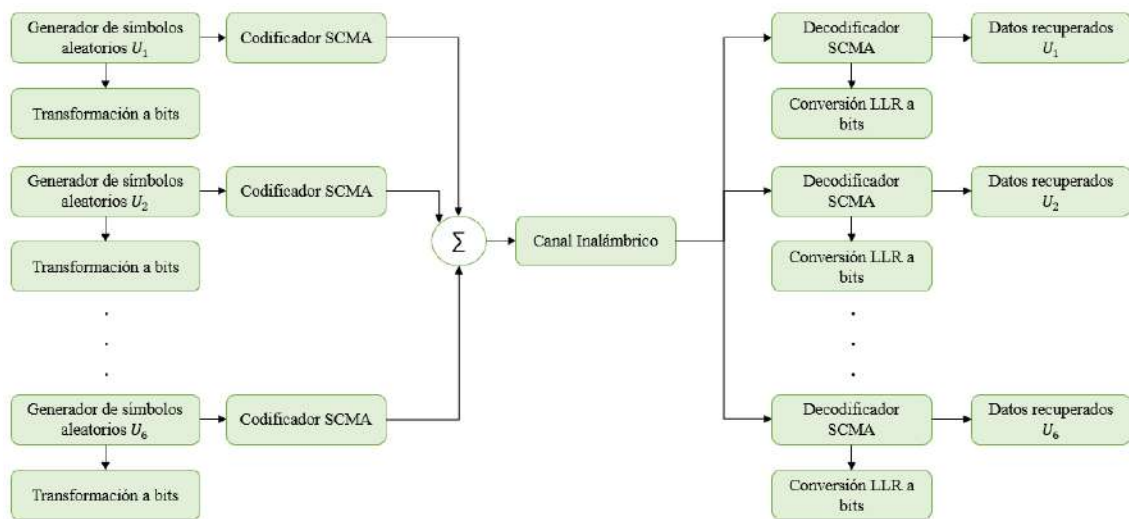


Figura 9-2: Diagrama de bloques de la simulación SCMA.

Realizado por: Armas, Roger, 2021.

En esta técnica no existen los procesos de modulación y demodulación de los datos, en su lugar, el codificador y decodificador SCMA tiene la función de transformar los bits en el dominio del tiempo mediante las palabras de código dispersas. Los parámetros de entrada necesarios para la simulación son escasos debido a la limitación anteriormente mencionada, por lo que solo se tomó en cuenta la cantidad de símbolos N_{simb} a generarse como variable ingresada por el usuario. Por otro lado, los parámetros iniciales de la simulación se definen en la tabla 3-2.

Tabla 3-2: Parámetros iniciales del sistema SCMA.

Parámetro inicial	Valor	Descripción
R	4	Número de recursos ortogonales.
CW	4	Número de palabras de código para cada libro.
U	6	Número de usuarios a simular.

Realizado por: Armas, Roger, 2021.

2.2.5.1. Transmisor

Para el desarrollo de la simulación SCMA, se consideró los siguientes procesos:

- Generador de símbolos aleatorios

Con el objetivo de facilitar el mapeo de los usuarios en el codificador SCMA, fueron generados símbolos aleatorios los cuales representan el equivalente decimal al conjunto formado por dos bits continuos. Mediante la función `randi` se pudo obtener valores en el rango de 0 a $CW-1$.

```
%% Generación de símbolos de log2(CW) bits
Nsimb = 10000;
x = randi([0 CW-1], U, Nsimb);
```

- Conversión de símbolo a bit

Se realizó la conversión de los símbolos con una transformación inicial decimal a binario donde cada símbolo da como resultado 2 bits. Como parámetro de control se configuró a la función `dec2bin` con el argumento “left-msb” para obtener el bit más significativo a la izquierda. Posteriormente se realizó una reconstrucción mediante la función `reshape` para la organización de los bits correspondientes a cada usuario.

```
r = dec2bin(x, log2(CW), 'left-msb'); % Transformación de decimal a
binario
data = zeros(log2(CW)*length(x), U); % Creación de la matriz data
for kk = 1:U
    data(:,kk) = reshape(reshape(r, U, kk-1).', [], 1); % Reconstruir
el vector r
end
```

- Codificador SCMA y multiplexación

El codificador SCMA tiene la función de mapear los símbolos de cada usuario. Mediante estructuras `for` se realizó la localización de las palabras de código correspondiente a cada usuario. En este proceso también se hizo la multiplexación de las palabras de código y se generó la señal a transmitirse por el canal inalámbrico.

```
pc = zeros(R, length(x));
for t = 1:length(x)
    for j = 1:U
        pc(:,t) = pc(:,t)+CB(:, x(j,t)+1, j).*h(:,j,t); %Suma de
las palabras de código de cada usuario con la ganancia de canal
    end
end
```

2.2.5.2. Receptor

La etapa de recepción cuenta con una complejidad computacional elevada debido al algoritmo de paso de mensajes MPA que se realiza como decodificador SCMA.

- Decodificador SCMA

Para el proceso de decodificación se empleó una función creada en Matlab como script cuyo nombre es `scmadec`, donde los parámetros de entrada de esta función se detallan en la tabla 4-2.

Tabla 4-2: Parámetros de entrada para la función `scmadec`.

Parámetro inicial	Descripción
y	Señal SCMA multiplexada y afectada por el canal.
CB	Libro de códigos para cada usuario.
No	Varianza de ruido (Canal AWGN).
Nit	Número de iteraciones para el algoritmo MPA.

Realizado por: Armas Roger, 2021.

En la función se proporcionó los parámetros K, M y V los cuales representan a los recursos ortogonales, palabras de código y número de usuarios respectivamente. Para el desarrollo del algoritmo MPA se tomó en cuenta el proceso explicado por Altera y sociedades como parte de su concurso de simulación SCMA. Antes de iniciar el algoritmo, se obtiene el gráfico de factores correspondiente al sistema SCMA planteado.

```
% Calculo del gráfico de factores  
F = zeros(K, V); % Matriz correspondiente al gráfico de factores  
s = [K, M]; % Número de recursos ortogonales y palabra de código  
for k = 1:V  
    IND = find(CB(:, :, k)); % Se obtiene el indice donde se encuentra  
    un valor distinto de cero  
    [I, ~] = ind2sub(s, IND); % Conversión de indice lineal en  
    subindice  
    F(unique(I), k) = 1; % En el subindice establecido se asigna un 1  
end
```

El algoritmo MPA es un conjunto de pasos cuyo resultado es el valor de LLR (Log-Likelihood Ratio) correspondiente al acercamiento a cada constelación del libro de códigos. El proceso es iterativo lo cual se puede visualizar en el paso 2 del algoritmo. A continuación, se detallan los pasos del algoritmo implementado en Matlab.

- Paso 1: cálculos iniciales

En este paso se realizó los cálculos iniciales previos al proceso iterativo del paso dos. Se obtuvieron todas las señales residuales posibles dependiendo del canal conocido o estimado y las palabras de código. Para localizar las palabras de código de cada usuario independiente, mediante la función find se encontró los índices lineales de los elementos distintos de cero de la matriz de la gráfica de factores lo cual sirvió para ubicar la palabra de código del código de libros y es restado a la señal recibida del canal de comunicación. También se creó los vectores donde se almacenó los mensajes provenientes de los recursos ortogonales K hacia los usuarios V y viceversa.

```
% Paso 1: cálculos iniciales
f = zeros(M, M, M, K); % Matriz de señales residuales
for k = 1:K % Proceso iterativo dependiendo los recursos
    ind = find(F(k, :) == 1); % detección de los elementos distintos
de cero en la matriz del gráfico de factores
    for m1 = 1:M % Proceso iterativo dependiendo las palabras de
código
        for m2 = 1:M % Proceso iterativo dependiendo las palabras
de código
            for m3 = 1:M % Proceso iterativo dependiendo las
palabras de código
                d = y(k, jj) - (CB(k, m1, ind(1)) * h(k, ind(1), jj)
+CB(k, m2, ind(2)) * h(k, ind(2), jj) + CB(k, m3, ind(3)) * h(k, ind(1), jj)); %
Calculo de las señales residuales
                f(m1, m2, m3, k) = -Noise * sum(real(d)^2 + imag(d)^2); %
Sumatoria de los módulos de las señales residuales
            end
        end
    end
end

Ap = log(1/M); % Probabilidad previa de la palabra de código k,
asumiendo igual probabilidad
Igv = zeros(K, V, M); % Vector para el paso de mensajes desde el
recurso ortogonal hacia los usuarios
Ivg = Ap * ones(K, V, M); % Vector para el paso de mensajes desde
los usuarios hacia el recurso ortogonal
```

- Paso 2: Proceso iterativo

En este proceso se ven involucrados 3 usuarios denominados nodos VN v_1 , v_2 y v_3 , y un recurso ortogonal denominado nodo FN g . El traspaso del mensaje se da de la siguiente manera: el recurso ortogonal g envía el mensaje a v_1 cuyo contenido es la suposición de qué señal en g puede recibir todas las posibilidades de v_1 . Los usuarios v_1 y v_2 envían la probabilidad previa calculada en el paso 1 hacia el recurso g como se visualiza en la figura 10-2. De esta manera ocurre una actualización del vector Igv.

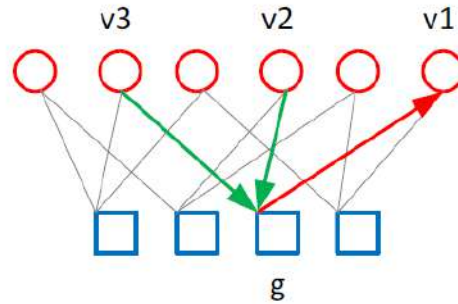


Figura 10-2: Traspaso de mensajes del nodo g al nodo v utilizando la gráfica de factores.

Realizado por: Armas, Roger, 2021.

El código de Matlab que se implementó para esta parte del proceso es el siguiente:

```
% Actualización del vector Igv
for k = 1:K % Proceso iterativo en base a los recursos
    ind = find(F(k, :)==1); % Detección de los elementos distintos de
    cero en la matriz del gráfico de factores
    for m1 = 1:M % Proceso iterativo dependiendo las palabras de
    código
        sIgv = zeros(1, M*M); % Vector de almacenamiento de las
        señales residuales
        for m2 = 1:M % Proceso iterativo dependiendo las palabras
        de código
            for m3 = 1:M % Proceso iterativo dependiendo las
            palabras de código
                sIgv((m2-1)*M+m3) =
                f(m1,m2,m3,k)+Ivg(k,ind(2),m2)+Ivg(k,ind(3),m3); % Traspaso del
                mensaje a sIgv
            end
        end
        Igv(k,ind(1),m1) = log_sum_exp(sIgv); % Suma
        logaritmica exponencial
    end

for m2 = 1:M % Proceso iterativo dependiendo las palabras de código
    sIgv = zeros(1, M*M); % Vector de almacenamiento de las señales
    residuales
    for m1 = 1:M % Proceso iterativo dependiendo las palabras de
    código
        for m3 = 1:M % Proceso iterativo dependiendo las palabras de
        código
            sIgv((m1-1)*M+m3) =
            f(m1,m2,m3,k)+Ivg(k,ind(1),m1)+Ivg(k,ind(3),m3); % Traspaso del
            mensaje a sIgv
        end
    end
    Igv(k,ind(2),m2) = log_sum_exp(sIgv); % Suma logaritmica
    exponencial
end

for m3 = 1:M % Proceso iterativo dependiendo las palabras de código
    sIgv = zeros(1, M*M); % Vector de almacenamiento de las señales
    residuales
```

```

        for m1 = 1:M % Proceso iterativo dependiendo las palabras de
código
            for m2 = 1:M % Proceso iterativo dependiendo las palabras
de código
                sIgv((m1-1)*M+m2) =
f(m1,m2,m3,k)+Ivg(k,ind(1),m1)+Ivg(k,ind(2),m2); % Traspaso del
mensaje a sIgv
            end
        end
        Igv(k,ind(3),m3) = log_sum_exp(sIgv); % Suma logaritmica
exponencial
    end
end

```

También se utilizó una función denominada `log_sum_exp.m` la cual realizó la sumatoria logaritmica exponencial de los mensajes traspasados durante el proceso.

```

function y = log_sum_exp(x)
xm = max(x); % Obtención del máximo valor del vector x
x = x-repmat(xm, size(x,1), 1); % Resta de cada elemento del vector x
con el valor máximo
y = xm + log(sum(exp(x))); % Suma del valor máximo de x con el
logaritmo de la suma de los elementos x como potencia de euler.

```

Finalmente, se realizó la actualización del vector `Ivg`, donde el nodo VN (usuario) v pasa las actualizaciones obtenidas de la información extrínseca a sus nodos FN (recursos ortogonales) vecinos como se puede observar en la figura 11-2.

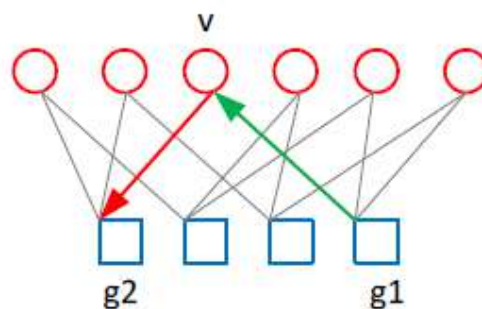


Figura 11-2: Envío de actualizaciones entre nodos g mediante el traspaso de mensajes con el nodo v .

Realizado por: Armas, Roger, 2021.

```

% Actualización del vector Ivg
for k = 1:V % Proceso iterativo para cada
    ind = find(F(:,k)==1); % Detección de los elementos distintos de
cero en la matriz del gráfico de factores
    s1 = log(sum(exp(Igv(ind(1),k,:)))); % Logaritmo de la suma de los
elementos del vector Igv como potencia de euler
    s2 = log(sum(exp(Igv(ind(2),k,:)))); % Logaritmo de la suma de los
elementos del vector Igv como potencia de euler
    for n = 1:M
        % Proceso análogo de la normalización en MPA, se puede
eliminar (s1 y s2), pero en SNR alto y / o número de iteraciones

```

pueden existir valores de LLR de NaN, por lo que se requiere Max-Log-MPA

```

    Igv(ind(1), k, n) = Igv(ind(2), k, n) - s2;
    Igv(ind(2), k, n) = Igv(ind(1), k, n) - s1;
end
end

```

- Paso 3: Calculo de LLR

Después de obtener las posibles opciones de palabra de código para cada usuario denominado suposiciones, se calculó la suposición general considerando la probabilidad previa calculada en el paso 1 y se suma con las suposiciones obtenida en Igv. Después de obtener la suposición de probabilidad de la palabra de código para cada usuario, se necesitó calcular la tasa de probabilidad de registro (LLR) de cada bit codificado mediante la siguiente ecuación.

$$LLR_x = \log\left(\frac{P(b_x = 0)}{P(b_x = 1)}\right)$$

$$LLR_x = \log\left(\frac{\sum_{m:b_{m,x}=0} Q_v(m)}{\sum_{m:b_{m,x}=1} Q_v(m)}\right) = \log\left(\sum_{m:b_{m,x}=0} Q_v(m)\right) - \log\left(\sum_{m:b_{m,x}=1} Q_v(m)\right)$$

Ecuación 1-2:

Ecuación para el cálculo de LLR.

Donde Q_v representa a la suposición de palabra de código para cada usuario.

```

Q = zeros(M, V); % Matriz de suposiciones para cada usuario
for k = 1:V
    ind = find(F(:, k) == 1); % Detección de los elementos distintos de
    % cero en la matriz del gráfico de factores
    for m = 1:M
        Q(m, k) = Ap + Igv(ind(1), k, m) + Igv(ind(2), k, m); % Suma de las
    % probabilidades
    end
end
LLR_tmp = zeros(log2(M) * V, 1);
for k = 1:V % Calculo de LLR
    LLR_tmp(2*k-1) =
    log((exp(Q(1, k)) + exp(Q(2, k))) / ((exp(Q(3, k)) + exp(Q(4, k)))));
    LLR_tmp(2*k) =
    log((exp(Q(1, k)) + exp(Q(3, k))) / ((exp(Q(2, k)) + exp(Q(4, k)))));
end
LLR(:, jj) = LLR_tmp; % Obtención de LLR para cada bit

```

Por último, se realizó la conversión de cada valor de LLR en bits correspondiente a la información enviada por los usuarios. Para este proceso se desarrolló la reconstrucción de una matriz cuando el valor de LLR es menor o igual a 0, cuyas dimensiones fueron configuradas en base a la extensión de los datos y la cantidad de bits continuos utilizados para los símbolos.

```

% Conversión de LLR a bit
datadec = reshape((LLR <= 0), [log2(CW) length(x)*U]).'; %
Decodificación de los datos
datar = zeros(log2(CW)*length(x), U); % Matriz de datos recuperados
for kk = 1:U
    datar(:,kk) = reshape(reshape(datadec, U, kk-1).', [], 1); %
Reconstruir el vector datadec y ponerlo en datar

```

2.2.6. Simulación OPDMA

Para la simulación de la técnica OPDMA se tomó en cuenta el modelo de sistema propuesto por Han (2016) el cual establece la división de la potencia en segmentos cuyas propiedades cumplen con el principio de ortogonalidad y equidad de bits. Esta técnica se enfoca en el cumplimiento de requerimientos de Calidad de Servicio por lo que se quieren tasas de transmisión objetivo con el objetivo de definir la potencia de transmisión adecuada. Los procesos de simulación para la etapa de transmisión y recepción de la técnica se muestran en el diagrama de bloques de la figura 12-2.

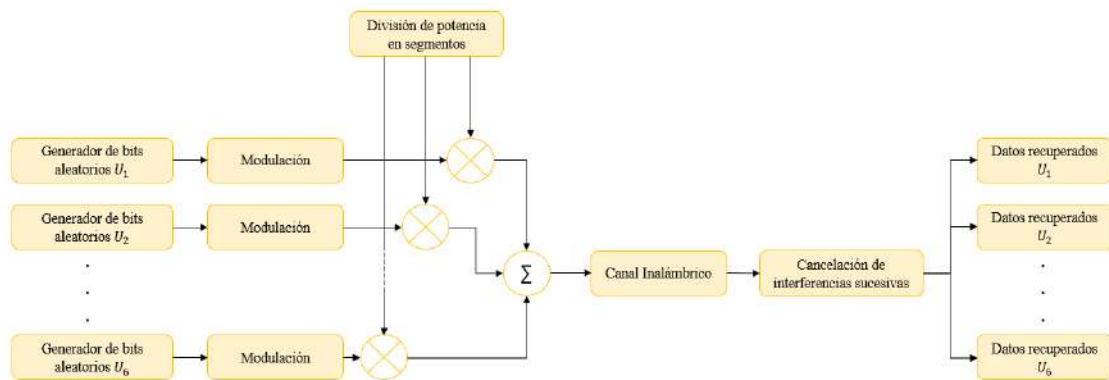


Figura 12-2. Diagrama de bloques del sistema OPDMA simulado.

Realizado por: Armas, Roger, 2021.

Los parámetros necesarios que se deben ingresar para realizar la simulación están relacionados con los requerimientos en cuanto a tasas de transmisión para cada usuario. Para el desarrollo de la técnica se tomó en cuenta las siguientes variables de entrada:

Tabla 5-2: Parámetros de entrada requeridos al inicio de la simulación de OPDMA.

Parámetro inicial	Descripción
M	Número de usuarios a simular (Máximo 6 usuarios).
Nbits	Cantidad de bits por usuario.
Mod	Tipo de modulación (BPSK, QPSK, 16-QAM, 64-QAM)
fbeta	Factor de aumento de potencia para cada usuario.
S_max	Valor máximo de potencia en dBm.
B	Ancho de banda del canal en MHz.

c	Tasas de transmisión para cada usuario en Mbps.
h	Ganancia de canal

Realizado por: Armas, Roger, 2021.

2.2.6.1. Transmisor

Los códigos de MATLAB correspondientes a cada proceso en la etapa de transmisión se detallan a continuación.

- Generador de bits aleatorios

De manera similar a las técnicas ortogonales, se generó un vector de bits \mathbf{x} aleatoriamente con la función `randi` de MATLAB permitiendo obtener valores enteros entre 0 y 1.

```
% Generador de N bits de datos para M usuarios
M = 3; % Numero de usuarios
Nbits = 10000; % Cantidad de bits
x = randi([0 1], Nbits, M); % Vector de bits aleatorios
```

- Modulación

El modelo propuesto por Han (2016) establece que los bits generados por cada usuario deben ser modulados mediante QAM, sin embargo, se realizó el simulador para modulaciones BPSK y QPSK con el fin de ampliar el rango de análisis respecto a los esquemas de modulación utilizados en las técnicas OMA. Para las modulaciones BPSK y QPSK se empleó objetos de la caja de herramientas de comunicaciones en MATLAB, `comm.BPSKmodulator` y `comm.QPSKmodulator` donde el tipo de entrada se configuró como bit. Por otro lado, la modulación 4-QAM se realizó mediante la función `qammod` donde se especifica el tipo de entrada como bit y el nivel de modulación $M = 4$.

```
switch Mod
    case 1
        for u = 1:M
            BPSKmod = comm.BPSKModulator('BitInput',true); % Objeto para
modulador BPSK
            Smod(:,u) = BPSKmod(x(:,u)); % Aplicación de la modulación
        end
    case 2
        for u = 1:M
            QPSKmod = comm.QPSKModulator('BitInput',true); % Objeto para
modulador QPSK
            Smod(:,u) = QPSKmod(x(:,u));
        end
    case 3
        for u = 1:M
            Smod(:,u) = qammod(x(:,u),4,'InputType','bit'); % Aplicación
de la modulación
```

```

end
end

```

- División de potencia en segmentos

Para el cálculo de los segmentos de potencia se empleó la estrategia de división y asignación de potencias óptima para la técnica OPDMA la cual es propuesta por Han (2016). El algoritmo de cálculo se basa en 2 pasos esenciales. En el primer paso se utilizó la función sort de MATLAB para ordenar las ganancias de canal descendientemente y se ingresó los valores del factor β correspondiente a cada usuario. Luego, en el paso dos se ingresó la potencia máxima en dBm, el ancho de banda del canal en **B** MHz y las tasas de transmisión **c** de cada usuario. Posteriormente se realizó en un ciclo for el cálculo de los segmentos de potencia y un control sobre la misma debido a que la suma de los segmentos de potencia no debe superar el valor de potencia máxima establecida.

```

% Paso 1 - Clasificar las ganancias de canal en orden descendente
h = sort(h, 'descend'); % Ordenar de mayor a menor
h_m=h.*fbeta;
% Paso 2
S_max = 46;% Potencia máxima en dBm
s_max = (10^-3)*db2pow(S_max); % Potencia máxima lineal
sum_p = 1 ; % Suma de cada segmento de potencia para control
% Parámetros de transmisión de usuarios
N = - 174 + 10*log10(B); %Potencia de ruido en escala logaritmica
n = (10^-3)*db2pow(N); %Potencia de ruido en escala lineal
% Calculo de los segmentos de potencia PS
s_i = 0;
for m =1:1:M
    s(m) = ((2.^(c(m)/B))-1)*(n/h_m(m))*sum_p); % Calculo del
segmento de potencia
    sum_p = s_i + s(m); % Resta del PS al total de potencia para
control
    s_i = sum_p;
    if sum_p > s_max % Control de los segmentos de potencia
        s(:) = 0;
        break
    end
end
end

```

- Asignación de potencia

La asignación de potencia se lleva a cabo en una multiplicación de las señales moduladas con la raíz cuadrada de cada segmento de potencia calculado en el paso anterior.

```

% Asignación de potencia para cada usuario y suma de los mismos
for m = 1:1:M
    y(:,m) = sqrt(s(m)).*Smod(:,m); % Multiplicación de la señal con
el segmento de potencia
end

```

- Superposición de códigos

El principio de superposición de códigos establece la transmisión de los datos de diferentes usuarios al mismo tiempo. Para aplicar este principio en la técnica OPDMA se realizó una suma de las señales almacenadas en un vector mediante la función sum donde se especifica que se realiza la suma de las columnas del vector con el valor de argumento 2.

```
y_f = sum(y,2); % Suma de señales / Superposición de código
```

2.2.6.2. Receptor

En la etapa de recepción solo un proceso realiza el trabajo de demultiplexar y demodular las señales de cada usuario y es la cancelación de interferencias sucesivas. Para la aplicación de esta técnica, los usuarios se ordenaron acorde al nivel de potencia asignado, de mayor a menor potencia. El código de este proceso se detalla a continuación.

- Cancelación de Interferencias Sucesivas SIC

La técnica SIC se caracteriza por decodificar las señales de forma sucesiva en base al nivel de potencia asignado a los usuarios. Los primeros datos recuperados corresponden al usuario con mayor segmento de potencia asignada y se repite esta lógica en orden descendente. La demodulación de los datos se realizó mediante los objetos comm.BPSKdemodulator y comm.QPSKdemodulator para las modulaciones PSK. Por otro lado, la demodulación QAM se empleó con la utilización de las funciones qammod y qamdmod que proporciona MATLAB. Por medio de una estructura switch se aplicó la técnica SIC acorde a la modulación empleada. Con el objetivo de explicar el código se tomó en cuenta la cantidad de 3 usuarios debido a que los pasos a seguir son repetitivos para cualquier número de usuarios asignado.

```
for j = 1:1:M
    if j == 1
        dec(:,j) = step(BPSKdemod, s_canal); %
Demodulacion directa para obtener datos de U1
    elseif j == 2
        dec12 = step(BPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
        dec12_remod = step(BPSKmod, dec12);
%Remodulacion de los datos de U1
        rem2 = s_canal - sqrt(s(1))*dec12_remod; %SIC
para remover los datos de U1
        dec(:,j) = step(BPSKdemod, rem2);
%Demodulación directa para obtener los datos de U2
    elseif j == 3
        dec13 = step(BPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
        dec13_remod = step(BPSKmod, dec13);
%Remodulacion de los datos de U1
```

```

        rem31 = s_canal - sqrt(s(1))*dec13_remod;    %SIC
para remover los datos de U1
        dec23 = step(BPSKdemod, rem31);           %Demodulación
directa para obtener los datos de U2
        dec23_remod = step(BPSKmod, dec23);
%Remodulacion de los datos de U2
        rem32 = rem31 - sqrt(s(2))*dec23_remod; %SIC para
remover los datos de U2
        dec(:,j) = step(BPSKdemod, rem32);
%Demodulación directa para obtener los datos de U3
end

```

Para obtener los datos del usuario con mayor potencia asignada basta con demodular directamente la señal recibida y afectada por el canal inalámbrico **s_canal**, posteriormente el receptor realiza nuevamente a la modulación a los datos obtenidos inicialmente con el fin de ser utilizado para la resta de esta con la señal proveniente del canal. Al realizar esta sustracción de la señal del usuario 1, la señal resultante **rem** contiene los datos del segundo usuario con mayor potencia y se procedió a demodular directamente para obtener los bits de información. Este procedimiento se repite de manera exponencial debido a que, al aumentar el número de usuarios, también aumenta las líneas de código correspondiente al proceso de cancelación de interferencias sucesivas.

2.2.7. Canales inalámbricos

Para la simulación de un canal AWGN se utilizó la función `awgn` que proporciona MATLAB donde los argumentos de entrada son la señal proveniente del transmisor, el SNR en dimensiones logarítmicas o lineales, en caso de ser lineal, se debe especificar el nivel de potencia de la señal.

```

s_canal = awgn(s_tx, snr, p_tx, 'linear'); % SNR en escala lineal
s_canal = awgn(s_tx, SNR, 'measured'); % SNR en escala logarítmica

```

Por otro lado, la simulación del canal Rice se realizó mediante el cálculo de los coeficientes de Rician los cuales representan la ganancia del canal. Se tomó en cuenta el factor K de la distribución de Rice y se calculó la media y varianza utilizando la misma. Además, se añadió ruido mediante la utilización de la función `random` para generar valores aleatorios con distribución normal. Para el canal Rayleigh la metodología de la simulación es similar, a diferencia que no cuenta con el factor K . La señal de salida del canal es la multiplicación de los datos modulados del transmisor multiplicado con la ganancia del canal h y se suma el ruido blanco gaussiano.

```

% Canal Rice
mean=sqrt(k/(k+1)); % Media obtenida con el factor K ingresado
sigma=sqrt(1/(2*(k+1))); % Varianza
Nr=randn(1,length(s_tx))*sigma+mean; %Genera valores aleatorios para
ricean
Ni=randn(1,length(s_tx))*sigma;
h=sqrt(Nr.^2+Ni.^2); % Coeficientes de desvanecimiento Rician
Np=1/snr; % Potencia de ruido

```



```

sd=sqrt(Np/2);% Desviación estandar para ruido gaussiano.
n=random('Normal',0,sd,1,length(s_tx))+1i*random('Normal',0,sd,1,lengt
h(s_tx)); %Genera ruido gaussiano
sigcanal = s_tx.*h' + n'; % Otra opción puede ser con la función awgn
sigcanal = awgn(s_tx, snr, 'measured');
% Canal Rayleigh
Nc=random('normal',0,1/(sqrt(2)),1,length(s_tx))+1i*random('normal',0,
1/(sqrt(2)),1,length(s_tx)); % Ruido complejo
Nr=real(Nc); % Parte real del Ruido
Ni=imag(Nc); % Parte imaginaria del ruido
h=sqrt(Nr.^2+Ni.^2); % rayleigh fading coefficient
Np=1/snr; % Potencia de ruido
sd=sqrt(Np/2);% Desviación estandar para ruido gaussiano.
n=random('Normal',0,sd,1,length(s_tx))+1i*random('Normal',0,sd,1,lengt
h(s_tx)); %Genera ruido gaussiano
sigcanal = s_tx.*h' + n'; % Otra opción puede ser con la función awgn
sigcanal = awgn(s_tx, snr, 'measured');

```

CAPITULO III

3. MARCO DE RESULTADOS Y DISCUSIÓN DE LOS RESULTADOS

En este capítulo se presentan los resultados de las simulaciones de las técnicas de acceso múltiple ortogonales y no ortogonales planteadas, mediante la representación gráfica del BER en función de la relación señal a ruido SNR. Con el objetivo de validar el trabajo realizado, se realiza una comparación con las gráficas de las curvas BER de estudios y pruebas realizadas en investigaciones publicadas. Además, se presenta un análisis individual del rendimiento de las técnicas en base al canal de comunicación simulado.

3.1. Validación de las curvas BER

3.1.1. Comparación de resultados CDMA

Para el análisis comparativo de los resultados obtenidos se consideró las curvas de BER proporcionadas en el documento “Software Defined Radio implementation of DS-CDMA in inter-satellite communications for small satellites” donde se prueba el rendimiento de BER en función del E_b/N_0 para 1, 2, 5 y 7 usuarios con modulaciones BPSK y QPSK en un canal AWGN y Rayleigh, con potencia de transmisión de 2 Watts (Pinto, et al., 2015, p. 5). A continuación, en el gráfico 1-3 se observa las curvas correspondientes a los parámetros establecidos en el documento mencionado.

Con respecto al canal Rice, en el documento “Performance analysis of CDMA under Rayleigh, Rician and Nakagami fading channels” desarrollado por Machado Mahajan y D. B. Bhoyarse se obtienen curvas BER en función del SNR para una modulación BPSK en un canal con desvanecimiento y valor de factor K igual a 1 (Mahajan y Bhoyar, 2011, p. 318), como se observa en el gráfico 2-3.

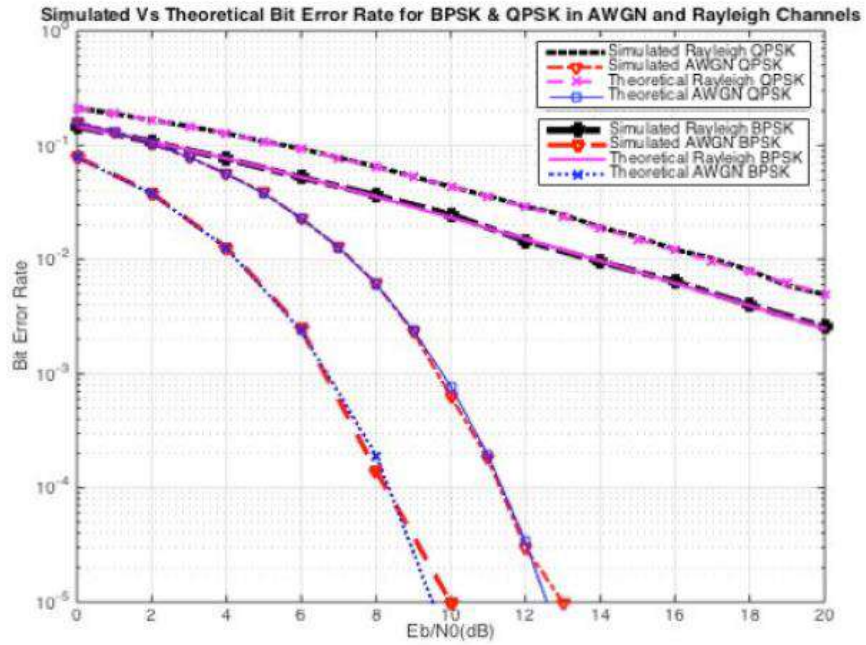


Gráfico 1-3: BER vs E_b/N_0 para DS-CDMA con modulaciones BPSK y QPSK en un canal AWGN y Rayleigh.

Fuente: Pinto, Afghah, Radhakrishnan, & Edmonson. 2015, p. 6.

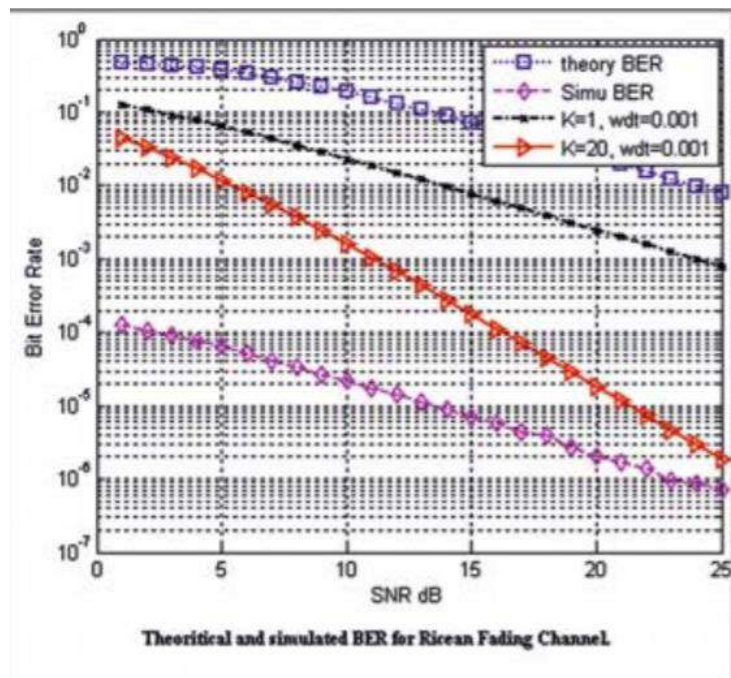


Gráfico 2-3: BER vs SNR para CDMA con modulación BPSK y canal Rician.

Fuente: Mahajan, & Bhoyarse, 2011, p. 319.

En el gráfico 3-3 se presenta los resultados obtenidos del simulador de CDMA con los esquemas de modulación BPSK y QPSK sobre los canales AWGN y Rayleigh, con una potencia de señal de 2 Watts. Cabe recalcar que la disposición de las curvas para cada escenario de las simulaciones realizadas para esta técnica está adaptada al del gráfico 1-3 con el fin de obtener visualmente una comparación más rápida.

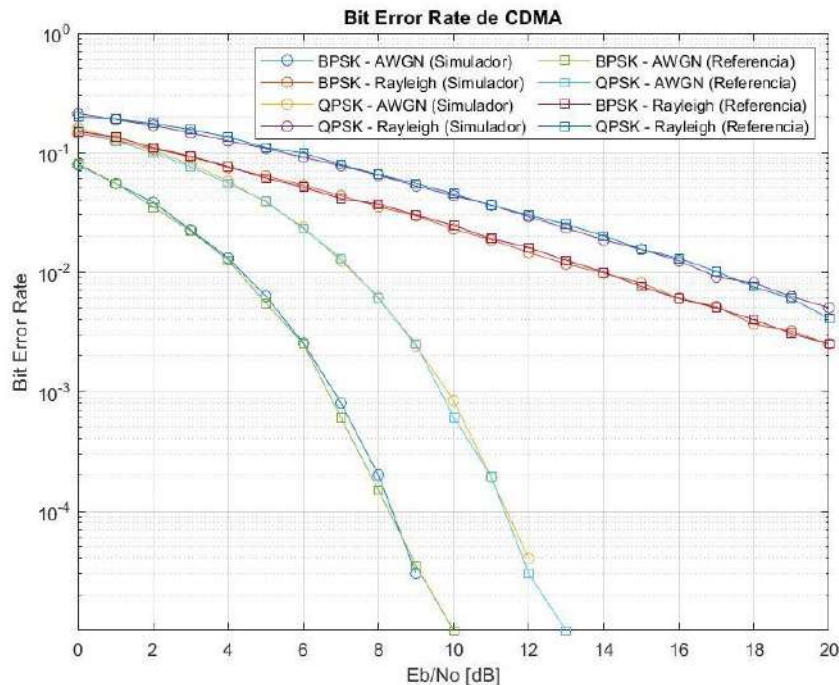


Gráfico 3-3: Comparación del BER en función del SNR de CDMA con modulaciones BPSK y QPSK en canales AWGN y Rayleigh.

Realizado por: Armas, Roger, 2022.

En la curva BER de CDMA para el canal AWGN se observó que se logra una gran similitud en las curvas obtenidas en el simulador con las curvas de referencia, las modulaciones BPSK y QPSK tienen una diferencia notable de aproximadamente 3 dB de E_b/N_0 , la cual se presencié tanto en los resultados obtenidos como en las curvas de referencia. Además, se pudo notar que las curvas obtenidas llegan hasta una probabilidad de 3×10^{-5} para la modulación BPSK y 4×10^{-5} para QPSK. Contrario a esto, las curvas de referencia avanzan hasta una probabilidad de 1×10^{-5} en ambas modulaciones. Por otro lado, para el canal Rayleigh el comportamiento de las curvas de referencia resultó similar a lo obtenido por el simulador, mismo que presenta pequeñas fluctuaciones en la tasa de errores. Adicionalmente, entre las modulaciones planteadas se presentó una diferencia de aproximadamente 4 dB lo cual se reflejó en todas las curvas presentadas para este último canal.

En cuanto al canal Rice, para validar los resultados obtenidos que se muestran en la curva BER del gráfico 4-3, se cambió el parámetro de evaluación a SNR con el fin de realizar un análisis más directo con lo presentado en la curva de referencia del gráfico 2-3 y evaluar los resultados en base a el mismo.

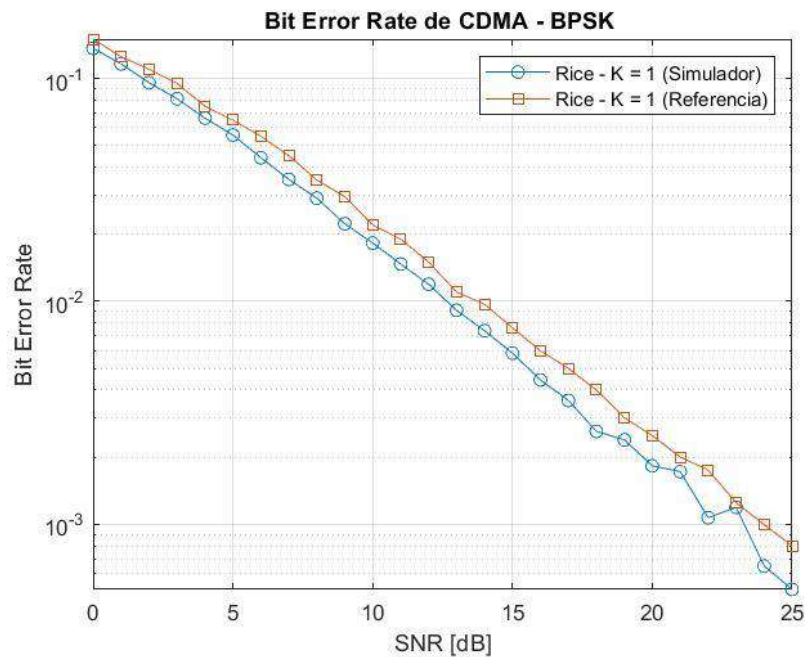


Gráfico 4-3: Comparación del BER en función del SNR de CDMA con modulación BPSK sobre canal Rice con factor K = 1.

Realizado por: Armas, Roger, 2022.

En general, la tasa de errores de bits para un canal Rice aumentó siendo similar al canal Rayleigh con la diferencia que la pendiente de caída de la tasa resultó más rápida conforme aumenta el SNR. Para una tasa de errores de 10^{-1} con modulación BPSK se requirió de una SNR de 1,9 dB concordando de manera similar con lo obtenido en la curva de referencia donde se logra el mismo nivel de BER con 2,9 dB de SNR aproximadamente evidenciando una diferencia de 1 dB con los resultados del simulador. Del mismo modo, para una tasa de error de 10^{-3} el sistema requirió de 23,3 dB de SNR, difiriendo en pequeña medida con lo mostrado en los resultados obtenidos por Mahajan y Bhoyar donde se pudo observar que, para la tasa de error planteada se precisó de 24 dB en cuanto a la SNR habiendo una diferencia de 0,7 dB para el nivel de BER planteado entre lo obtenido con la curva de referencia.

3.1.2. Comparación de resultados OFDMA

Para la validación de esta técnica se consideró los resultados obtenidos de simulaciones con parámetros de LTE debido a la aplicabilidad de OFDMA en el enlace descendente de esta tecnología de comunicaciones móviles los cuales se pueden observar en el documento “SC-FDMA & OFDMA in LTE physical layer”. A continuación, en las curvas correspondientes del gráfico 5-3 se observa las curvas BER en función del SNR para un canal AWGN y modulaciones BPSK, QPSK, 16-QAM y 64-QAM.

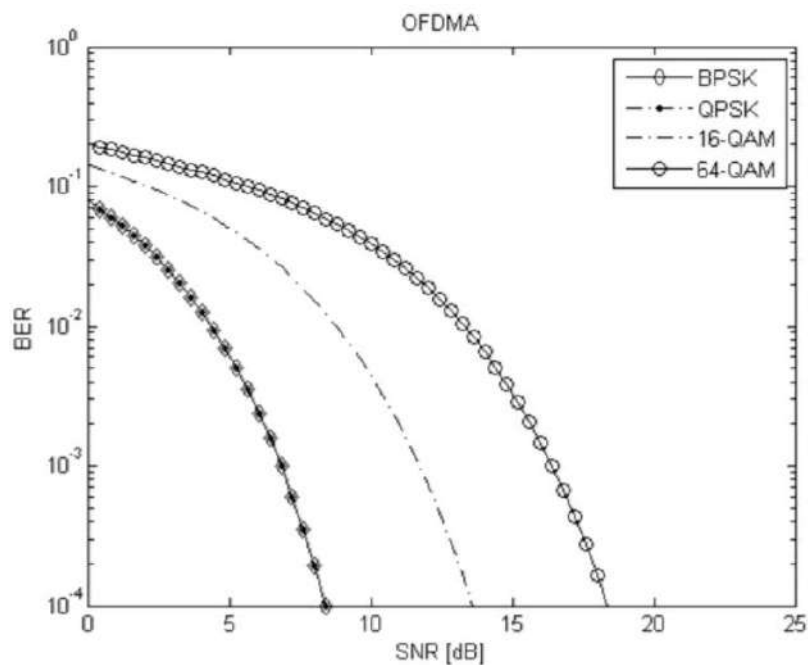


Gráfico 5-3: BER vs SNR de OFDMA para BPSK, QPSK, 16-QAM y 64-QAM sobre canal AWGN.

Fuente: Nsaif Sukar & Pal, 2014, p. 82.

En los gráficos 6-3 y 7-3 se muestran los resultados obtenidos en “BER Performance of OFDM System over Nakagami-m Fading Channels with Different Modulation Schemes”, este documento proporciona curvas BER para modulaciones BPSK y QPSK para canales de desvanecimiento Rayleigh y Rice, las mismas que sirvieron de referencia para analizar y comparar con los resultados que arroja el simulador desarrollado en este trabajo.

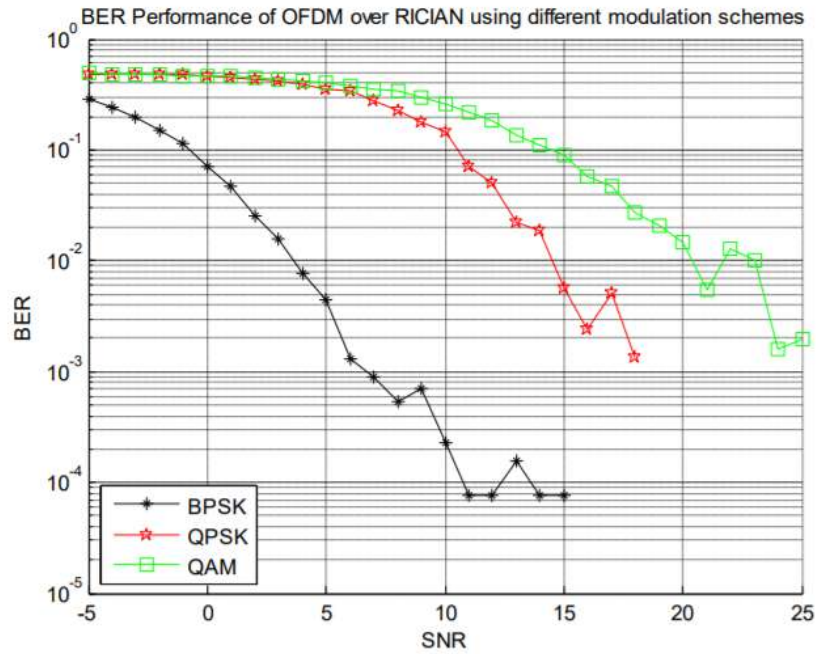


Gráfico 7-3: BER vs SNR de OFDM para modulación BPSK y QPSK sobre canal Rice.

Fuente: Jain & Dubey, 2012, p. 115.

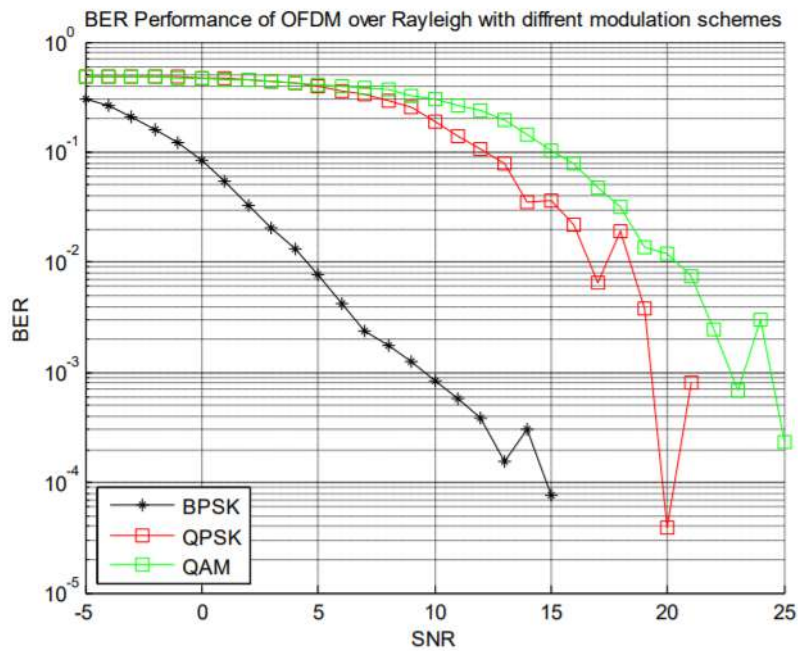


Gráfico 8-3. BER vs SNR de OFDM para modulación BPSK y QPSK sobre canal Rayleigh.

Fuente: Jain & Dubey, 2012, p. 115.

En el gráfico 8-3 se presentan los resultados obtenidos con el simulador de OFDMA con los esquemas de modulación BPSK, QPSK, 16-QAM y 64-QAM para un canal AWGN y parámetros similares a los presentados en el documento de referencia.

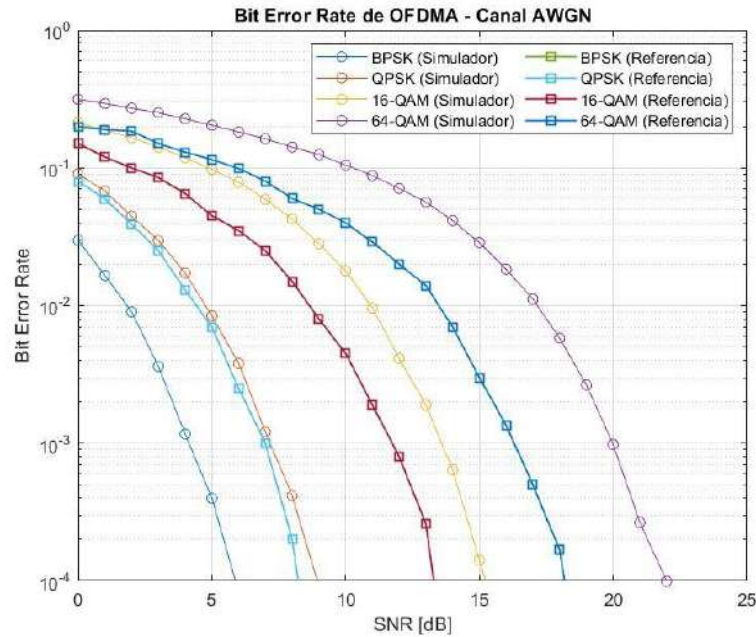


Gráfico 8-3: Comparación del BER en función del SNR de OFDMA para canal AWGN con distintas modulaciones.

Realizado por: Armas, Roger, 2022.

La disposición de las curvas para cada escenario simulado por esta técnica se encuentra adaptada para realizar un análisis comparativo más directo respecto al gráfico 7-3. En la curva BER de OFDMA para un canal AWGN se evidenció que existe una similitud significativa en las curvas correspondiente a la modulación QPSK siendo el BER obtenido por el simulador el que supera levemente a la curva de referencia. No obstante, en la modulación BPSK se presenció una diferencia de SNR de 3 dB respecto al BER del documento. Este comportamiento continúa con las modulaciones restantes donde se registró una diferencia entre la curva obtenida por el simulador y la de referencia que oscila en el intervalo de 2 a 3 dB para 16-QAM y de 3 a 4 dB para 64 QAM. Estos desfases en las curvas se deben a la falta de información de los parámetros de transmisión en el documento de referencia donde no se especifica la potencia de la señal a transmitirse por el canal. En los resultados del simulador se calculó automáticamente el valor de la potencia en base al esquema de modulación mediante la propiedad ‘measured’ de la función awgn de Matlab.

Las curvas BER que se observan en el gráfico 9-3 representan los resultados obtenidos del simulador de OFDMA para un canal Rice con modulaciones BPSK y QPSK además de las curvas

de referencia. Respecto a la modulación BPSK, se evidenció que existe una diferencia significativa en las curvas a partir de 2 dB correspondiente al SNR. Para un BER de 10^{-3} , se visualizó el mayor rango de diferencia entre la curva del simulador y la de referencia con un SNR de 4,2 dB. Por otro lado, para la modulación QPSK las curvas obtenidas difirieron en gran medida a los presentados en el documento de referencia donde se observó un BER que desciende en un valor de SNR más alto a comparación del obtenido por el simulador que tiene una diferencia de 7,5 dB respecto a la curva de referencia. Si bien es cierto, los parámetros de simulación de las curvas de referencia no se especifican por completo, lo cual ocasiona que las curvas tengan una gran disimilitud, sin embargo, cabe recalcar que la configuración de los canales se mantiene igual que en CDMA.

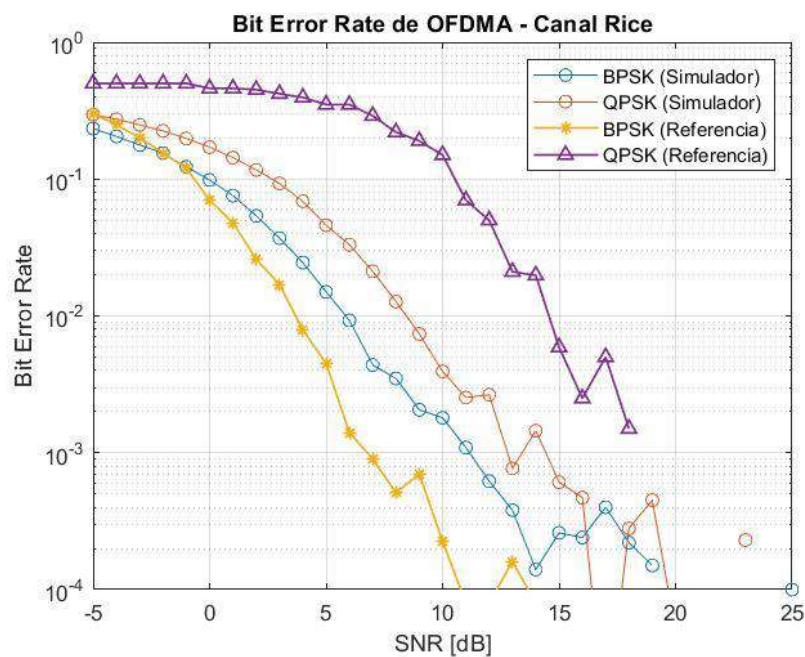


Gráfico 9-3: Comparación del BER en función del SNR de OFDMA con modulaciones BPSK y QPSK sobre canal Rice.

Realizado por: Armas, Roger, 2022.

En el gráfico 10-3 se presentan los resultados correspondientes a las modulaciones BPSK y QPSK en OFDMA sobre un canal Rayleigh obtenidos del simulador desarrollado en este trabajo y las pertenecientes al documento de referencia para esta comparación. De manera similar a lo evidenciado en el canal Rice, las curvas distan entre sí en gran medida entre éstas. Para la modulación BPSK esta diferencia resultó más notoria que en QPSK cuyo escenario difirió con un máximo de 3 dB con la curva de referencia. También se observó que para un valor de SNR superior a 16 dB existen grandes fluctuaciones en el BER expuesto en el documento lo cual no se presenta en lo obtenido por el simulador. Por otro lado, los resultados obtenidos con modulación BPSK difirió con 13 dB aproximadamente respecto a la curva de referencia, lo cual se produjo

por la falta de características del canal de desvanecimiento en los resultados presentados por el artículo de referencia. Cabe recalcar que las gráficas tomadas como base para la comparación con lo obtenido por el simulador corresponden a una técnica OFDM.

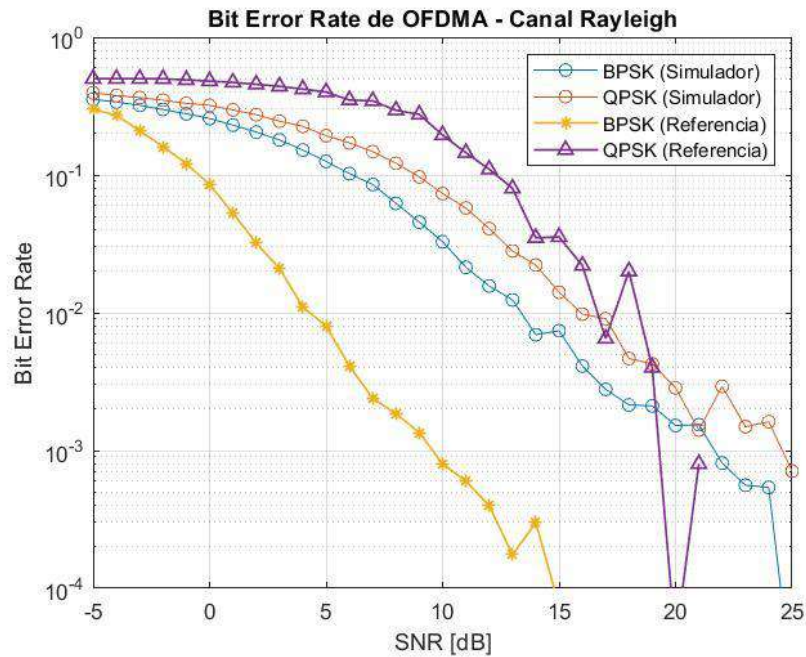


Gráfico 10-3: Comparación del BER en función del SNR de OFDMA con modulsiones BPSK y QPSK sobre canal Rayleigh.

Realizado por: Armas, Roger, 2022.

3.1.3. Comparación de resultados SCMA

Para el análisis y validación de los resultados obtenidos con SCMA, se consideró las curvas BER planteadas en el documento “On Fixed-point Implementation of Log-MPA for SCMA Signals” desarrollado por Jiaqi Liu, Gang Wu, Shaoqian Li, y Olav Tirkkonen, donde se empleó la técnica con el algoritmo de paso de mensajes logarítmica, la cual se implementó en el simulador. Adicionalmente se consideró un sistema con $K = 4$, $N = 2$, $J = 6$, $M = 4$ y factor de sobrecarga de 1,5 o 150%. K corresponde al número de recursos ortogonales utilizados en cada palabra de código, N se utilizó para referirse al número de bits empleados en la codificación, J son los números de usuarios y M se refiere al número de constelaciones para cada palabra de código. Es importante señalar que no se especifica el libro de códigos utilizados o el tipo de libro. En el gráfico 11-3 se muestra las curvas BER para un canal AWGN.

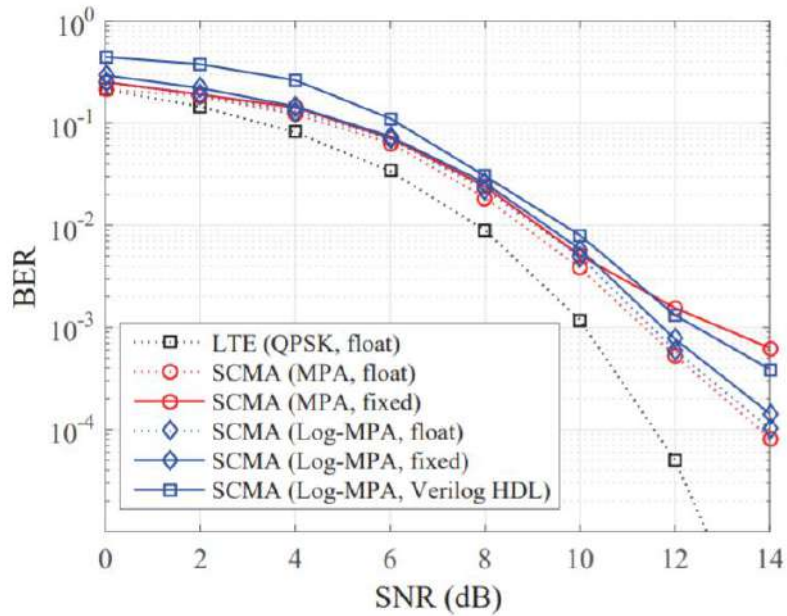


Gráfico 11-3: BER vs SNR de SCMA para un canal AWGN.

Fuente: Liu, et al., 2016, p. 327.

Por otro lado, el análisis en el canal Rayleigh se realiza con las curvas BER obtenidas en “Comparison Study of Non-Orthogonal Multiple Access Schemes for 5G” desarrollado por Bichai Wang, Kun Wang, Zhaohua Lu, Tian Xie y Jinguo Quan, las cuales fueron tomados como referencia para la validación del trabajo realizado por el simulador. En el gráfico 12-3 se puede visualizar los resultados obtenidos por este documento, el cual se realizó bajo parámetros del sistema similares a excepción del libro de códigos.

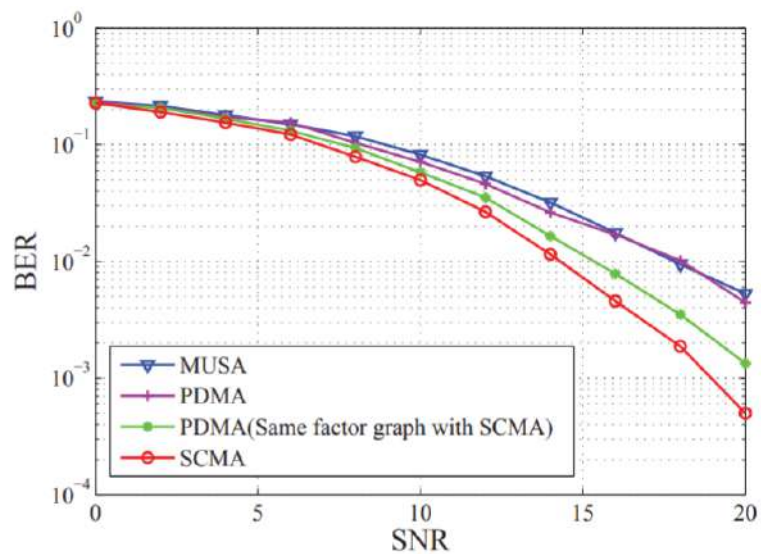


Gráfico 12-3: BER vs SNR de SCMA para canal Rayleigh.

Fuente: Wang, et al., 2015, p. 4.

Las curvas visualizadas en el gráfico 13-3 representan el BER obtenido por el simulador de SCMA para el canal AWGN junto con el BER proporcionado por el gráfico 11-3. Se observó que existe una diferencia significativa para valores de SNR superiores a 4 dB. Para un nivel BER de 10^{-1} se requirió un SNR de 5,15 dB según lo estipula la curva de referencia, mientras que el resultado del simulador requiere un valor de 7,2 dB, obteniéndose una diferencia de 2,05 dB entre las curvas. También se evidenció que la disimilitud aumenta proporcionalmente al valor de SNR ya que para un BER de 10^{-3} se requirió un valor SNR de 11,8 dB en la curva de referencia y en la curva obtenida por el simulador, se necesitó un valor de 15,3 dB constatando una diferencia de 3,5 dB entre las curvas. La variación de los resultados se debe a la utilización de libros de códigos distintos a los que se utilizaron en el artículo que provee la curva tomada como base de la comparación.

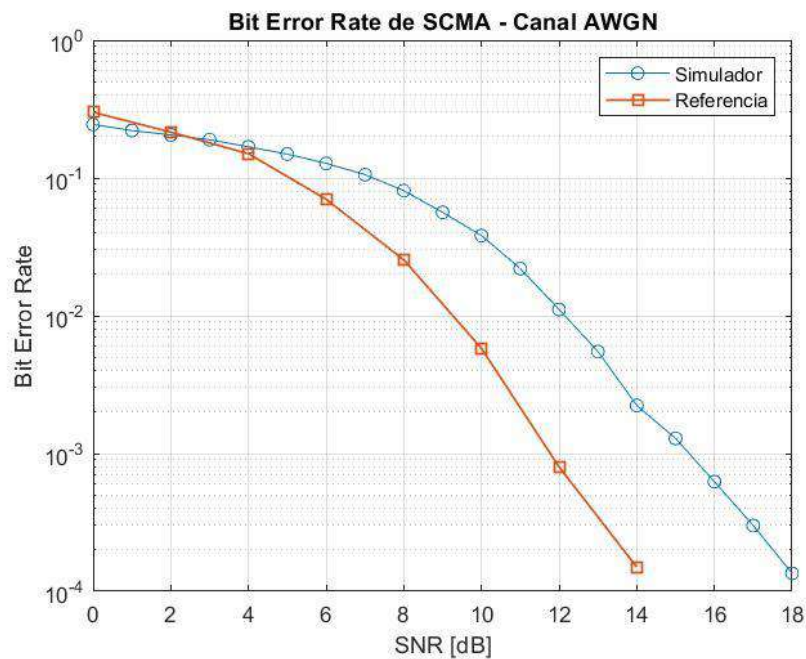


Gráfico 13-3: Comparación del BER en función del SNR de SCMA para canal AWGN.

Realizado por: Armas, Roger, 2022.

En el gráfico 14-3 se puede observar la curva de BER obtenida por el simulador SCMA para un canal de desvanecimiento Rayleigh y la curva expuesta en el artículo de referencia para la comparación de los resultados. El comportamiento de las curvas en el canal planteado resultó similar siendo el BER obtenido por el simulador un poco elevado respecto al otro. Para un nivel BER de 10^{-1} , se requirió un valor de SNR de 7 dB proporcionado por la curva base, mientras que el resultado del simulador evidencia un valor de 9,9 dB de SNR para alcanzar el mismo BER, esto expone una diferencia de 2,9 dB entre las curvas. Este rendimiento se vio marcado para todos

los niveles de SNR. De la misma manera que en el canal AWGN, estas disimilitudes se produjeron debido a la falta de especificación tanto el libro de códigos utilizados, como el tipo de libro.

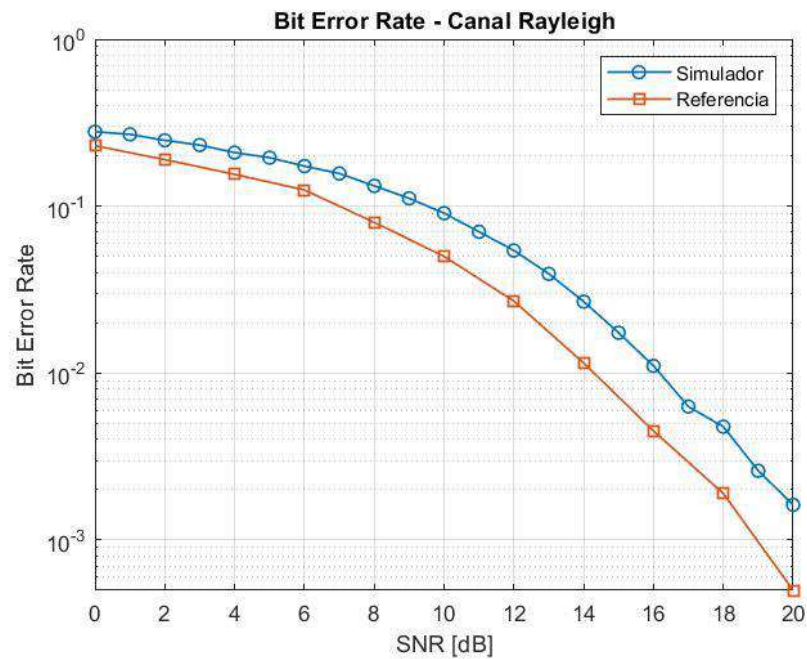


Gráfico 14-3: Comparación del BER en función del SNR de SCMA para canal Rayleigh.

Realizado por: Armas, Roger, 2022.

3.1.4. Comparación de resultados OPDMA

Para la validación de las curvas obtenidas por el simulador se tomó como referencia los resultados expuestos en el artículo “Study of the Symbol Error Rate/Bit Error Rate in Non-orthogonal Multiple Access (NOMA) Systems” desarrollado por Weijia Han, Yu Wang, Di Tang, Ronghui Hou, y Xiao Ma, mismos que propusieron esta técnica de Acceso Múltiple No-ortogonal. En este documento se realizó el análisis de BER para una transmisión de NOMA en el dominio de la potencia sobre un canal de pruebas AWGN empleando modulación QPSK y un número de 2 usuarios. Adicionalmente se mostró un gráfico lineal de las curvas BER en función del SNR mismo que fue reescrito en Matlab y se puede visualizar en el gráfico 15-3.

La modulación 4-QAM se tomó en cuenta en nuevas pruebas realizadas para NOMA las cuales se encuentran en el artículo “Study of SER and BER in NOMA Systems” desarrollado por Weijia Han, Xiao Ma, Di Tang, y Nan Zhao donde se realizó el análisis de BER cuyas ecuaciones planteadas pueden ser aplicadas para el análisis de la tasa de errores de bits de una transmisión basado en OPDMA para un enlace descendente (Han, et al., 2021, p. 3332). Las gráficas de BER fueron

reescritas en Matlab y se visualizan en los gráficos 16-3 y 17-3, donde se puede visualizar las curvas para 2 y 5 usuarios con un número de segmentos de potencia de 2 y L respectivamente.

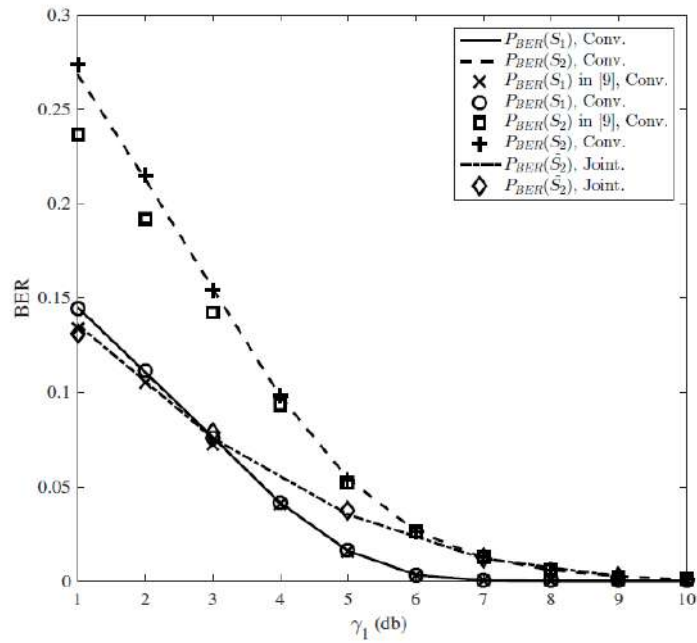


Gráfico 15-3: BER vs SNR de OPDMA con modulación QPSK en canal AWGN.

Fuente: Han, Wang, Tang, Hou, & Ma, 2018, p. 106.

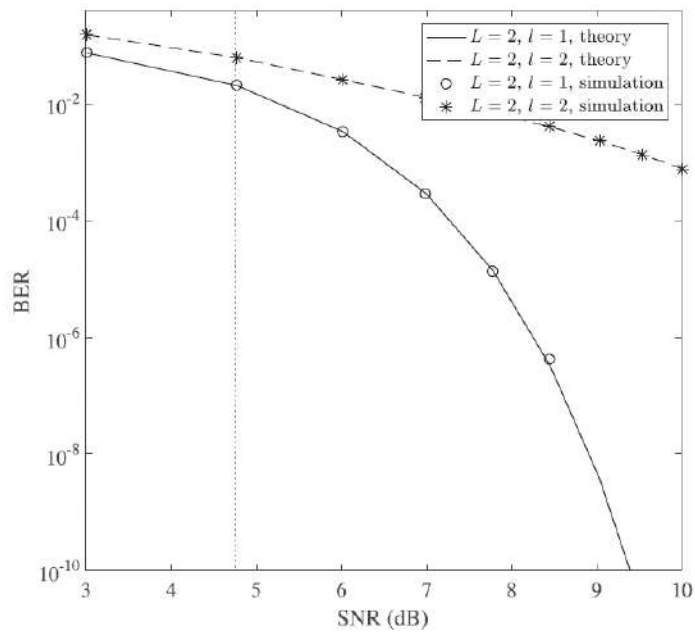


Gráfico 16-3: BER vs SNR de OPDMA con modulación 4-QAM en canal AWGN para 2 usuarios y 2 segmentos de potencia.

Fuente: Han, Ma, Tang, & Zhao, 2021, p. 3334.

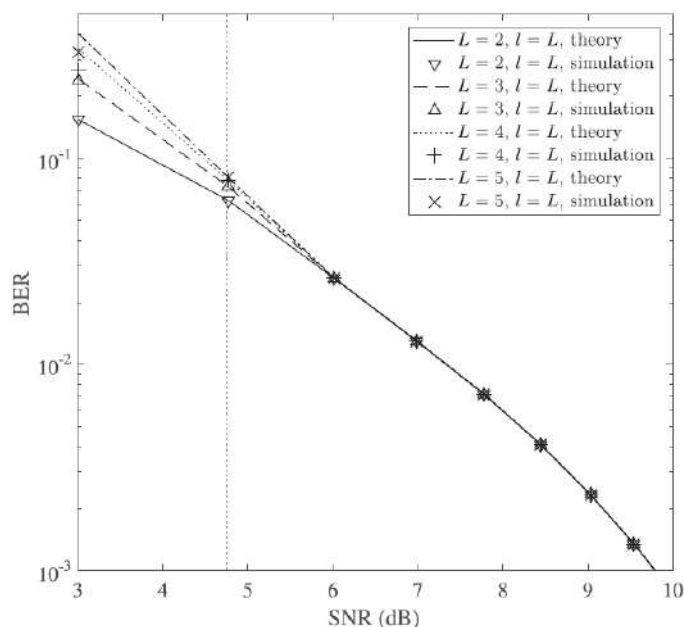


Gráfico 17-3: BER vs SNR de OPDMA con modulación 4-QAM en canal AWGN para 5 usuarios y L segmentos de potencia.

Fuente: Han, Ma, Tang, & Zhao, 2021, p. 3334.

El algoritmo planteado en la tabla 1-1 por el autor de esta técnica propone una ecuación en el paso 2 para el cálculo de los segmentos de potencia de cada usuario, sin embargo, este algoritmo no es eficiente en cuanto a las tasas de transmisión requeridas, por lo que se planteó una modificación a esta ecuación con el fin de lograr un mejor resultado.

$$s = \left(2^{\frac{c}{B}} - 1\right) \left(\frac{\sigma^2}{h_m} + |s|_1\right)$$

Ecuación 1-3:

Segmentos de potencia, algoritmo original.

3.1.4.1. Segmentos de potencia con el algoritmo original

Para el cálculo de los segmentos de potencia se plantearon los parámetros tales como la potencia de transmisión 46 dBm (39,81 W), ancho de banda 10 MHz, tasa de transmisión de 8 Mbps, la cual es la misma tasa para todos los usuarios. Adicionalmente, la prioridad de requerimiento de potencia se relaciona con el número del usuario, es decir, el usuario 1 requiere de un mayor segmento de potencia que al usuario 2, mientras que a este le corresponde una potencia menor al primero, y así sucesivamente. Con estos parámetros y lineamientos se obtuvieron los siguientes segmentos de potencia en Watts mostrados en la tabla 1-3. En base a lo calculado, se denotó que existe una variación grande entre el primero y segundo segmento de potencia, debido a la

prioridad que requiere el mismo, no obstante, esta diferencia decreció por la restricción de suma de segmentos de potencia, la cual no debe exceder la potencia de transmisión máxima. En este caso se obtuvo una sumatoria de 39,7988 W, siendo inferior a la potencia máxima.

Tabla 1-3: Segmentos de potencia obtenidos con el algoritmo original.

Nro. Segmentos de Potencia	Potencia [W]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
1	29,5038	-	-	-	-	-
2	29,5038	7,6385	-	-	-	-
3	29,5038	7,6385	1,9776	-	-	-
4	29,5038	7,6385	1,9776	0,512	-	-
5	29,5038	7,6385	1,9776	0,512	0,1326	-
6	29,5038	7,6385	1,9776	0,512	0,1326	0,0343

Realizado por: Armas, Roger, 2022.

3.1.4.2. Segmentos de potencia con el algoritmo modificado

La modificación que se realizó a la ecuación del algoritmo para el cálculo del segmento de potencia es el reemplazo de la operación suma por la multiplicación en el segundo factor como se muestra en la ecuación 2-3. Los parámetros necesarios para el cálculo de plantearon con los siguientes valores: Potencia de transmisión máxima 46 dBm (39,81 W), ancho de banda 10 MHz y las tasas de transmisión en este caso variaron dependiendo el número de segmentos de potencia.

$$s = \left(2^{\frac{c}{B}} - 1\right) \left(\frac{\sigma^2}{h_m} |s|_1\right)$$

Ecuación 2-3:

Segmentos de potencia, algoritmo modificado

Las tasas de transmisión necesarias para el algoritmo modificado aumentaron significativamente respecto a lo propuesto en el escenario del algoritmo original. La tabla 2-3 muestra los valores de estas tasas en Mbps.

Tabla 2-3: Tasas de transmisión requeridos para los usuarios en base al número de segmentos de potencia.

Nro. Segmentos de Potencia	Tasas de transmisión [Mbps]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
1	493,966	-	-	-	-	-
2	490,747	402,424	-	-	-	-
3	490,747	398,273	359,608	-	-	-
4	488,821	398,273	369,607	339,362	-	-
5	483,9666	402,424	375,456	349,362	319,68	-
6	480,747	405,643	375,457	349,362	319,7	300,11

Realizado por: Armas, Roger, 2022.

Tomando en cuenta las tasas de transmisión de datos de los usuarios y los demás parámetros, se realizó el cálculo de los segmentos de potencia según la ecuación 2-3, cuyos resultados se presentan en la tabla 3-3. Se apreció que las magnitudes de las potencias no varían significativamente entre lo obtenido por el algoritmo original y el modificado.

Tabla 3-3: Segmentos de potencia obtenidos con el algoritmo modificado.

Nro. Segmentos de Potencia	Potencia [W]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
1	29,5025	-	-	-	-	-
2	29,5030	7,6385	-	-	-	-
3	29,5030	7,6381	1,9777	-	-	-
4	29,5040	7,6384	1,9776	0,512	-	-
5	29,5037	7,6386	1,9776	0,512	0,1326	-
6	29,5030	7,6383	1,9776	0,512	0,1327	0,0343

Realizado por: Armas, Roger, 2022.

Mediante el algoritmo modificado se tuvo un mayor control sobre los requerimientos de tasas de transmisión y no se generaron valores muy diferentes a lo obtenido por el algoritmo original. Se tomó en cuenta los segmentos de potencia mostrados en la tabla 3-3 para la obtención de las curvas BER para la técnica OPDMA.

3.1.4.3. Curvas BER de OPDMA

En el gráfico 18-3 se puede observar las curvas BER de la señal OPDMA para un número de 2 usuarios, modulación QPSK, correspondientes a los resultados obtenidos por la simulación y las curvas de referencia planteadas en el artículo. Los parámetros como el ancho de banda, potencia de la señal y las tasas de transmisión fueron establecidos en base a lo obtenido con el algoritmo modificado para el cálculo de segmentos de potencia. Además, se consideró el efecto de un canal AWGN para realizar el análisis del rendimiento de la técnica. Respecto a la escala de los ejes, se consideró una escala lineal para una comparación más directa.

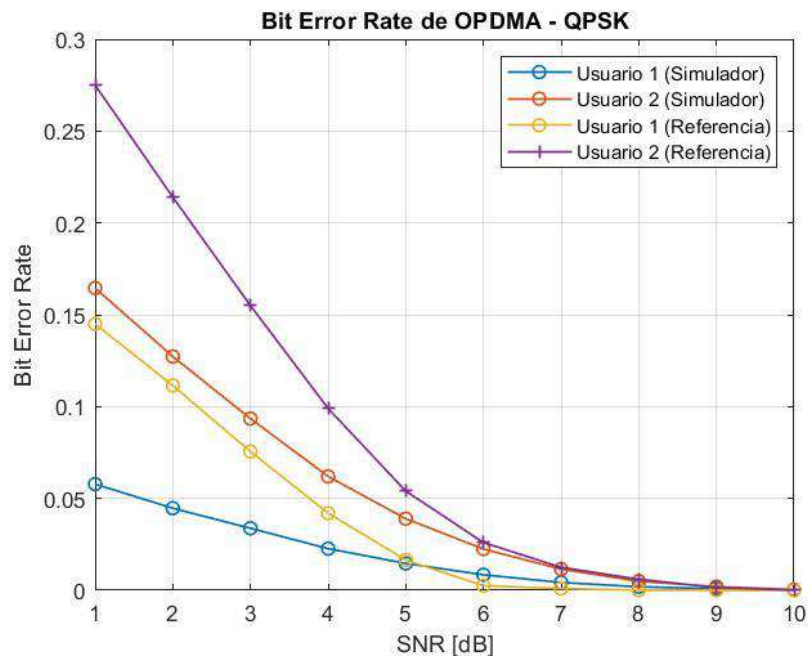


Gráfico 18-3: Comparación del BER en función del SNR de OPDMA con modulación QPSK y canal AWGN.

Realizado por: Armas, Roger, 2022.

La reducción en el BER es un resultado que se pudo denotar al comparar lo obtenido en el simulador con las curvas de referencia. Esta variación se debió a la metodología aplicada para la simulación de la técnica ya que puede variar según el autor. Por otra parte, la falta de información sobre los parámetros de simulación con los que fueron obtenidas las curvas de referencia es una causa de esta diferencia en los resultados. Se observó que para un BER de 0,05 o 5×10^{-2} , la señal OPDMA del simulador requirió un SNR de 1,5 dB para el usuario 1, mientras que la curva de referencia establece que es necesario un SNR de 3,76 dB, lo cual marca una diferencia de 2,26 dB entre las curvas. Sin embargo, para un valor SNR de 5 dB se presenció una similitud en el comportamiento del BER obtenido.

Con relación al usuario 2, en ambos resultados se observó que el BER tiende a aumentar de acuerdo con el número de usuarios. Esto se debe a la técnica de Cancelación Sucesiva de Interferencias SIC aplicada en la etapa de recepción donde se realizaron procesos de demodulación y resta de señales. Para el caso de un BER de $0,15$ o $1,5 \times 10^{-1}$, la señal OPDMA del simulador requirió un SNR de $1,39$ dB y en la curva de referencia se necesita de $3,08$ dB, evidenciando una diferencia de $1,69$ dB entre las curvas, misma que se redujo conforme aumenta el SNR hasta un valor de 6 dB donde las curvas empiezan a superponerse, lo cual implicó un rendimiento similar entre las señales OPDMA del simulador y del artículo a partir de este SNR.

En cuanto a la señal OPDMA con modulación 4-QAM, se realizó dos comparaciones en base a dos gráficas con número de usuarios distintos. Las curvas tanto del simulador como las de referencia son presentadas en los gráficos 19-3 y 20-3 para 2 y 5 usuarios respectivamente. En el análisis de la técnica con 2 usuarios se consideró un número de segmentos de potencia igual al número de usuarios y se analizó las curvas bajo el efecto de un canal AWGN.

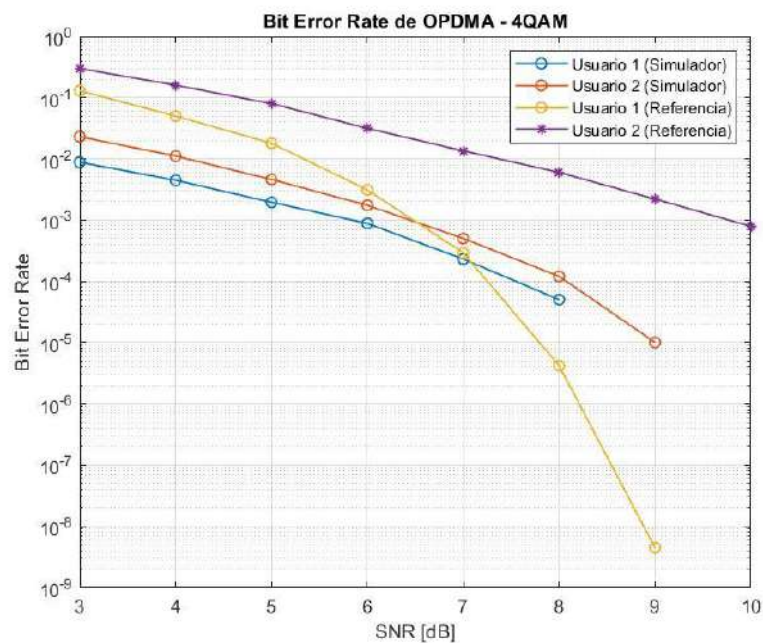


Gráfico 19-3: Comparación del BER en función del SNR de OPDMA con modulación 4QAM y canal AWGN.

Realizado por: Armas, Roger, 2022.

Se observó que las curvas obtenidas con el simulador distan en gran medida con las curvas de referencia tanto para el usuario 1 y el usuario 2. En cuanto a las curvas proporcionadas por el simulador, para lograr un BER de 10^{-3} el usuario 1 requirió un SNR de $5,85$ dB y el usuario 2 un valor de $6,45$ dB mientras que, en las curvas de referencia el usuario 1 requiere de $6,47$ dB de SNR mientras que el usuario 2 necesita un valor de $9,65$ dB. Con estos resultados se registró una

diferencia de 0,62 dB respecto al usuario 1, lo cual evidenció una aproximación del simulador a las curvas de referencia. Este comportamiento se observó a partir de un SNR igual a 6 dB y para un valor de 7,09 dB las probabilidades son similares. Por otro lado, en el usuario 2 la diferencia aumentó a un valor de 3,2 dB, dicha diferencia sigue una misma tendencia para todo el rango de SNR establecido lo cual ocasionó que no se aproxime la curva obtenida por el simulador con la curva de referencia. No obstante, se presencié el aumento del BER del usuario 2 respecto al usuario 1 en ambos resultados presentados, se sigue marcando lo cual implicó un comportamiento acorde a la metodología implementada en el receptor.

Finalmente, en el gráfico 20-3 se presenta las curvas correspondientes al BER en función del SNR de la señal OPDMA para una modulación 4-QAM, un número de 5 usuarios sobre un canal AWGN y L segmentos de potencia. Para este caso solo se muestran los resultados obtenidos por el simulador, debido a que analizar en el mismo rango de SNR del gráfico 17-3 no se obtendría una buena apreciación de las curvas. Se observó que el BER de los 5 usuarios aumenta en gran medida a comparación con las curvas de referencia debido al efecto de la técnica SIC en el receptor, sin embargo, se evidenció una tendencia a unificarse las curvas BER a partir de un SNR de 25 dB, donde el BER converge entre los usuarios.

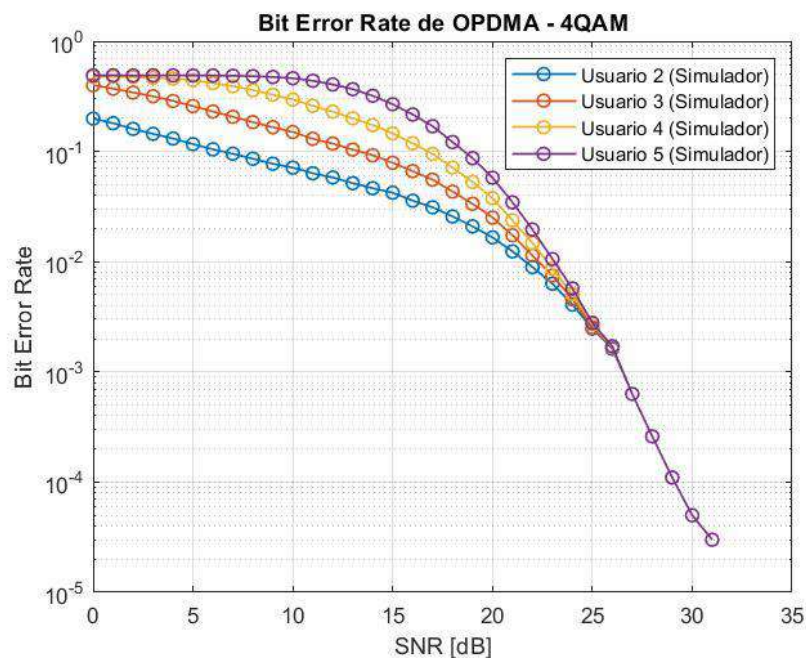


Gráfico 20-3: BER en función del SNR obtenido del simulador OPDMA con modulación 4-QAM para un número de 5 usuarios y canal AWGN.

Realizado por: Armas, Roger, 2022.

Del mismo modo, en las curvas de referencia del gráfico 17-3 se observó el mismo comportamiento de las curvas para un rango de SNR menos elevado que lo obtenido por el

simulador. Dicha convergencia del BER se generó a partir de un SNR de 6 dB, es decir, un 19 dB menos que lo requerido por la señal OPDMA del simulador. De igual manera que en el resto de los casos, este comportamiento de las curvas se debe al efecto de la técnica SIC, mismo que al aumentar el SNR, el nivel de BER de todos los usuarios se unificó en un punto determinado y continua bajo un mismo comportamiento.

3.2. Análisis de resultados del simulador

En esta sección se muestran todas las curvas BER calculadas y mostradas mediante el simulador desarrollado en este trabajo para los parámetros de entrada que el usuario ingrese.

3.2.1. Escenarios de simulación

Los parámetros de entrada para cada simulación dependen del requerimiento de la técnica planteada, por lo tanto, se detallan de manera más específica en los siguientes apartados.

- CDMA

Entre los parámetros generales que se aplicaron en todas las técnicas están el número de usuarios que tiene un máximo de 6 para establecer un máximo estándar entre todas las técnicas y la cantidad de bits generados es de 10^5 . Por otro lado, el factor de ensanchamiento se configuró en base a la tasa de bits requerida para lo cual se tomó como referencia la tasa de chips mencionada en el Release 4 de 3GPP para UMTS 3.84Mcps (3GPP, 2004). Los valores de las tasas de transmisión en base al factor de ensanchamiento se muestran en la tabla 1-3. Los tipos de modulación seleccionados para la simulación se eligieron en base a la frecuencia de aplicabilidad de estos como BPSK, QPSK, 16-QAM y 64-QAM. Finalmente se presenta las opciones de canal a simular los cuales son AWGN, Rice y Rayleigh.

Tabla 4-3: Tasas de transmisión en función del factor de ensanchamiento para una tasa de chips de 3.84 Mcps.

Factor de ensanchamiento SF	Tasa de transmisión
4	960 kbps
8	480 kbps
16	240 kbps
32	120 kbps
64	60 kbps

Realizado por: Armas, Roger, 2022.

- OFDMA

En esta técnica se estableció escenarios con distintos valores en los parámetros adicionales al número de usuarios y cantidad de bits generados. Los tamaños de IFFT elegidos para la simulación de la técnica se muestran en la tabla 2-3 donde se observa los valores en función del ancho de banda y el número de subportadoras. Por otro lado, para la duración del prefijo cíclico se proporciona fracciones como 1/2, 1/4, 1/8, 1/16, 1/32 y 1/64 como representación de un segmento del símbolo OFDM en función del tiempo. Los esquemas de modulación, al igual que en CDMA, son BPSK, QPSK, 16-QAM y 64-QAM. En cuanto al canal de comunicación, se puede seleccionar entre los canales planteados.

Tabla 5-3: Tamaño de IFFT y FFT para downlink.

Ancho de Banda (MHz)	Número de Bloques de Recursos (RBs)	Número de subportadoras	Tamaño IFFT Downlink
1.4	6	72	128
3	15	180	256
5	25	300	512
10	50	600	1024

Fuente: Wang, et al.,2014, p.3934.

Realizado por: Armas, Roger, 2021.

- SCMA

Para SCMA el número de usuarios es fijo ya que al disminuir el valor se requiere el diseño de nuevos libros de códigos. En cuanto al flujo de información a transmitir, se estableció una cantidad de 10^4 símbolos en decimal para reducir el tiempo de ejecución de la simulación y cuya conversión a binario está conformado por 2 bits. Los canales por simular en esta técnica son AWGN y Rayleigh debido a los estudios realizados sobre estos en SCMA.

- OPDMA

Para la simulación de esta técnica se consideró los siguientes parámetros: máximo 6 usuarios, 10^5 bits generados aleatoriamente, potencia máxima de transmisión 46 dBm, ancho de banda de 10 MHz y tasas de transmisión basados en la tabla 2-3. Los esquemas de modulación elegidos son QPSK y 4-QAM mientras que el canal a simular es AWGN. Estos parámetros fueron seleccionados debido a los estudios previos realizados, sin embargo, pueden ser modificados por el usuario con el fin de visualizar el comportamiento de las curvas.

3.2.2. Curvas BER

En esta sección se analiza las curvas BER que se visualizan en el simulador, donde cada gráfica puede ser obtenida en función al número de usuarios con el objetivo de evaluar el rendimiento de la técnica acorde a este parámetro. Se presentan una cantidad de 3 curvas para las simulaciones de las técnicas CDMA, OFDMA, y SCMA, mientras que para OPDMA se presenta una curva. El proceso paso a paso para llevar a cabo las simulaciones se detalla en el anexo D.

3.2.2.1. CDMA

Para las simulaciones de los distintos escenarios de esta técnica se plantearon los siguientes parámetros: factor de ensanchamiento 16 para una tasa de transmisión de 240 kbps, modulación QPSK, canales AWGN, Rice y Rayleigh para un número de 6 usuarios. En el gráfico 21-3 se presenta el gráfico que se obtiene de la interfaz gráfica del simulador.

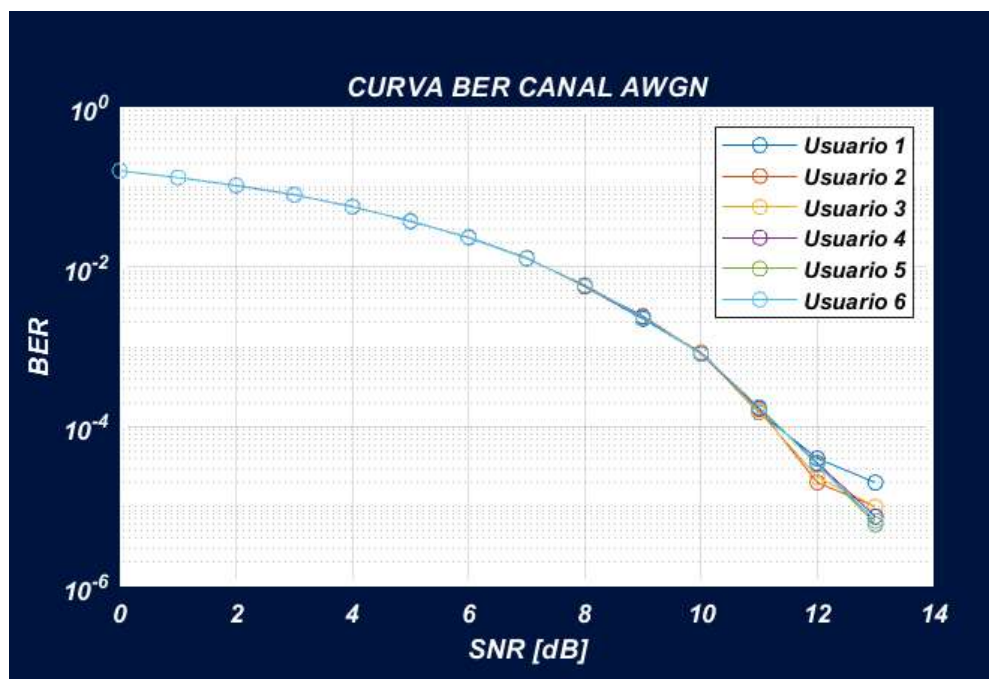


Gráfico 21-3: Gráfica desplegada en la interfaz gráfica de la simulación CDMA con canal AWGN.

Realizado por: Armas, Roger, 2022.

La disposición de las curvas BER en el gráfico 21-3 permitió visualizar el desempeño de la técnica de Acceso Múltiple Ortogonal para un número específico de usuarios y un canal de ruido AWGN. El comportamiento de las curvas resultó similar en todos los usuarios evidenciando estabilidad para el número establecido, no obstante, a partir de 12 dB de SNR se observó una pequeña

variación entre las curvas. En cuanto a los valores requeridos de SNR para alcanzar las tasas de error de bit visualizadas, se presenta una tabla de resumen para los requerimientos aproximados para los BER establecidos. Estos valores obtenidos son aproximados, para observar los resultados exactos el simulador proporciona la opción de guardarlos en un documento de formato compatible con Microsoft Excel.

Tabla 6-3: Resumen de SNR requeridos para varias tasas BER de CDMA con canal AWGN.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	2	2	2	2	2	2
10^{-2}	7,25	7,25	7,25	7,25	7,25	7,25
10^{-3}	9,8	9,8	9,8	9,8	9,8	9,8
10^{-4}	11,20	11,15	11,15	11,20	11,20	11,20
10^{-5}	-	13	13	12,7	12,6	12,62

Realizado por: Armas, Roger, 2022.

Las curvas obtenidas para un canal Rice se presentan en el gráfico 22-3 para un factor Rician igual a 6 dB y con los mismos parámetros del primer escenario. Del mismo modo que en el canal AWGN, se presencié una misma tendencia de disminución del BER para todos los usuarios hasta un valor SNR de 11 dB donde se presentan fluctuaciones que van creciendo conforme el SNR aumenta. Por otro lado, se observó que la curva adquiere un bajo declive a comparación con lo obtenido en el canal AWGN. Para un valor de BER de 10^{-2} se requirió de 9,8 dB evidenciando una diferencia de 2,55 dB para niveles altos de tasas, mientras que para una tasa de 10^{-5} , el sistema necesitó un valor de SNR que oscila entre 28,52 y 29 dB dependiendo el usuario, donde se obtuvo una diferencia muy alta de aproximadamente 16 dB entre las respuestas de los canales. En la tabla 7-3 se resumen los requerimientos de SNR para varios niveles de BER.

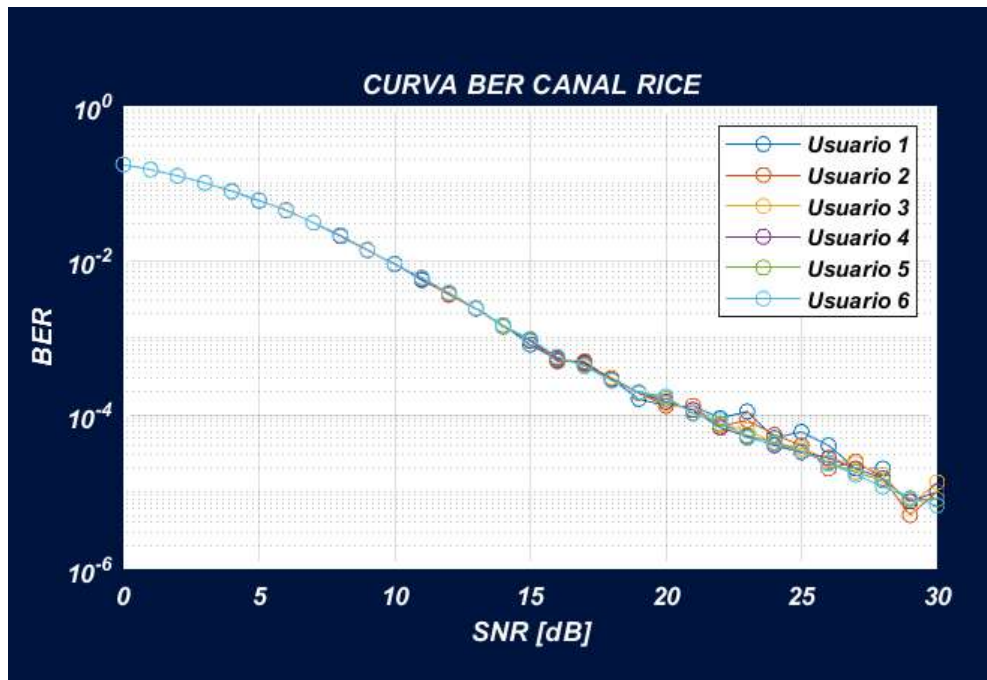


Gráfico 22-3: Gráfica desplegada en la interfaz gráfica de la simulación CDMA con canal Rice ($K = 6$ dB).

Realizado por: Armas, Roger, 2022.

Tabla 7-3: Resumen de SNR requeridos para varias tasas BER de CDMA con canal Rice.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	3	3	3	3	3	3
10^{-2}	9,86	9,86	9,83	9,83	9,86	9,86
10^{-3}	14,75	14,75	14,75	14,76	14,77	14,75
10^{-4}	21,8	21,3	21,3	21,25	21,25	21,18
10^{-5}	-	28,15	28,15	28,62	28,62	28,55

Realizado por: Armas, Roger, 2022.

Por último, la simulación de CDMA con canal Rayleigh se presenta en el gráfico 23-3 cuyo escenario consta de los mismos parámetros de entrada que los escenarios anteriores. El simulador mostró una tendencia similar en cuanto al comportamiento de las curvas en función del número de usuarios, es decir, el BER no varía en gran medida para el número de usuario establecidos. Por otro lado, se notó el efecto del canal de desvanecimiento al producir un aumento del BER para el rango de SNR establecido a comparación con los otros canales. Para un BER de 10^{-1} se requirió un SNR de 5,3 dB a comparación con el canal Rice donde se calculó una diferencia de 2,3 dB y 3,3 dB con el canal AWGN. Esta diferencia aumentó proporcionalmente al SNR debido a las características de los canales inalámbricos.

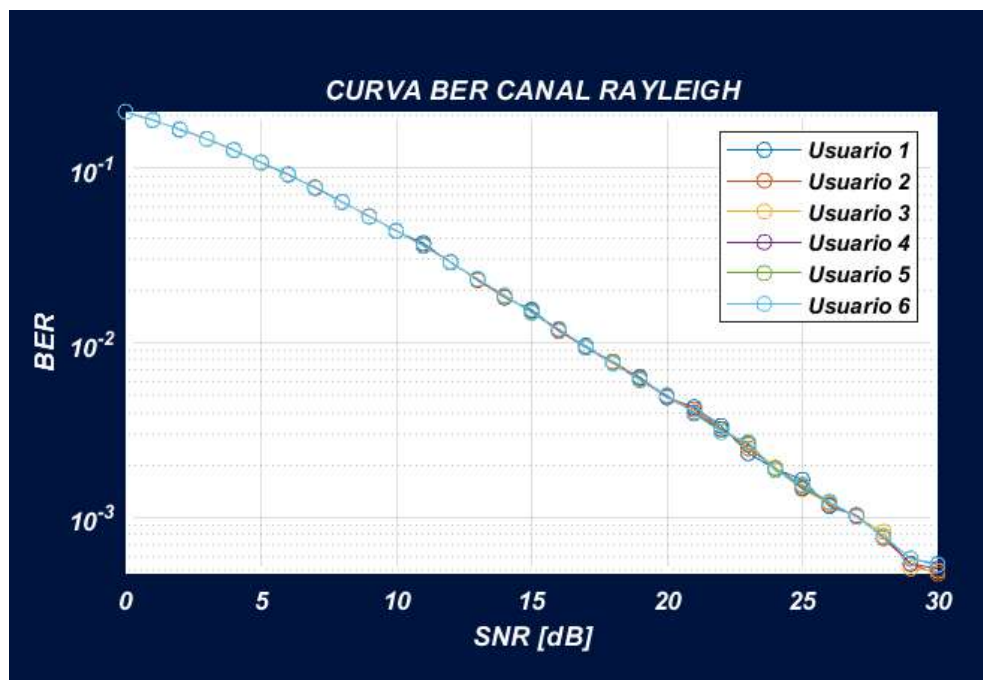


Gráfico 23-3: Gráfica desplegada en la interfaz gráfica de la simulación CDMA con canal Rayleigh.

Realizado por: Armas, Roger, 2022.

Para una mejor visualización de los resultados, en la tabla 5-3 se resume los valores de SNR necesarios para lograr distintos niveles de BER que se pueden observar en el gráfico 23-3. Los valores presentados son aproximados ya que son obtenidos de las curvas obtenidas por el simulador.

Tabla 8-3: Resumen de SNR requeridos para varias tasas BER de CDMA con canal Rayleigh.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	5,3	5,3	5,3	5,3	5,3	5,3
10^{-2}	16,95	16,95	16,95	16,95	16,95	16,96
10^{-3}	27,1	27,05	27,06	27,11	27,1	27,1

Realizado por: Armas, Roger, 2022.

3.2.2.2. OFDMA

Los parámetros establecidos para la simulación de ésta técnicas son similares a la técnica CDMA en cuanto al número de usuarios, cantidad de bits, esquema de modulación y canales inalámbricos. Adicionalmente se estableció un tamaño de FFT de 512 correspondiente a un ancho de banda de

5 MHz la cual requiere un total de 300 subportadoras. En el gráfico 24-3 se presenta las curvas presentadas en la sección Resultados del simulador.

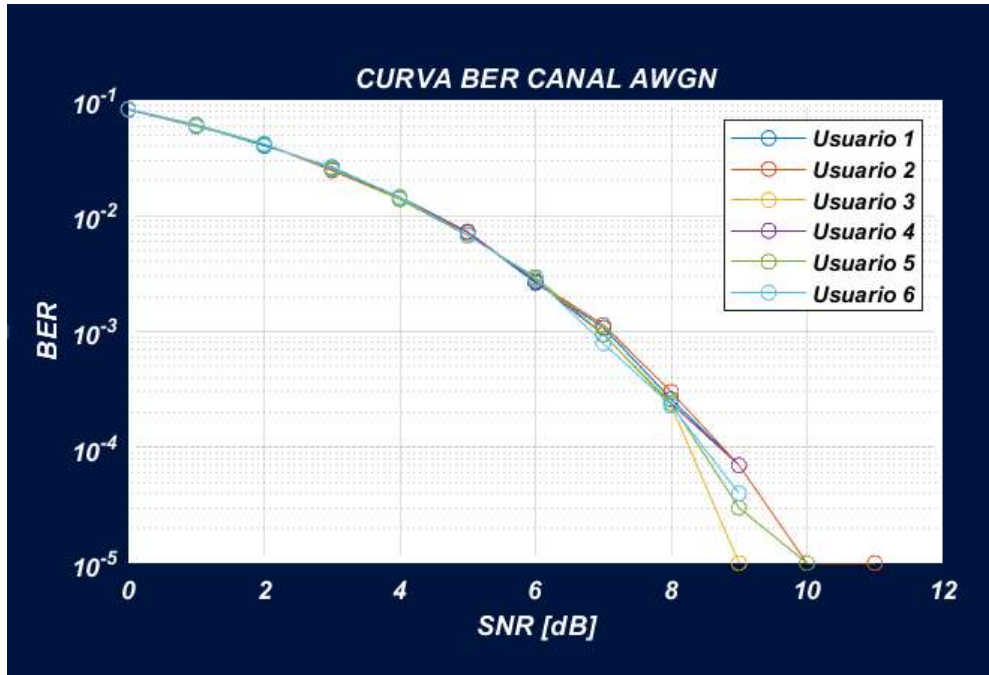


Gráfico 24-3: Gráfica desplegada en la interfaz gráfica de la simulación OFDMA con canal AWGN.

Realizado por: Armas, Roger, 2022.

Se observó que las curvas obtenidas para cada usuario se encuentran superpuestas, es decir, tienen una tasa de errores de bit similar hasta un valor SNR de 7 dB, a partir de este punto, se presentó una variación ligera entre ellas. Esto ocurrió debido al número de usuarios que se establece como parámetro de entrada, lo cual provocó una variación en el BER. Por otro lado, se observó que para el primer caso de un nivel BER de 10^{-2} se requirió un SNR de 4,5 dB para todos los usuarios, mientras que para la técnica CDMA se necesitó de un valor de 7,5 dB, por lo tanto, la técnica OFDMA con los parámetros establecidos evidencian una mejora en el BER. Del mismo modo, para un BER de 10^{-4} el SNR requerido varía respecto al usuario, para el usuario 1, 2 y 5 se requirió un SNR de 8,8 dB, mientras que para los usuarios 3, 5 y 6 se necesitó de 8,3, 8,43 y 8,5 dB respectivamente. En la tabla 6-3 se presenta de manera resumida los requerimientos de SNR para cada usuario en base a una tasa de errores.

Tabla 9-3: Resumen de SNR requeridos para varias tasas BER de OFDMA con canal AWGN.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-2}	4,5	4,5	4,5	4,5	4,5	4,5
10^{-3}	7,01	7,06	7	7,01	7	6,81
10^{-4}	8,8	8,8	8,3	8,43	8,8	8,5
10^{-5}	-	10	9	-	10	-

Realizado por: Armas, Roger, 2022.

En cuanto al canal Rice, en el gráfico 25-3 se presentan las curvas BER de OFDMA para un escenario de simulación con parámetros de entrada idénticos al escenario para el canal AWGN. Se visualizó un aumento en el BER para todos los usuarios, cuyo efecto es característico de un canal de desvanecimiento. Las curvas de cada usuario se superponen hasta un SNR de 11 dB y para valores mayores a este, el BER cambió ligeramente, no obstante, resultó un comportamiento irregular a partir de 18 dB de SNR. Como primer caso, para un BER de 10^{-1} la señal OFDMA requirió de un SNR de 7,95 dB, lo cual evidenció el incremento de la tasa de errores respecto al canal AWGN. Para el caso de una tasa de 10^{-2} , el requerimiento de SNR fluctuó en cuanto a los valores necesarios por cada usuario. El usuario 5 necesitó un SNR de 14,46 dB para el nivel establecido, mientras que para el usuario 1 se necesitó un valor de 14,75 dB, por lo que estas variaciones responden al canal y el número de usuarios para la técnica OFDMA. Los resultados para todos los usuarios se presentan en la tabla 7-3.

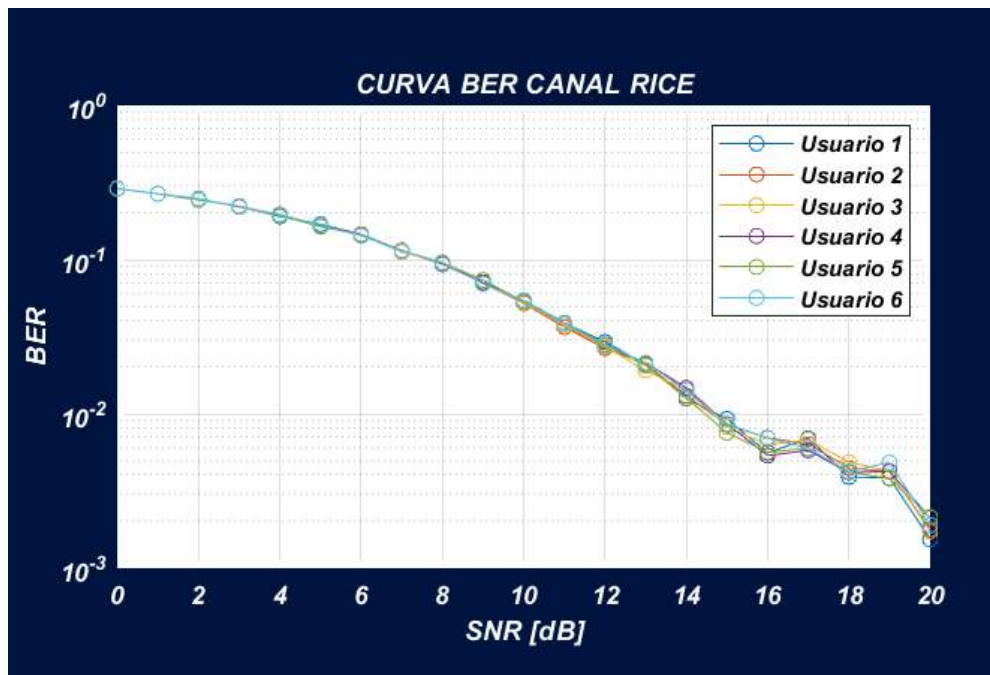


Gráfico 25-3: Gráfica desplegada en la interfaz gráfica de la simulación OFDMA con canal Rice (K=2).

Realizado por: Armas, Roger, 2022.

Tabla 10-3: Resumen de SNR requeridos para varias tasas BER de OFDMA con canal Rice.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	7,66	7,66	7,74	7,71	7,75	7,71
10^{-2}	14,76	14,63	14,66	14,74	14,46	14,71

Realizado por: Armas, Roger, 2022.

Para finalizar con esta técnica, las curvas obtenidas de la simulación de OFDMA en un canal Rayleigh se visualiza en el gráfico 26-3 cuyo escenario consta de parámetros semejantes a los ya planteados. El simulador proporcionó un aumento de BER para este tipo de canal respecto a los demás debido a su característica de no poseer una señal con línea de vista entre el transmisor y receptor. Para un BER de 10^{-1} se requirió de valores alrededor de 9 dB para cada usuario, mientras que para un BER de 10^{-2} se presenció una mayor variación del SNR necesario para alcanzar la tasa planteada cuyos valores sondean los 16,2 dB, es decir, 2 dB de diferencia respecto al canal Rice. En la tabla 8-3 se puede apreciar con mayor exactitud los valores correspondientes a cada usuario. Si bien es cierto, el BER empieza a variar abruptamente a partir de 14 dB, la respuesta del simulador mostró una superposición de las curvas en gran parte del rango de SNR lo cual evidencia que la técnica trabaja de manera eficiente para el número de usuarios establecido.

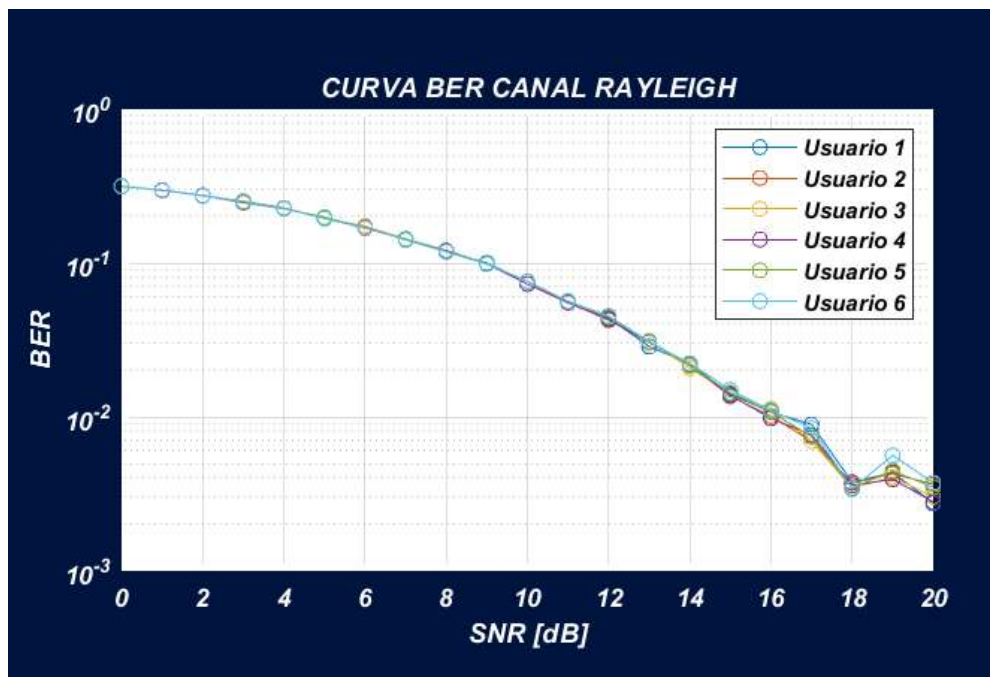


Gráfico 26-3: Gráfica desplegada en la interfaz gráfica de la simulación OFDMA con canal Rayleigh.

Realizado por: Armas, Roger, 2022.

Tabla 11-3: Resumen de SNR requeridos para varias tasas BER de OFDMA con canal Rayleigh.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	9	8,92	8,94	8,95	8,956	8,92
10^{-2}	16,39	16,04	16,25	15,94	16,2	16,36

Realizado por: Armas, Roger, 2022.

3.2.2.3. SCMA

La simulación de SCMA tiene solo 3 escenarios principales correspondiente a los canales inalámbricos, para cada escenario se consideró el mismo número de usuarios debido a restricciones del libro de códigos. En el gráfico 27-3 se presenta las curvas BER de SCMA bajo efectos del canal AWGN para los usuarios planteados.

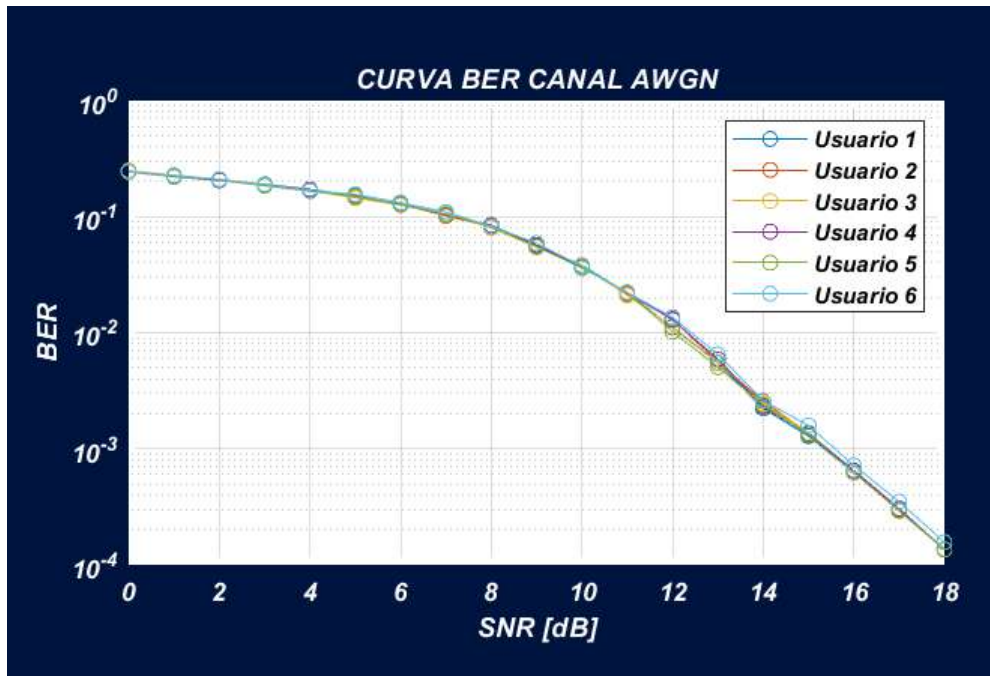


Gráfico 27-3: Gráfica desplegada en la interfaz gráfica de la simulación SCMA con canal AWGN.

Realizado por: Armas, Roger, 2022.

En SCMA las curvas BER no presentaron grandes variaciones entre los usuarios, por lo que en gran parte del intervalo de SNR, las curvas se superponen. Para el caso de una tasa de errores de bit de 10^{-1} , los usuarios requirieron de un SNR de 7 dB aproximadamente cuando se utilizó esta técnica, lo cual representa un aumento en el BER a comparación con las técnicas de Acceso Múltiple Ortogonales, no obstante, el comportamiento es más estable respecto a las demás técnicas. Del mismo modo, para un BER de 10^{-2} , los usuarios requirieron valores de SNR más distantes entre sí, rondando cerca de 12,25 dB para cada uno, esta tendencia se ve marcada también para un nivel de BER de 10^{-3} donde los usuarios necesitaron un SNR alrededor de 15,4 dB excepto el usuario 6 mismo que experimentó un aumento en el BER requiriendo así un SNR de 15,6 dB. En la tabla 9-3 se puede observar detalladamente los SNR necesarios para distintas tasas de errores de bit cuando se simula esta técnica.

Tabla 12-3: Resumen de SNR requeridos para varias tasas BER de SCMA con canal AWGN.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	7,2	7,06	7,2	7,2	7,2	7,34
10^{-2}	12,15	12,31	12,17	12,33	12,04	12,41
10^{-3}	15,35	15,41	15,4	15,41	15,37	15,58

Realizado por: Armas, Roger, 2022.

Para la simulación de la técnica SCMA bajo efectos del canal Rice se estableció un factor K igual a 2 dB, tanto el número de símbolos generados como el número de usuarios son los mismos parámetros empleados en el escenario con canal AWGN para poder visualizar el rendimiento de la técnica en condiciones equivalentes. Las curvas que se muestran en la interfaz gráfica del simulador se muestran en el gráfico 28-3 correspondiente a la tasa de errores de bit BER en función del SNR para cada usuario.

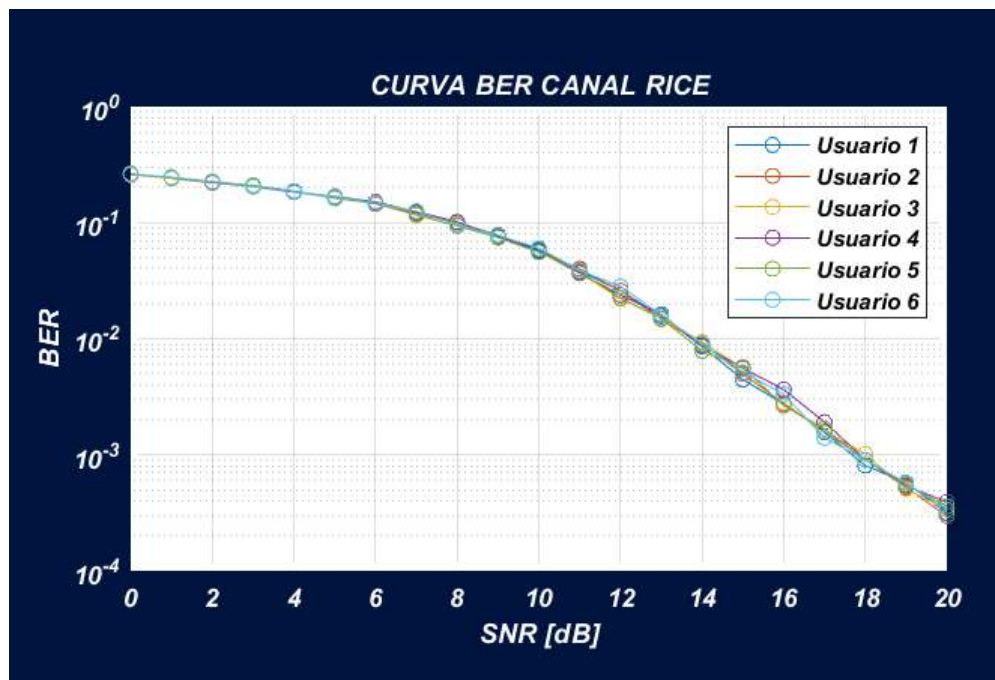


Gráfico 28-3: Gráfica desplegada en la interfaz gráfica de la simulación SCMA con canal Rice.

Realizado por: Armas, Roger, 2022.

Del gráfico se observó que para un BER de 10^{-1} , la señal SCMA requirió un valor medio de 7,88 dB de SNR. A partir de 11 dB en el SNR, se apreció que las curvas cambian su comportamiento entre los usuarios, por lo que los requerimientos varían para el nivel de BER establecido en cada caso. Debido a esto, los valores requeridos fluctúan significativamente, siendo el caso de una tasa de errores de 10^{-2} , donde el usuario 1 requirió un SNR de 13,75 dB para alcanzar el BER planteado, el usuario 3 necesitó un SNR de 13,87 dB y el usuario 5 requirió un valor de 13,62 dB. Este comportamiento se evidenció también en el caso de que se requiera un BER de 10^{-3} , en base al gráfico el usuario 1 es el que menor SNR requiere con un valor de 17,53 dB mientras que

el usuario 3 requirió del mayor SNR para esta tasa, con un valor de 18,03 dB. A comparación con el canal AWGN, el canal Rice presentó una diferencia de entre 0,5 a 2,5 dB en cada caso, lo cual evidenció el aumento del BER por efecto del canal. En la tabla 10-3 se puede observar los requerimientos de SNR en función del BER con mayor detalle.

Tabla 13-3: Resumen de SNR requeridos para varias tasas BER de SCMA con canal Rice.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	7,96	7,91	7,94	8,05	7,71	7,84
10^{-2}	13,75	13,78	13,87	13,77	13,62	13,81
10^{-3}	17,53	17,84	18,03	17,87	17,84	17,77

Realizado por: Armas, Roger, 2022.

Por último, los efectos del canal Rayleigh sobre la técnica SCMA se pueden visualizar en el gráfico 29-3, donde se denotó un ligero aumento del BER respecto al canal Rice. Las curvas de cada usuario se superponen entre si evidenciando estabilidad del sistema para la cantidad de usuarios establecido.

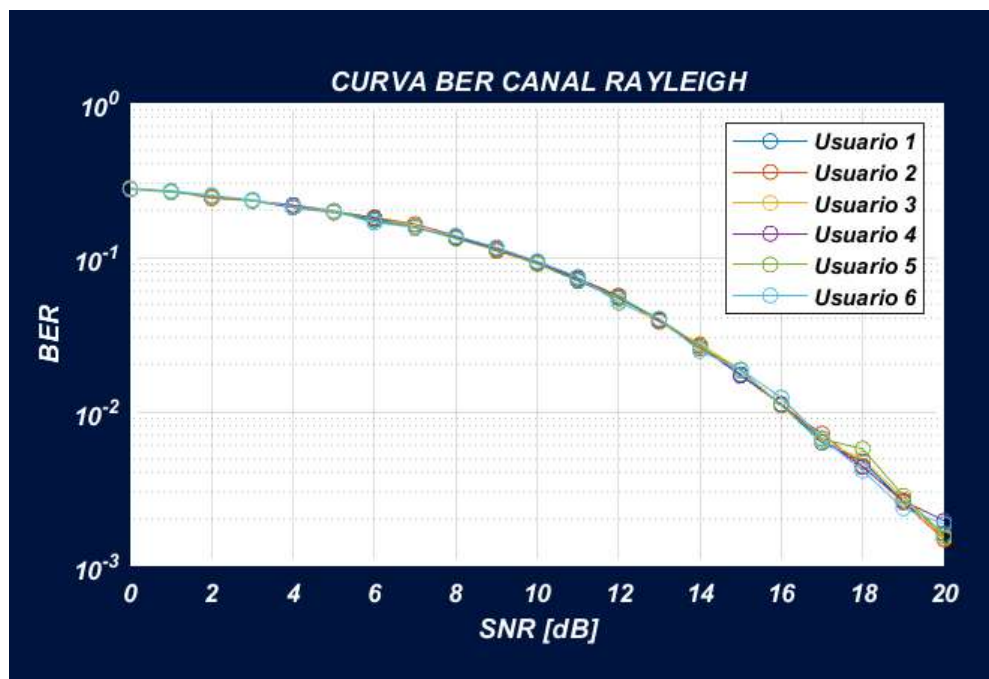


Gráfico 29-3: Gráfica desplegada en la interfaz gráfica de la simulación SCMA con canal Rayleigh.

Realizado por: Armas, Roger, 2022.

Adicionalmente, se observó que para un BER de 10^{-1} la señal SCMA requiere un SNR de 9,62 dB aproximadamente, este valor presentó variaciones en mínima medida entre los usuarios debido al rendimiento de la simulación con el número de usuarios establecido. Del mismo modo, para una tasa de errores de bit de 10^{-2} se necesitó un SNR alrededor de 16,25 dB el cual oscila de igual manera, para el número de usuarios correspondiente. Se evidenció una diferencia de 2 y 4 dB para los casos de BER analizados respectivamente, en relación con lo obtenido para el canal

AWGN, lo cual afirma el aumento de BER como efecto del canal en el rendimiento de la técnica de Acceso Múltiple No-ortogonal NOMA. Por otro lado, se obtuvo una ligera diferencia de 1,57 dB para el primer caso de BER respecto al canal Rice y 2,39 dB para el segundo caso de BER que se puede denotar en el gráfico.

Tabla 14-3: Resumen de SNR requeridos para varias tasas BER de SCMA con canal Rayleigh.

BER	SNR [dB]					
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Usuario 5	Usuario 6
10^{-1}	9,53	9,52	9,61	9,64	9,52	9,72
10^{-2}	16,14	16,18	16,28	16,19	16,14	16,32

Realizado por: Armas, Roger, 2022.

3.2.2.4. OPDMA

El simulador de la técnica OPDMA cuenta con diferentes escenarios en base al ancho de banda, tasa de transmisión y potencia de la señal que el usuario ingrese en la interfaz gráfica. No obstante, las tasas de transmisión disponibles para realizar una simulación eficiente están limitadas y dependen del ancho de banda. En este caso se consideró un número de 6 usuarios, esquema de modulación QPSK, potencia de la señal de 46 dBm, ancho de banda de 10 MHz y tasas de transferencia de datos proporcionadas por la tabla 2-3. En el gráfico 30-3 se presenta las curvas BER desplegada en la interfaz gráfica del simulador.

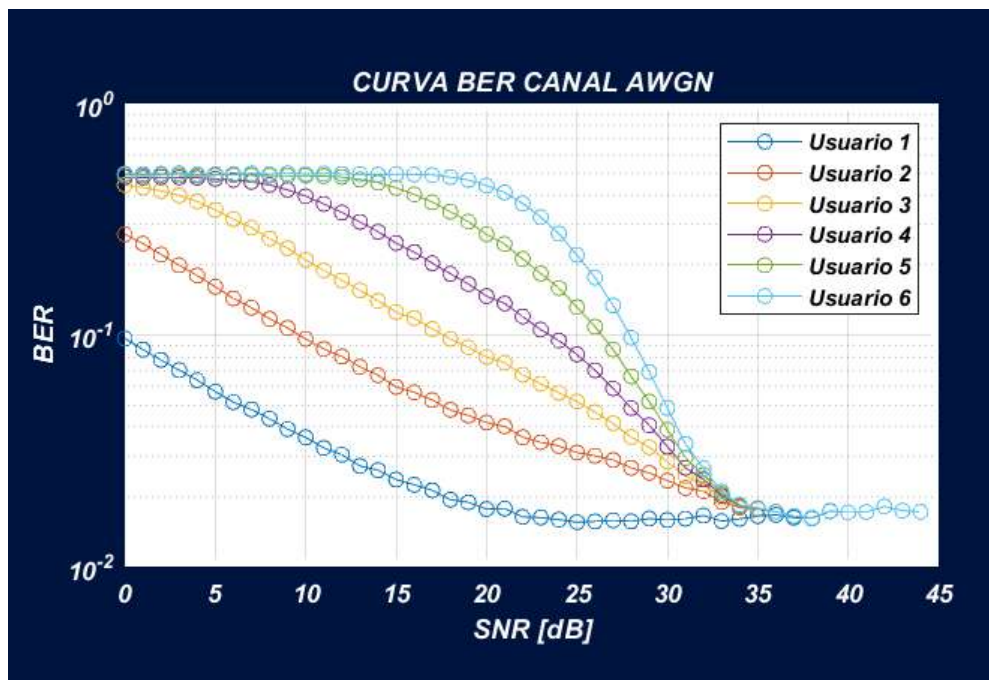


Gráfico 30-3: Gráfica desplegada en la interfaz gráfica de la simulación OPDMA con canal AWGN.

Realizado por: Armas, Roger, 2022.

Del gráfico se observó que el BER aumenta en gran medida cuando se simula la técnica para el número de 6 usuarios. Esto se debe al efecto de la técnica de decodificación que se implementó en el receptor denominada SIC, esto generó que al eliminar las señales que son tratadas como interferencias no se realice una correcta extracción de los bits de información, haciendo de este sistema ineficiente para el número de usuarios planteados. Sin embargo, se mantuvo la tendencia marcada en los gráficos 19-3 y 20-3 el cual hace referencia al incremento del BER entre los usuarios. Otra posible causa de este resultado son los niveles de potencia generados por el algoritmo con las tasas de transmisión planteadas, éstas pueden no ser lo suficientemente altas o diferenciables entre las potencias de las señales de los usuarios, lo cual provocó un bajo rendimiento del receptor SIC.

Adicionalmente, en el caso para un nivel BER de 10^{-1} el usuario 2 requirió de un SNR de 9,5 dB, el usuario 3 necesitó un valor de 17,5 dB, el usuario 4 un SNR de 23,92 dB, el usuario 5 requiere de un valor SNR de 26,51 dB y, por último, el usuario 6 requirió un SNR de 28,11 dB para lograr la tasa de errores especificada. La diferencia en el requerimiento de SNR entre el usuario 2 y 3 es de 8 dB, mientras que entre los usuarios 3 y 4 distan con 6,42 dB, esta diferencia disminuyó entre los demás usuarios, llegando a un valor de 1,6 dB entre el usuario 5 y 6. Este comportamiento evidenció una tendencia marcada en la convergencia de las curvas BER a partir de un punto en específico del rango de SNR.

Para la simulación de la técnica OPDMA bajo los efectos de un canal Rice se estableció un factor K de 4 dB y los mismos parámetros de entrada del escenario con canal AWGN. En el gráfico 31-3 se presentan las curvas correspondientes a la gráfica del BER en función del SNR que se visualiza en la interfaz gráfica del simulador. Se pudo observar que el requerimiento de SNR para los niveles de BER aumentaron significativamente por lo que el descenso de este tiene una menor inclinación respecto a lo obtenido en el canal AWGN, no obstante, el BER se mantiene en el rango de 0 a 10^{-2} . A partir de un SNR de 40 dB las curvas empiezan a converger y se denotó la tendencia a tener el mismo BER en todos los usuarios para SNR elevados. Para una tasa de 10^{-1} , la señal OPDMA del usuario 1 necesitó un valor de 1,21 dB SNR, por otro lado, el usuario 6 es el que mayor SNR requirió con un valor de 29,79 dB lo cual implicó una diferencia de 1,68 dB respecto a lo obtenido en el canal AWGN para el mismo usuario.

Las pruebas realizadas en artículos de referencia solo realizan simulaciones con un máximo de 5 usuarios debido al efecto de error de propagación donde al existir un error en la demodulación en uno de los usuarios, este se va acumulando. De acuerdo con Han, et al., (2021, p. 3332), esta acumulación de errores se considera uno de los principales inconvenientes de los sistemas

NOMA/OPDMA, lo cual se puede evidenciar en los resultados obtenidos por este trabajo a pesar de no coincidir con el comportamiento de las curvas.

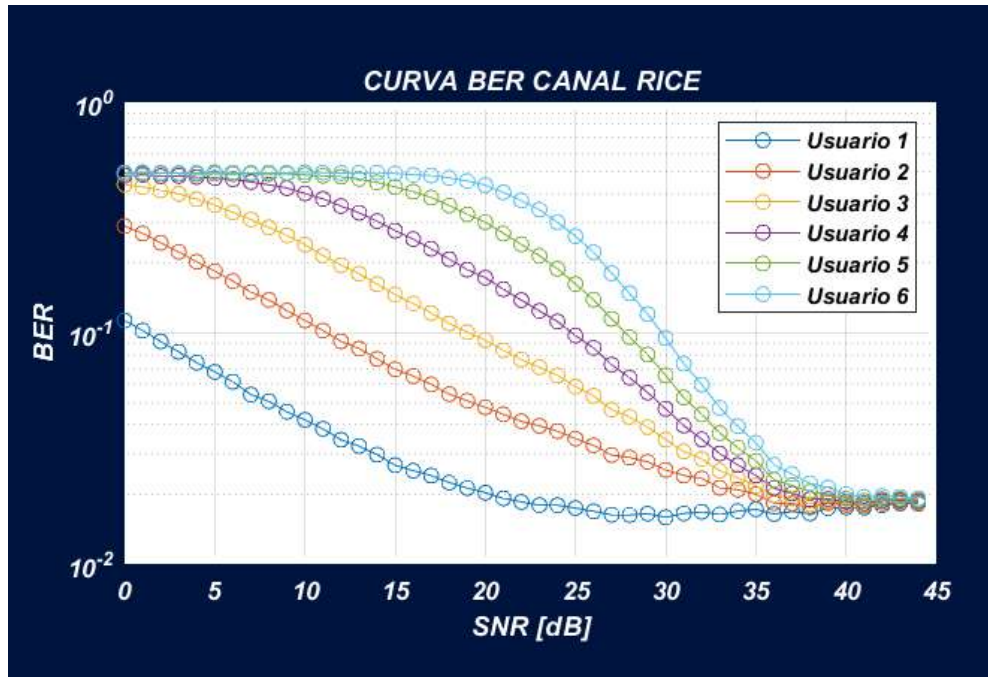


Gráfico 31-3: Gráfica desplegada en la interfaz gráfica de la simulación OPDMA con canal Rice.

Realizado por: Armas, Roger, 2022.

En el gráfico 32-3 se muestra las curvas del BER de los usuarios bajo los efectos de un canal de desvanecimiento Rayleigh. Se presenció un aumento del BER más notable entre los usuarios lo cual implica mayor requerimiento de SNR en un sistema de comunicaciones inalámbrico con técnica de acceso múltiple NOMA/OPDMA. En este escenario de simulación, las curvas de BER no llegan a converger completamente, pero si se logró apreciar la tendencia a unificarse la tasa de errores en todos los usuarios para valores de SNR más elevados. Para una tasa de 10^{-1} , el usuario 1 requirió de un SNR de 3,76 dB, lo cual hizo que exista una diferencia de 2,55 dB respecto a lo obtenido con el canal Rice. Del mismo modo, el usuario 6 para lograr la misma tasa requirió un SNR de 32,74 dB, significando una diferencia igual a 2,95 dB con el escenario de simulación anterior y 4,63 dB respecto al canal AWGN.

Por otro lado, también se pudo apreciar que los usuarios 4, 5 y 6 tienen un rendimiento similar hasta un SNR de 6 dB, en este rango los usuarios 1, 2 y 3 tienen una alta tasa de errores por lo que afecta al BER del resto de usuarios, sin embargo, cuando la curva de los primeros usuarios empieza a decrecer, la tasa de los demás disminuyó levemente, evidenciando así la acumulación de errores provocadas por la técnica SIC.

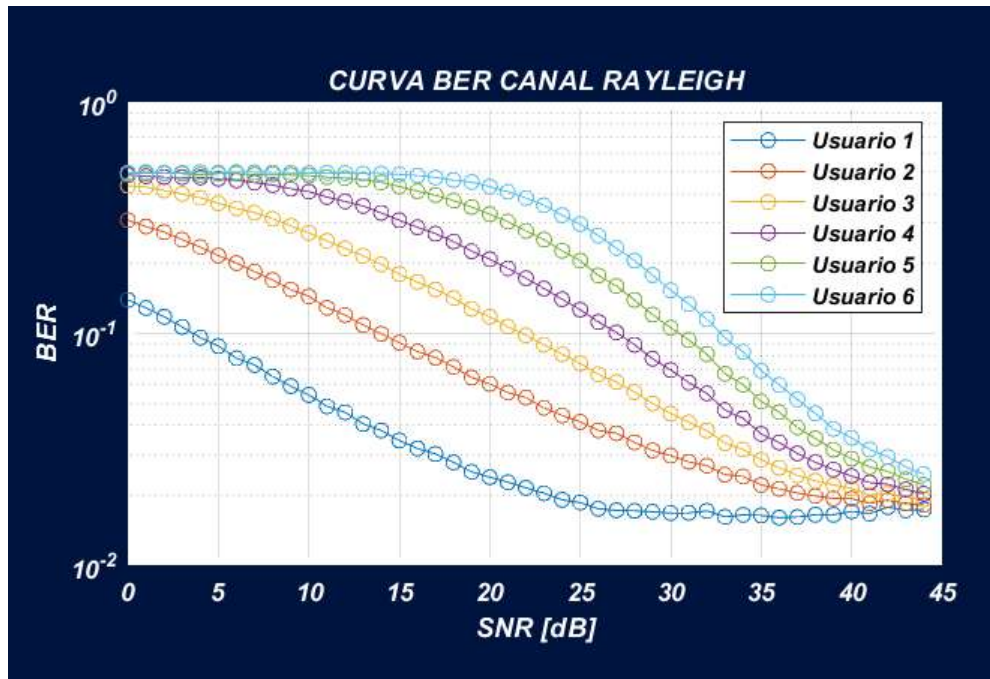


Gráfico 32-3: Gráfica desplegada en la interfaz gráfica de la simulación OPDMA con canal Rayleigh.

Realizado por: Armas, Roger, 2022.

3.2.3. Comparación de las técnicas según el canal

En base a los resultados obtenidos del simulador con los escenarios planteados, se realizó una comparación de las curvas BER de las técnicas OMA y NOMA planteadas donde se determinó el rendimiento de las técnicas en un canal de ruido AWGN, y los canales de desvanecimiento Rice y Rayleigh. En el gráfico 33-3 se puede visualizar las curvas para un canal AWGN. En esta se observó que para un nivel de BER de 10^{-1} , las técnicas CDMA y SCMA requieren de un SNR de 2,13 y 7,21 dB respectivamente, lo cual representa una diferencia de 5,08 dB entre las mismas. Por otro lado, para un BER de 10^{-3} , la técnica CDMA requirió un SNR de 9,68 dB, mientras que la técnica OFDMA requirió de 6,93 dB para el mismo nivel de BER, por último, la técnica SCMA requirió de 15,35 dB. En cuanto a la técnica OPDMA, se observó que la curva BER no decrece drásticamente conforme aumenta el SNR, alcanzando un nivel de BER de $1,76 \times 10^{-2}$ con un SNR de 20 dB.

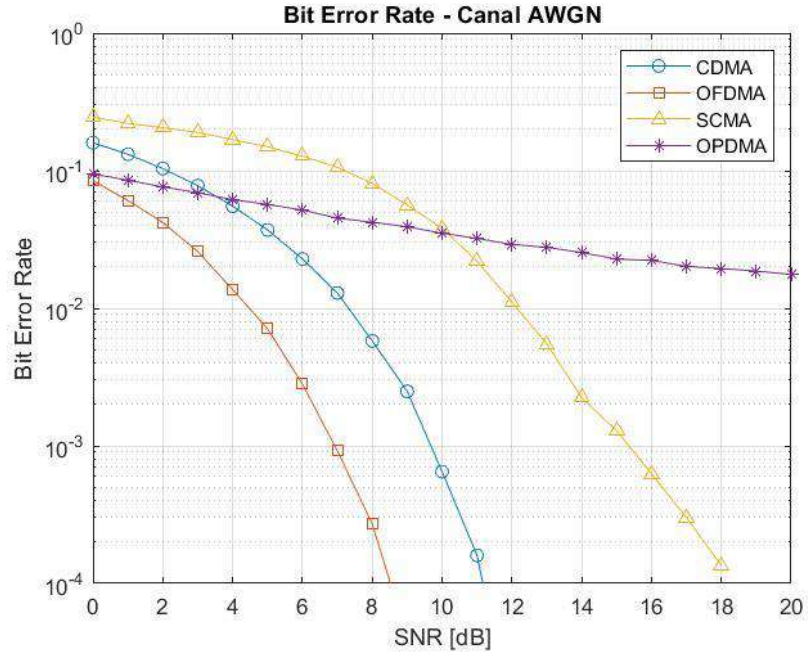


Gráfico 33-3: Curvas BER en función del SNR de las técnicas OMA y NOMA para un canal AWGN.

Realizado por: Armas, Roger, 2022.

En el gráfico 34-3 se muestra las curvas BER de todas las técnicas en un escenario de simulación con canal de desvanecimiento Rice.

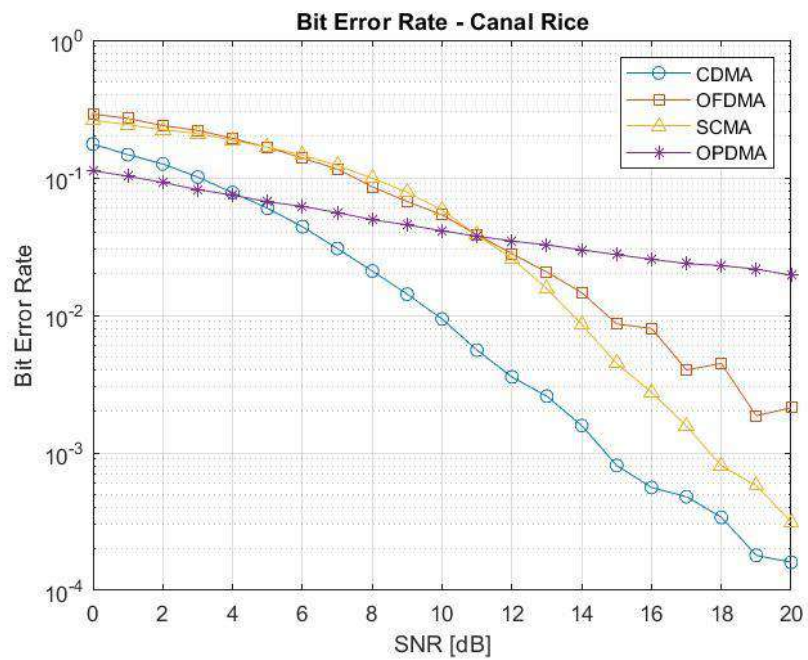


Gráfico 34-3: Curvas BER en función del SNR de las técnicas OMA y NOMA para un canal Rice.

Realizado por: Armas, Roger, 2022.

Analizando los resultados obtenidos, se observa que existe una ventaja de OPDMA respecto a las demás técnicas para un nivel de BER con probabilidad de 10^{-1} , donde el requerimiento de SNR para CDMA fue de 3,04 dB, mientras que para OFDMA se requirió de 7,49 dB. Por último, para las técnicas SCMA y OPDMA, se requirió un SNR de 7,96 y 1,23 dB respectivamente, para alcanzar la misma tasa de errores de bits. No obstante, para probabilidades BER más bajas, la técnica CDMA requirió de valores de SNR más bajos respecto a las demás técnicas. En el caso de una tasa de 10^{-3} , esta técnica logró alcanzar el nivel de BER con un SNR de 14,68 dB, seguido de la técnica SCMA cuyo BER fue alcanzado con un SNR de 17,68 dB. Por último, las técnicas OFDMA y OPDMA alcanzaron una probabilidad de $2,13 \times 10^{-3}$ y $1,95 \times 10^{-2}$ respectivamente, para el máximo valor de SNR equivalente a 20 dB.

Finalmente, en el gráfico 35-5 se puede visualizar las curvas BER de las técnicas correspondiente a un escenario de simulación con canal de desvanecimiento Rayleigh.

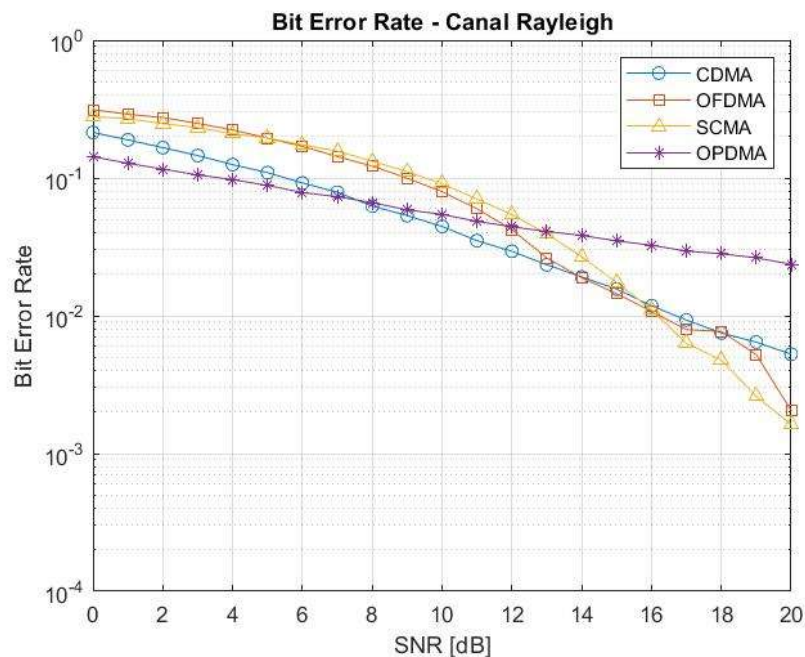


Gráfico 35-3: Curvas BER en función del SNR de las técnicas OMA y NOMA para un canal Rayleigh.

Realizado por: Armas, Roger, 2022.

Respecto al canal Rayleigh, se evidenció una ventaja de la técnica OPDMA para un BER de probabilidad 10^{-1} donde esta requirió un SNR de 3,66 dB, a diferencia de las demás técnicas CDMA, OFDMA y SCMA, cuyos requerimientos fueron de 5,52, 8,98 y 9,53 dB respectivamente. No obstante, para una tasa BER de 10^{-2} , se registró un menor requerimiento de SNR en la técnica SCMA con un valor de 16,14 dB, por otro lado, la técnica CDMA requirió de 16,7 dB de SNR y la técnica OFDMA de 16,24 dB, lo cual evidencia una diferencia de la técnica

SCMA de 0,64 dB con CDMA y de 0,1 dB con la técnica OFDMA. Por último, la técnica OPDMA alcanza una probabilidad de $2,36 \times 10^{-2}$ con un SNR de 20 dB.

3.3. Implementación del simulador

La implementación del simulador de las técnicas de Acceso Múltiple Ortogonales OMA y No ortogonales NOMA se llevó a cabo en el Laboratorio de comunicaciones de la Facultad de Informática y Electrónica, donde se utilizó las computadoras a disposición de los estudiantes de las carreras afines. Para realizar la instalación del simulador se creó un ejecutable Standalone de la aplicación mediante el compilador de Matlab con el objetivo de acceder a la herramienta desde cualquier computadora, sin el requerimiento de tener el software previamente instalado. Cuando el proceso de compilación y empaquetamiento del simulador mediante esta opción termina, se crearon tres carpetas las cuales contienen instaladores para diferentes propósitos como pruebas de testeo, distribución de únicamente los archivos y la distribución de toda la aplicación. Esta última es la que contiene el instalador del simulador y archivos necesarios para el proceso de instalación. El computador utilizado para la implementación del simulador tiene como características: sistema operativo Windows 10, procesador Intel I7-2600 y memoria RAM de 4 GB. En la figura 1-3 se puede visualizar la barra de estado de la instalación del simulador.

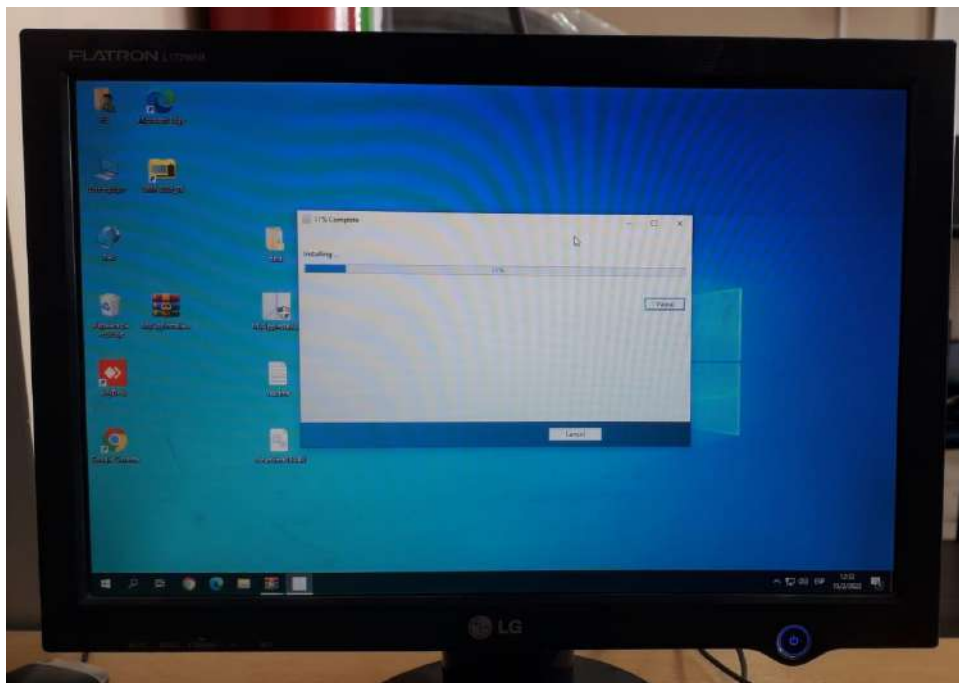


Figura 1-3: Proceso de instalación del simulador en el Laboratorio de Comunicaciones.

Realizado por: Armas, Roger, 2022.

Luego de culminar el proceso de instalación, se creó automáticamente el ícono de aplicación en la pantalla de inicio del computador y se pudo observar la primera ventana con los datos informativos de este trabajo como se puede observar en la figura 2-3.

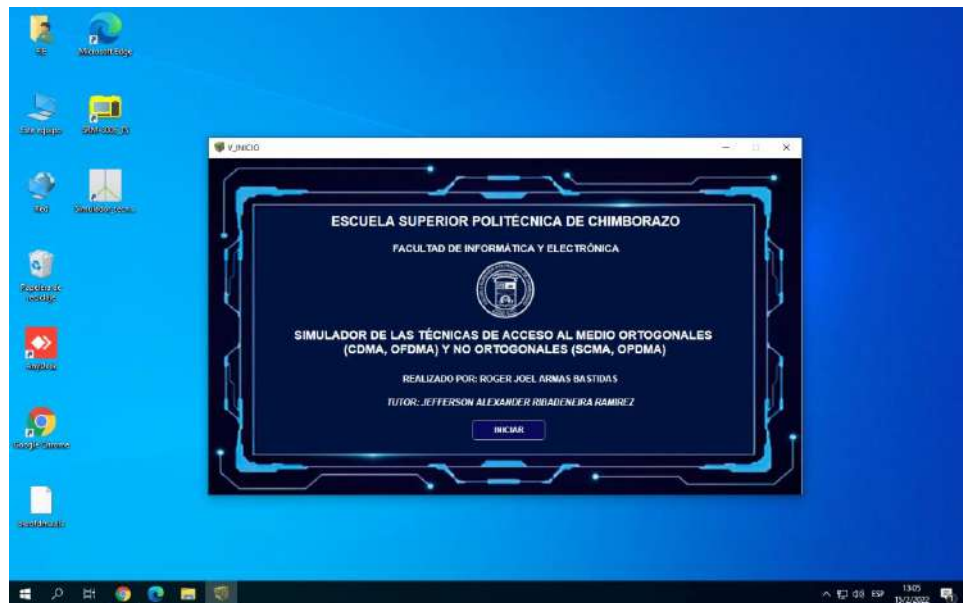


Figura 2-3: Ventana inicial del simulador.

Realizado por: Armas, Roger, 2022.

Luego de dar clic en Iniciar, se desplegó la siguiente ventana denominada V_principal la cual contiene los parámetros de entrada para la simulación de las técnicas. En la parte superior cuenta con pestañas con nombres de las técnicas simuladas con el fin de separar e identificar las técnicas OMA y NOMA. En el anexo C se encuentran los pasos para la instalación correcta del simulador en cualquier computados y en el anexo D se puede visualizar el manual de usuario para la simulación de cada una de las técnicas.

CONCLUSIONES

- Con la investigación y análisis de las Técnicas de Acceso Múltiple Ortogonales y No Ortogonales planteadas se identificó los procesos fundamentales en el funcionamiento de estas, así como los parámetros necesarios en cada subproceso de las etapas de transmisión y recepción. Respecto a lo último, se llega a la conclusión que las técnicas NOMA presentan métodos de recepción más complejos como el Algoritmo de Paso de mensajes MPA en SCMA y el Cancelador de Interferencias sucesivas respectivamente en OPDMA debido a los cálculos repetitivos que se llevan a cabo en cada uno de estos métodos.
- Se diseñó la interfaz gráfica y se programó el código para cada técnica OMA y NOMA en el software matemático MATLAB. Los subprocesos en la etapa de recepción de la técnica SCMA provocaron un tiempo de reacción del simulador mayor a 30 minutos con un computador de 12 GB de memoria RAM y procesador de 4 núcleos.
- En la curva BER de CDMA para los canales AWGN y Rayleigh se observó una gran similitud en las curvas obtenidas en el simulador con las curvas presentadas en el artículo científico de referencia para modulaciones BPSK y QPSK donde se denota una diferencia de 3 dB Eb/No. En el canal Rice con modulación BPSK, para un BER de 10^{-1} se requiere de un SNR de 1,9 dB, obteniéndose una diferencia de 1 dB respecto a la curva de referencia.
- En el caso de OFDMA con canal AWGN y modulación BPSK, se determinó una diferencia de 3 dB en el requerimiento de SNR entre lo obtenido del simulador y la curva de referencia, adicionalmente se registró una diferencia que oscila en el intervalo de 2 a 3 dB para 16-QAM y de 3 a 4 dB para 64-QAM. En los canales Rice y Rayleigh, la diferencia asciende de 4,2 a 13 dB para un mismo nivel de BER y modulación BPSK, debido a la falta de características del canal de desvanecimiento en los resultados presentados por el artículo de referencia.
- En SCMA se denotó una similitud aceptable entre las curvas donde se obtuvo una diferencia en el requerimiento de SNR para un BER de 10^{-1} de 2,05 dB en canal AWGN, y 2,9 dB en canal Rayleigh debido a los libros de códigos aplicados, los cuales pueden diferir de los utilizados en el artículo de referencia.
- Mediante el algoritmo modificado para el cálculo de segmentos de potencia, en OPDMA con modulación QPSK y canal AWGN se observó que para un BER de 5×10^{-2} , la señal del simulador requirió un SNR para un usuario de 1,5 dB, mientras que la curva de referencia establece que es necesario un SNR de 3,76 dB, lo cual marca una diferencia de 2,26 dB. Para

un segundo usuario, el BER de $1,5 \times 10^{-1}$ fue alcanzado con 1,39 dB de SNR evidenciando una diferencia de 1,69 dB respecto a la curva de referencia, debido al efecto de acumulación de errores provocado por la técnica SIC.

- Se realizó simulaciones de todas las técnicas en un escenario de máximo 6 usuarios con el fin de comparar su rendimiento donde se logró determinar que la técnica OFDMA tiene un mejor rendimiento en el canal AWGN y Rice para una tasa BER de 10^{-3} evidenciando una diferencia de 2,75 dB respecto a CDMA y 8,42 respecto a SCMA. En el canal Rayleigh, para una tasa BER de 10^{-2} , se determinó que la técnica SCMA requiere un menor valor de SNR para alcanzar la tasa BER, evidenciando una diferencia de la técnica SCMA de 0,64 dB con CDMA y de 0,1 dB con la técnica OFDMA.
- Se implementó el simulador en el Laboratorio de comunicaciones de la FIE, donde inicialmente se produjo un error por la falta de una librería para lo cual resultó indispensable generar un manual de instalación para los usuarios. Se obtuvieron largos tiempos de reacción del simulador, por lo tanto, para un funcionamiento eficiente del simulador es importante considerar la utilización de un computador de mediana o alta gama con un procesador de mínimo 4 núcleos y una memoria RAM de al menos 16 GB.

RECOMENDACIONES

- Para comprender el principio de funcionamiento de las técnicas CDMA y OFDMA, es recomendable relacionar los procesos de cada uno de ellos con una tecnología de comunicación como UMTS el cual emplea una variación de CDMA y LTE, mismo que emplea OFDMA para el enlace descendente. De la misma manera con las técnicas SCMA y OPDMA las cuales se plantean como técnicas para 5G NR, esto proporciona una base de especificaciones respecto a los requerimientos y parámetros que son necesarios en el principio de funcionamiento de las técnicas.
- No se implementó un proceso de codificación para la corrección de errores con el fin de poder observar solo el rendimiento de las técnicas de acceso múltiple ortogonal y no ortogonal, sin embargo, se recomienda implementar diferentes métodos de codificación en las técnicas NOMA para la obtención de una mejora en la tasa de errores de bit BER y realizar una comparación con las técnicas OMA.
- La técnica SCMA tiene una limitación en cuanto a los números de usuarios que se puede establecer en el escenario de simulación, esto se debe a que se requiere el diseño de libros de códigos adicionales, los cuales pueden obtenerse mediante varias metodologías mencionadas en este trabajo, por lo que se recomienda el diseño de nuevos libros de códigos para aumentar más opciones en los parámetros de entrada y generar un simulador más robusto.
- Para reducir tiempos de espera de resultados extensos, se puede modificar el código de programación con funciones tales que permitan utilizar todos los núcleos del procesador. Aunque se reduce la complejidad del código en cierta medida, se recomienda cambiar el código programación para que los cálculos se realicen sobre una unidad de procesamiento gráfico GPU.
- Aunque el simulador proporcione gráficas que puedan ser utilizadas como referencia para evaluar el rendimiento de las técnicas, también puede servir como base para futuras implementaciones en entornos reales mediante la transmisión y recepción de las señales generadas en el simulador con dispositivos USRP, a través de los cuales se podría tener una comparación entre el BER obtenido con la simulación y lo obtenido con la implementación del sistema inalámbrico.

BIBLIOGRAFÍA

3GPP Release 4. Overview of 3GPP Release 4, Summary of all Release 4 Features.

AAMER, Syed, et al. “A review of codebook design methods for sparse code multiple access”. *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, nro . 2, 2021, pp. 319-327.

ANWAR, Asim, et al. “A Survey on Application of Non-Orthogonal Multiple Access to Different Wireless Networks”. *Electronics*, vol. 8, nro. 11 , 2019, pp. 1-46.

ARRIETA, Cristina. Simulación de un sistema de comunicación inalámbrica usando la técnica de acceso NOMA (Non-Orthogonal Multiple Access) [En Línea]. (Trabajo de titulación). (Pregrado). Escuela Politécnica Nacional, Quito. 2020. [Consulta: 2021-11-02] Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/20944/1/CD%2010466.pdf>

BAYESTEH, Alireza, et al. “Low Complexity Techniques for SCMA Detection”, *2015 IEEE Globecom Workshops (GC Wkshps)*, 2015, pp. 1-6.

CAI, Donghong, et al. “Multi-Dimensional SCMA Codebook Design Based on Constellation Rotation and Interleaving”. *IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1-5.

CHEN, Yan, et al. “Toward the Standardization of Non-Orthogonal Multiple Access for Next Generation Wireless Networks”. *IEEE Communications Magazine*, vol. 56, nro. 3, 2018, pp. 19-27.

COX, Chistopher. *An Introduction to LTE: LTE, LTE-Advanced, SAE, VoLTE and 4G Mobile Communications, Segunda ed.* s.l. : John Wiley & Son, 2014.

DAI, Linglong, et al. “Non-Orthogonal Multiple Access for 5G: Solutions, Challenges, Opportunities, and Future Research Trends”. *IEEE Communications Magazine*, vol. 53, nro. 9, 2015, pp. 74-81.

ESCUADERO, José y CÉSPEDES, Máximo. *Orthogonal Multiple Access.* Madrid : s.n., 2021, Wiley 5G ref, pp. 1-24.

HAN, Weijia, et al. “Orthogonal Power Division Multiple Access: A Green Communication Perspective”. *IEEE Journal on Selected Areas in Communications*, vol. 34, nro. 12, 2016, pp. 3828-3842.

HAN, Weijia y MA, Xiao. “Power Division Multiplexing”. *IEEE/CIC International Conference on Communications in China (ICCC)*, 2016, pp. 1-6.

HAN, Weijia, et al. “Study of the Symbol Error Rate/Bit Error Rate in Non-orthogonal Multiple Access (NOMA) Systems”. *IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, 2016, pp. 101-106.

HAN, Weijia, et al. “Study of SER and BER in NOMA Systems”. *IEEE Transactions on Vehicular Technology*, vol. 70, nro. 4, 2021, pp. 3325-3340.

HUIDROBO, José Manuel. *Comunicaciones móviles: sistemas GSM, UMTS y LTE*. Madrid : RA-MA Editorial, 2014. RA-MA Editorial.

INSTITUTO NACIONAL DE ESTADÍSTICA Y CENSOS. *Indicadores de tecnología de la información y comunicación. Ecuador en cifras*. [En línea] Abril de 2021. [Consultado: 2021-11-05] Disponible en: https://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2020/202012_Boletin_Multiproposito_Tics.pdf.

ISLAM, Riazul S.M., et al. “Power-Domain Non-Orthogonal Multiple Access (NOMA) in 5G Systems: Potentials and Challenges”. *IEEE Communications Surveys & Tutorials*, vol. 19, nro. 2, 2017, pp. 721-742.

JAIN, Gunjita y DUBEY, Rupesh. “BER Performance of OFDM System over Nakagami-m Fading Channels with Different Modulation Schemes”. *International Journal of Advanced Research in Computer Engineering & Technology*, Vol. nro. 1, 6, 2012, pp. 111-116.

LI, Li Erran, et al. “Superposition coding for wireless mesh networks”. *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, 2007, pp. 330–333.

LIU, Jiaqi, et al. “On Fixed-Point Implementation of Log-MPA for SCMA Signals”. *IEEE Wireless Communications Letters*, Vol. 5, nro. 3, 2016, pp. 324-327.

LIU, Yuanwei. *Non-Orthogonal Multiple Access for 5G: Design and Performance Enhancement*. Londres : Queen Mary University of London, 2016.

LOU, Shun, et al. “SCMA with Low Complexity Symmetric Codebook Design for Visible Light Communication”. *IEEE International Conference on Communications (ICC)*, 2018, pp. 1-6.

MAHAJAN, Mahadev y BHOYAR, D. B. “Performance analysis of CDMA under Rayleigh Rician and Nakagami fading channels”. *Thinkquest~2010*, 2011, pp. 316-319.

MAKKI, Behrooz, et al. “A Survey of NOMA: Current Status and Open Research Challenges”. *IEEE Open Journal of the Communications Society*, vol. 1, 2020, pp. 179-189.

MCELIECE, Robert. “The Theory of m-Sequences”. *Springer, Finite Fields for Computer Scientists and Engineers*, vol. 23, 1987, Boston, pp. 151-167.

MIN, Baocheng y SUN, Jun. “Message Passing Algorithm with Dynamic Thresholds in SCMA”. *International Conference on Communications, Information System and Computer Engineering (CISCE)*, 2019, pp. 324-327.

MITRA, Rangeet, et al. “Color-domain SCMA NOMA for Visible Light Communication”. *IEEE Communications Letters*, vol. 25, nro. 1, 2020, pp. 200-204.

NSAIF Sukar, MURTADHA Ali y PAL, Maninder. SC-FDMA & OFDMA in LTE physical layer. *International Journal of Engineering Trends and Technology*, vol. 12, nro. 2, 2014, pp. 74-85.

PHASOULIOTIS, Antonis y K. C. SO, Daniel. “Performance Analysis and Comparison of Downlink MIMO MC-CDMA and MIMO OFDMA Systems”. *VTC Spring 2009-IEEE 69th Vehicular Technology Conference*, 2009, pp. 1-6.

PINTO, Frank, et al. “Software Defined Radio implementation of DS-SS-CDMA in inter-satellite communications for small satellites”. *IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 2015, pp. 1-6.

PROAKIS, Jhon y SALEHI, Masoud. *Digital Communications*. Quinta ed. New York : McGraw-Hill, 2007.

RODRÍGUEZ, Andrés y PÉREZ, Alipio. “Métodos científicos de indagación y de construcción del conocimiento”. *Revista EAN*, Vols. s.n., 82, Bogotá, 2017, pp. 179-200.

TAHERZADEH, Mahmoud, et al. “SCMA Codebook Design”. *IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1-5.

WANG, Bichai, et al. "Comparison Study of Non-Orthogonal Multiple Access Schemes for 5G". *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2015, pp. 1-5.

WANG, Peng, XIAO, Jun y PING, Li. "Comparison of orthogonal and non-orthogonal approaches to future wireless cellular systems". *IEEE Vehicular Technology Magazine*, vol. 1, nro. 3, 2006, pp. 4-11.

WANG, Qi, et al. "Non-Orthogonal Multiple Access: A Unified Perspective". *IEEE Wireless Communications*, vol. 25, nro. 2, 2018, pp. 10-16.

YIN, Hujun y ALAMOUTI, Siavash. "OFDMA: A Broadband Wireless Access Technology". *IEEE Sarnoff Symposium*, 2006, pp. 1-4.

YU, Lisu, et al. "Design and Analysis of SCMA Codebook Based on Star-QAM Signaling Constellations". *IEEE Transactions on Vehicular Technology*, vol. 20, nro. 20, 2018, pp. 1-11.

ZHANG, Xuewan, et al. "A Lattice-Based SCMA Codebook Design for IoMT Communications". *IEEE/CIC International Conference on Communications Workshops*, 2019, pp. 169-173.

ZHONG, Chao, et al. "A Novel SCMA Codebook Construction Based on Extended Factor Graph Design". *IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1-6.

ANEXOS

ANEXO A: Código de programación de las simulaciones.

- CDMA

```
%% Generador de las señales de usuarios
NU = 6; % Número de usuarios
Nbits = 100; % Cantidad de bits por usuario
data = randi([0 1], Nbits, NU); % Generador de datos aleatorios
Canal = 'AWGN'; % 1. AWGN, 2. Rician, 3. Rayleigh

%% Ensanchamiento

dataE=zeros(Nbits*max(SF),NU);

for u = 1:NU
    for i = 0:length(data)-1
        if data(i+1,u) == 1
            dataE(i*SF(u)+1:SF(u)*(i+1),u) = 1; % Ensanchamiento si el
bit es 1
        elseif data(i+1,u) == 0
            dataE(i*SF(u)+1:SF(u)*(i+1),u) = 0; % Ensanchamiento si el
bit es
        end
    end
end

%% Generador del código PN
ind = 1:18:2^25;
spreadseq=zeros(max(SF),NU);
for u = 1:NU
    PNseq = comm.GoldSequence('FirstPolynomial','x^25+x^3+1',...
        'FirstInitialConditions',[1 1 0 0 zeros(1, 20) 1], ...
        'SecondPolynomial','x^25+x^3+x^2+x+1',...
        'SecondInitialConditions',[1 1 0 0 zeros(1, 20) 1], ...
        'Index',ind(u),'SamplesPerFrame',SF(u));
    spreadseq(1:SF(u),u) = PNseq();
end

%% XOR
dataspread=zeros(length(dataE),NU);
for u = 1:NU
    for i=0:length(data)-1
        dataspread(i*SF(u)+1:SF(u)*(i+1),u) =
double(xor(dataE(i*SF(u)+1:SF(u)*(i+1),u),spreadseq(1:SF(u),u)));
    end
end

%% SUMA DE SEÑALES
```



```

for i=0: length(dataspread)-1
    for u = 1:NU
        SS(i*NU+u,1)=dataspread(i+1,u);
    end
end

%% Modulación
Mod = 2; % Esquema de modulación (1. BPSK, 2. QPSK, 3. 16-QAM, 4. 64-
QAM
switch Mod
    case 1
        bpskModulator = comm.BPSKModulator;
        bpskModulator.PhaseOffset = pi/2;
        SSmod = bpskModulator(SS);
    case 2
        QpskModulator = comm.QPSKModulator;
        QpskModulator.PhaseOffset = pi/4;
        QpskModulator.BitInput = 1;
        SSmod = QpskModulator(SS);
    case 3
        SSmod = qammod(SS,16,'InputType','bit');
    case 4
        SSmod = qammod(SS,64,'InputType','bit');
end
s_tx = SSmod;

for i=0:1:30
    i = 5;
    snr=10^(i/10);
    %% CANAL INALÁMBRICO
    switch Canal
        case 'AWGN' % Canal AWGN
            h = 1;
            sigcanal = awgn(s_tx,i,'measured');
        case 'Rice' % Canal Rice
            k=1; % Factor Rician
            media=sqrt(k/(k+1)); % Media
            sigma=sqrt(1/(2*(k+1))); % Varianza
            Nr=randn(1,length(s_tx))*sigma+media; %Genera valores
aleatorios para rician
            Ni=randn(1,length(s_tx))*sigma;
            h=sqrt(Nr.^2+Ni.^2); % Coeficientes de desvanecimiento Rician
            Np=1/snr; % Potencia de ruido
            sd=sqrt(Np/2);% Desviación estandar para ruido gaussiano.

n=random('Normal',0,sd,1,length(s_tx))+1i*random('Normal',0,sd,1,length
h(s_tx)); %Generates complex Gaussian noise
            sigcanal = s_tx.*h' + n';
        case 'Rayleigh' % Canal Rayleigh

Nc=random('normal',0,1/(sqrt(2)),1,length(s_tx))+1i*random('normal',0,
1/(sqrt(2)),1,length(s_tx)); % Ruido complejo
            Nr=real(Nc); % Parte real del Ruido
            Ni=imag(Nc); % Parte imaginaria del ruido
            h=sqrt(Nr.^2+Ni.^2); % rayleigh fading coefficient
            Np=1/snr; % Potencia de ruido

```

```

sd=sqrt(Np/2);% Desviación estandar para ruido gaussiano.

n=random('Normal',0,sd,1,length(s_tx))+1i*random('Normal',0,sd,1,length(s_tx)); %Generates complex Gaussian noise
sigcanal = s_tx.*h' + n';
end
%% RECEPTOR
% Ecuación
s_rx = sigcanal./h';
% DEMODULACIÓN
switch Mod
    case 1
        bpskDemodulator = comm.BPSKDemodulator; % Objeto para demodulador BPSK
        bpskDemodulator.PhaseOffset = pi/2; %Desfase de 90 grados
        SSdemod = bpskDemodulator(s_rx); % Aplica el demodulador en la señal del canal
    case 2
        QpskDemodulator = comm.QPSKDemodulator('BitOutput', true,'PhaseOffset', pi/4, 'SymbolMapping', 'Gray');% Objeto para demodulador QPSK
        SSdemod = QpskDemodulator(s_rx); % Aplica el demodulador en la señal del canal
    case 3
        SSdemod = qamdemod(s_rx,16,'OutputType','bit'); % Demodula la señal con salida tipo bit
    case 4
        SSdemod = qamdemod(s_rx,64,'OutputType','bit'); % Demodula la señal con salida tipo bit
end

%% Separación de las señales
dataspreadrx = zeros(length(dataspread),NU);
for j =0: length(dataspread)-1
    for u = 1:NU % Se repite para cada usuario
        dataspreadrx(j+1,u)=SSdemod(j*NU+u,1); % Se separa cada bit de la señal demodulada en un vector para cada usuario
    end
end

%% Reconocimiento de los datos con PN code
datadespread=zeros(length(dataE),NU);
for u = 1:NU % Se repite para cada usuario
    for j=0:length(data)-1
        datadespread(j*SF(u)+1:SF(u)*(j+1),u) = double(xor(dataspreadrx(j*SF(u)+1:SF(u)*(j+1),u),spreadseq(1:SF(u),u)));
    end
end

%% Desensanchamiento
datarx=zeros(Nbits,NU);
for u = 1:NU % Se repite para cada usuario
    for j = 0:length(data)-1
        if datadespread(j*SF(u)+1,u) == ones(1,SF(u)) % Reconoce a un intervalo de bits como 1
            datarx(j+1,u) = 1; % Obtención del bit 1 desensanchado
        end
    end
end

```

```

        elseif datadespread(j*SF(u)+1,u) == zeros(1,SF(u)) % Reconoce a un
intervalo de bits como 1
            datarx(j+1,u) = 0; % Obtención del bit 0 desensanchado
        end
    end
end
end
errorRate = comm.ErrorRate;
for u = 1:NU
    errorStats = errorRate(data(:,u),datarx(:,u));
    ber(i+1,u) = errorStats(1);
    numerr(i+1,u) = errorStats(2);
end
end
end

```

- OFDMA

```

% Generador de bits aleatorios
NU = 6; % Número de usuarios
Nbits = 100000; % Número de bits
data = randi([0 1], Nbits, NU);
Canal = 3; % 1. AWGN, 2. Rice, 3. Rayleigh
t_fft=512; % Tamaño de la FFT
switch t_fft
    case 128
        subp = 72;
        Nref = 120;
    case 256
        subp = 180;
        Nref = 300;
    case 512
        subp = 300;
        Nref = 500;
    case 1024
        subp = 600;
        Nref = 1000;
    case 2048
        subp = 900;
        Nref = 2000;
end
%% Modulación
Mod = 2; % Esquema de modulación (1. BPSK, 2. QPSK, 3. 16-QAM, 4. 64-
QAM)
switch Mod
    case 1
        for u = 1:NU
            bpskModulator = comm.BPSKModulator;
            bpskModulator.PhaseOffset = pi/2;
            Smod(:,u) = bpskModulator(data(:,u));
        end
    case 2
        for u = 1:NU
            QpskModulator = comm.QPSKModulator;
            QpskModulator.BitInput = 1;
            Smod(:,u) = QpskModulator(data(:,u));
        end
    case 3
        for u = 1:NU

```



```

                                k = k+1;
                                else
                                    break
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
end

%% Ifft

for i=1:numel(trama)/subp
    s_ifft(i,:)=ifft(trama(:,i),t_fft); % Calculo de la Ifft cada 140
portadoras
end
s_ifft=sqrt(subp)*s_ifft; % Normalización de la IFFT

%% Prefijo ciclico
t_pc = 1/8;
pc = ceil(t_fft*t_pc);
for l=1:1:((numel(trama))/subp) %Agrega CP de 40 unidades de tiempo a
cada slot
    s_cp(l,:)=[s_ifft(l,t_fft-(pc-1):t_fft) s_ifft(l,1:t_fft)];
end
[n1,n2] = size(s_cp);

%% Convertidor de paralelo a serie
for i=0:1:((numel(trama))/subp)-1
    s_ps(i*n2+1:n2*(i+1))= s_cp(i+1,:);
end

s_tx = s_ps;

a=1;
for snr = 0:1:20
    snr2 = 10^(snr/10);
    %% CANAL INALAMBRICO
    switch Canal
        case 1 % Canal AWGN
            h = 1;
            S_canal = awgn(s_tx,snr,'measured');
        case 2 % Canal Rice
            k=2; % Factor Rician
            mean=sqrt(k/(k+1)); % Media
            sigma=sqrt(1/(2*(k+1))); % Varianza
            Nr=randn(1,length(s_tx))*sigma+mean; %Genera valores
aleatorios para ricean
            Ni=randn(1,length(s_tx))*sigma;
            h=sqrt(Nr.^2+Ni.^2); % Coeficientes de desvanecimiento Rician
            Np=1/snr2; % Potencia de ruido
            sd=sqrt(Np/2); % Desviación estandar para ruido gaussiano.

```

```

n=random('Normal',0,sd,1,length(s_tx))+1i*random('Normal',0,sd,1
,length(s_tx)); %Generates complex Gaussian noise
S_canal1 = s_tx.*h;
S_canal = awgn(S_canal1,snr,'measured');
case 3 % Canal Rayleigh
Nc=random('normal',0,1/(sqrt(2)),1,length(s_tx))+1i*random('norm
al',0,1/(sqrt(2)),1,length(s_tx)); % Ruido complejo
Nr=real(Nc); % Parte real del Ruido
Ni=imag(Nc); % Parte imaginaria del ruido
h=sqrt(Nr.^2+Ni.^2); % rayleigh fading coefficient
Np=1/snr2; % Potencia de ruido
sd=sqrt(Np/2);% Desviación estandar para ruido gaussiano.

n=random('Normal',0,sd,1,length(s_tx))+1i*random('Normal',0,sd,1
,length(s_tx)); %Generates complex Gaussian noise
S_canal1 = s_tx.*h;
S_canal = awgn(S_canal1,snr,'measured');
end
%% Receptor
r_rx = S_canal./h;
%% Convertidor de serie a paralelo
for i=0:1:(numel(trama)/subp)-1
r_sp(i+1,:) = r_rx(i*n2+1:n2*(i+1));
end

%% Quitar el prefijo ciclico
%Bloque Remover prefijo cíclico
for l=1:1:(numel(trama)/subp) %Remueve el CP de los simbolos
r_cp(l,:)=r_sp(l,pc+1:n2);
end
%% Tranformada FFT
for i=1:1:(numel(trama)/subp) %Calcula la FFT
r_fft(:,i)=fft(r_cp(i,:));
end
r_fft=roundn(r_fft,-15); %Redondea a 15 decimales
for l=1:1:(numel(trama)/subp)
for i=1:1:subp
tramaR(i,l)=r_fft(i,l);%Adquiere RGrid con 300 subportadoras
end
end
tramaR=(1/sqrt(subp)).*tramaR; % Desnormalización

%% Demapeo
for u = 1:NU % Desplazamiento entre usuarios
k=1;
for I = 1:RB(u) % Desplazamiento entre elementos del vector RB
for i = 1:12:subp % Desplazamiento entre portadoras
if i == (ceil(RBAu(I,u)/(c/7))-1)*12+1 %Localización de la
subportadora
for j = 1:7:c % Desplazamiento entre simbolos
if j == ((RBAu(I,u) - (ceil(RBAu(I,u)/(c/7))-
1)*(20*Num))-1)*7+1 %Localización del slot
for l = 0:1:11 % Desplazamiento entre portadoras

```

```

        for h = 0:1:6 % Desplazamiento entre simbolos
de cada slot
            if k <= N_simb(u)
                d_mod(k,u) = tramaR(i+1,j+h);
%Asignación del simbolo de la trama a cada usuario
                k = k+1;
            else
                break
            end
        end
    end
end
end
end
end
end
end
end
end
end
end

%% Demodulación
switch Mod
    case 1
        for u = 1:NU
            bpskDemodulator = comm.BPSKDemodulator;
            bpskDemodulator.PhaseOffset = pi/2;
            dataR(:,u) = bpskDemodulator(d_mod(:,u));
        end
    case 2
        for u = 1:NU
            QpskDemodulator = comm.QPSKDemodulator( 'BitOutput',
true, 'PhaseOffset', pi/4, 'SymbolMapping', 'Gray');
            dataR(:,u) = QpskDemodulator(d_mod(:,u));
        end
    case 3
        for u = 1:NU
            dataR(:,u) = qamdemod(d_mod(:,u),16, 'OutputType', 'bit');
        end
    case 4
        for u = 1:NU
            dataR(:,u) = qamdemod(d_mod(:,u),64, 'OutputType', 'bit');
        end
end
end
%% Calculo del BER
for u = 1:NU
[nbit(a,u), ber(a,u)] = biterr(data(:,u),dataR(:,u));
end
a=a+1;
end
end

```

- SCMA

```

% Establecemos parámetros principales de simulación

R = 4; % Numero de recursos ortogonales K
CW = 4; % Numero de palabras de código M
U = 6; % Número de usuarios V

%% Generación de simbolos de log2(CW) bits

```

```

Nsimb = 10000;
x = randi([0 CW-1], U, Nsimb);
Canal = 2;
%% En bits
r = de2bi(x, log2(CW), 'left-msb'); % Transformación de decimal a
binario
data = zeros(log2(CW)*length(x), U); % Creación de la matriz data
for kk = 1:U
    data(:,kk) = reshape(reshape(reshape(r, U, kk-1).', [], 1), log2(CW), U); % Reconstruir
el vector r y ponerlo en data
end

for i = 0:20
    snr = 10^(i/10);
    N0 = 1/(10^(i/10)); % Potencia de ruido
    while ((min(Nerr(:,i+1)) < maxNumErrs) && (Nbits(1,i+1) <
maxNumBits))
        %% Generacion de las ganancias de canal
        switch Canal
            case 1 % Canal AWGN
                h = ones(R,U,Nsimb);
            case 2 % Canal Rice
                k=2; % Factor Rician
                mean=sqrt(k/(k+1)); % Media
                sigma=sqrt(1/(2*(k+1))); % Varianza
                Nr=randn(R,U,length(x))*sigma+mean; %Genera valores
aleatorios para rician
                Ni=randn(R,U,length(x))*sigma;
                h=sqrt(Nr.^2+Ni.^2); % Coeficientes de desvanecimiento
Rician
            case 3
                h = 1/sqrt(2)*(randn(R, U, Nsimb)+1j*randn(R, U,
Nsimb)); % Rayleigh channel
        end
        %% MAPEO DE SIMBOLOS A PALABRAS DE CÓDIGO
        pc = zeros(R, length(x));
        for t = 1:length(x)
            for j = 1:U
                pc(:,t) = pc(:,t)+CB(:, x(j,t)+1, j).*h(:,j,t); %Suma
de las palabras de código de cada usuario con la ganancia de canal
            end
        end
        s_fade = pc;
        switch Canal
            case 1
                s_canal = awgn(s_fade, i, 'measured');
            case 2
                Np=1/snr; % Potencia de ruido
                sd=sqrt(Np/2); % Desviación estandar para ruido
gaussiano.

                n=random('Normal',0,sd,R,length(x))+1i*random('Norma
1',0,sd,R,length(x)); %Generates complex Gaussian
noise
                s_canal = awgn(s_fade, i, 'measured');
            case 3
                Np=1/snr; % Potencia de ruido

```



```

sd=sqrt(Np/2);% Desviación estandar para ruido
gaussiano.

n=random('Normal',0,sd,R,length(x))+1i*random('Normal',0,sd,R,length(x)); %Generates complex Gaussian noise
s_canal = awgn(s_fade, i, 'measured');
end
LLR = scmadec(s_canal, CB, h, NO, Nit); % Decodificación de la señal mediante MPA
% Conversión de LLR a bit
datadec = reshape((LLR <= 0), [log2(CW) length(x)*U]).'; % Decodificación de los datos
datar = zeros(log2(CW)*length(x), U); % Matriz de datos recuperados
for kk = 1:U
    datar(:,kk) = reshape(downsample(datadec, U, kk-1).', [], 1); % Reconstruir el vector datadec y ponerlo en datar
end
err = sum(xor(data, datar)); % Detección de errores
Nerr(:,i+1) = Nerr(:,i+1) + err.'; % Obtención del número de bits errados
Nbits(:,i+1) = Nbits(:,i+1) + log2(CW)*length(x); % Suma del total de bits
end
BER(:,i+1) = Nerr(:,i+1)./Nbits(:,i+1); % Calculo del BER
end

```

- OPDMA

```

%% Etapa de transmisión
% Generador de N bits de datos para M usuarios
M = 6; % Numero de usuarios
Nbits = 100000; % Cantidad de bits
x = randi([0 1], Nbits, M); % Vector de bits aleatorios
Canal = 3; % 1. AWGN, 2. Rice, 3. Rayleigh
%% Modulación
Mod = 2; % Esquema de modulación (1. BPSK, 2. QPSK, 3. 4-QAM)
switch Mod
    case 1
        for u = 1:M
            BPSKmod = comm.BPSKModulator;% Objeto para modulador BPSK
            Smod(:,u) = BPSKmod(x(:,u)); % Aplicación de la modulación
        end
    case 2
        for u = 1:M
            QPSKmod = comm.QPSKModulator('BitInput',true); % Objeto para modulador QPSK
            Smod(:,u) = QPSKmod(x(:,u));
        end
    case 3
        for u = 1:M
            Smod(:,u) = qammod(x(:,u),4,'InputType','bit'); % Aplicación de la modulación
        end
end
end

```

```

%% Estrategia de división de potencia PS
if M == 1
    fbeta = 1; % Factor beta entre 0 a 1
    h = 1; % Ganancia de canal, para canal awgn h = 1
elseif M == 2
    fbeta = [0.8 0.2]; % Factor beta entre 0 a 1
    h = [ 1 1 ]; % Ganancia de canal, para canal awgn h = 1
elseif M == 3
    fbeta = [0.8 0.15 0.05]; % Factor beta entre 0 a 1
    h = [ 1 1 1 ]; % Ganancia de canal, para canal awgn h = 1
elseif M == 4
    fbeta = [0.7 0.15 0.1 0.05]; % Factor beta entre 0 a 1
    h = [ 1 1 1 1 ]; % Ganancia de canal, para canal awgn h = 1
elseif M == 5
    fbeta = [0.5 0.2 0.15 0.1 0.05]; % Factor beta entre 0 a 1
    h = [ 1 1 1 1 1 ]; % Ganancia de canal, para canal awgn h = 1
elseif M == 6
    fbeta = [0.4 0.25 0.15 0.1 0.05 0.05];
    h = [ 1 1 1 1 1 1 ]; % Ganancia de canal, para canal awgn h = 1
end
h = sort(h, 'descend'); % Ordenar de mayor a menor
h_m=h.*fbeta;
% Paso 2
S_max = 46;% Potencia máxima en dBm
s_max = (10^-3)*db2pow(S_max); % Potencia máxima lineal
sum_p = 1 ; % Suma de cada segmento de potencia para control
% Parámetros de transmisión de usuarios
% Ancho de banda del canal dependiendo la aplicación
B = 10e6;
% Tasas de transmisión requeridas para los usuarios
c=[480.747e6 405.643e6 375.457e6 349.362e6 319.7e6 300.11e6];
N = - 174 + 10*log10(B); %Potencia de ruido en escala logaritmica
n = (10^-3)*db2pow(N); %Potencia de ruido en escala lineal
% Calculo de los segmentos de potencia PS
s_i = 0;
for m =1:1:M
    s(m) = ((2.^(c(m)/B))-1)*((n/h_m(m))*sum_p); % Calculo del
segmento de potencia
    sum_p = s_i + s(m); % Resta del PS al total de potencia para
control
    s_i = sum_p;
    if sum_p > s_max % Control de los segmentos de potencia
        s(:) = 0;
        break
    end
end
end

%% Asignación de potencia para cada usuario y suma de los mismos
% Superposición de códigos
for m = 1:1:M
    y(:,m) = sqrt(s(m)).*Smod(:,m); % Multiplicación de la señal con
el segmento de potencia
end
y_f = sum(y,2); % Suma de señales / Superposición de código
%% Canal inalámbrico
SNR = 0:44;
snr = db2pow(SNR);

```

```

for i = 1:length(SNR)
    switch Canal
    case 1 % Canal AWGN
        h = 1;
        s_canal1 = awgn(y_f,snr(i),sqrt(39.8107),'linear');
    case 2 % Canal Rice
        k=4; % Factor Rician
        media=sqrt(k/(k+1)); % Media
        sigma=sqrt(1/(2*(k+1))); % Varianza
        Nr=randn(1,length(y_f))*sigma+media; %Genera valores
aleatorios para ricean
        Ni=randn(1,length(y_f))*sigma;
        h=sqrt(Nr.^2+Ni.^2); % Coeficientes de desvanecimiento
Rician
        Np=1/snr(i); % Potencia de ruido
        sd=sqrt(Np/2); % Desviación estandar para ruido gaussiano.

        n=random('Normal',0,sd,1,length(y_f))+1i*random('Normal',0
        ,sd,1,length(y_f)); %Generates complex Gaussian noise
        s_fade = y_f.*h';
        s_canal1 = awgn(s_fade,snr(i),sqrt(39.8107),'linear');
    case 3 % Canal Rayleigh

        Nc=random('normal',0,1/(sqrt(2)),1,length(y_f))+1i*random(
        'normal',0,1/(sqrt(2)),1,length(y_f)); % Ruido complejo
        Nr=real(Nc); % Parte real del Ruido
        Ni=imag(Nc); % Parte imaginaria del ruido
        h=sqrt(Nr.^2+Ni.^2); % rayleigh fading coefficient
        Np=1/snr(i); % Potencia de ruido
        sd=sqrt(Np/2); % Desviación estandar para ruido gaussiano.

        n=random('Normal',0,sd,1,length(y_f))+1i*random('Normal',0
        ,sd,1,length(y_f)); %Generates complex Gaussian noise
        s_fade = y_f.*h';
        s_canal1 = awgn(s_fade,snr(i),sqrt(39.8107),'linear');
    end
    % RECEPTOR SIC en base a la modulación
    s_canal = s_canal1./h';
    switch Mod
    case 1
        BPSKdemod = comm.BPSKDemodulator;
        for j = 1:1:M
            if j == 1
                dec(:,j) = step(BPSKdemod, s_canal); %
Demodulacion directa para obtener datos de U1
            elseif j == 2
                dec12 = step(BPSKdemod, s_canal);
                %Demodulacion directa para obtener datos de U1
                dec12_remod = step(BPSKmod, dec12);
                %Remodulacion de los datos de U1
                rem2 = s_canal - sqrt(s(1))*dec12_remod; %SIC
para remover los datos de U1
                dec(:,j) = step(BPSKdemod, rem2);
                %Demodulación directa para obtener los datos de U2
            elseif j == 3
                dec13 = step(BPSKdemod, s_canal);
                %Demodulacion directa para obtener datos de U1
            end
        end
    end
end

```

```

        dec13_remod = step(BPSKmod, dec13);
%Remodulacion de los datos de U1
        rem31 = s_canal - sqrt(s(1))*dec13_remod;    %SIC
para remover los datos de U1
        dec23 = step(BPSKdemod, rem31);    %Demodulación
directa para obtener los datos de U2
        dec23_remod = step(BPSKmod, dec23);
%Remodulacion de los datos de U2
        rem32 = rem31 - sqrt(s(2))*dec23_remod; %SIC para
remover los datos de U2
        dec(:,j) = step(BPSKdemod, rem32);
%Demodulación directa para obtener los datos de U3
        elseif j == 4
            dec14 = step(BPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
            dec14_remod = step(BPSKmod, dec14);
%Remodulacion de los datos de U1
            rem41 = s_canal - sqrt(s(1))*dec14_remod;    %SIC
para remover los datos de U1
            dec24 = step(BPSKdemod, rem41);    %Demodulación
directa para obtener los datos de U2
            dec24_remod = step(BPSKmod, dec24);
%Remodulacion de los datos de U2
            rem42 = rem41 - sqrt(s(2))*dec24_remod; %SIC para
remover los datos de U2
            dec34 = step(BPSKdemod, rem42);    %Demodulación
directa para obtener los datos de U3
            dec34_remod = step(BPSKmod, dec34);
%Remodulacion de los datos de U3
            rem43 = rem42 - sqrt(s(3))*dec34_remod; %SIC para
remover los datos de U3
            dec(:,j) = step(BPSKdemod, rem43);
%Demodulación directa para obtener los datos de U4
            elseif j == 5
                dec15 = step(BPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
                dec15_remod = step(BPSKmod, dec15);
%Remodulacion de los datos de U1
                rem51 = s_canal - sqrt(s(1))*dec15_remod;    %SIC
para remover los datos de U1
                dec25 = step(BPSKdemod, rem51);    %Demodulación
directa para obtener los datos de U2
                dec25_remod = step(BPSKmod, dec25);
%Remodulacion de los datos de U2
                rem52 = rem51 - sqrt(s(2))*dec25_remod; %SIC para
remover los datos de U2
                dec35 = step(BPSKdemod, rem52);    %Demodulación
directa para obtener los datos de U3
                dec35_remod = step(BPSKmod, dec35);
%Remodulacion de los datos de U3
                rem53 = rem52 - sqrt(s(3))*dec35_remod; %SIC para
remover los datos de U3
                dec45 = step(BPSKdemod, rem53);    %Demodulación
directa para obtener los datos de U4
                dec45_remod = step(BPSKmod, dec45);
%Remodulacion de los datos de U4

```

```

        rem54 = rem53 - sqrt(s(4))*dec45_remod; %SIC para
remover los datos de U4
        dec(:,j) = step(BPSKdemod, rem54);
%Demodulación directa para obtener los datos de U5
        elseif j == 6
            dec16 = step(BPSKdemod, s_canal);
%Demodulación directa para obtener datos de U1
            dec16_remod = step(BPSKmod, dec16);
%Remodulación de los datos de U1
            rem61 = s_canal - sqrt(s(1))*dec16_remod; %SIC
para remover los datos de U1
            dec26 = step(BPSKdemod, rem61); %Demodulación
directa para obtener los datos de U2
            dec26_remod = step(BPSKmod, dec26);
%Remodulación de los datos de U2
            rem62 = rem61 - sqrt(s(2))*dec26_remod; %SIC para
remover los datos de U2
            dec36 = step(BPSKdemod, rem62); %Demodulación
directa para obtener los datos de U3
            dec36_remod = step(BPSKmod, dec36);
%Remodulación de los datos de U3
            rem63 = rem62 - sqrt(s(3))*dec36_remod; %SIC para
remover los datos de U3
            dec46 = step(BPSKdemod, rem63); %Demodulación
directa para obtener los datos de U4
            dec46_remod = step(BPSKmod, dec46);
%Remodulación de los datos de U4
            rem64 = rem63 - sqrt(s(4))*dec46_remod; %SIC para
remover los datos de U4
            dec56 = step(BPSKdemod, rem64); %Demodulación
directa para obtener los datos de U5
            dec56_remod = step(BPSKmod, dec56);
%Remodulación de los datos de U5
            rem65 = rem64 - sqrt(s(5))*dec56_remod; %SIC para
remover los datos de U5
            dec(:,j) = step(BPSKdemod, rem65);
%Demodulación directa para obtener los datos de U6
        end
    end
case 2
    QPSKdemod = comm.QPSKDemodulator('BitOutput',true);
    for j = 1:1:M
        if j == 1
            dec(:,j) = step(QPSKdemod, s_canal); %
Demodulación directa para obtener datos de U1
            elseif j == 2
                dec12 = step(QPSKdemod, s_canal);
%Demodulación directa para obtener datos de U1
                dec12_remod = step(QPSKmod, dec12);
%Remodulación de los datos de U1
                rem2 = s_canal - sqrt(s(1))*dec12_remod; %SIC
para remover los datos de U1
                dec(:,j) = step(QPSKdemod, rem2);
%Demodulación directa para obtener los datos de U2
            elseif j == 3
                dec13 = step(QPSKdemod, s_canal);
%Demodulación directa para obtener datos de U1

```

```

        dec13_remod = step(QPSKmod, dec13);
%Remodulacion de los datos de U1
        rem31 = s_canal - sqrt(s(1))*dec13_remod;    %SIC
para remover los datos de U1
        dec23 = step(QPSKdemod, rem31);    %Demodulación
directa para obtener los datos de U2
        dec23_remod = step(QPSKmod, dec23);
%Remodulacion de los datos de U2
        rem32 = rem31 - sqrt(s(2))*dec23_remod; %SIC para
remover los datos de U2
        dec(:,j) = step(QPSKdemod, rem32);
%Demodulación directa para obtener los datos de U3
        elseif j == 4
            dec14 = step(QPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
            dec14_remod = step(QPSKmod, dec14);
%Remodulacion de los datos de U1
            rem41 = s_canal - sqrt(s(1))*dec14_remod;    %SIC
para remover los datos de U1
            dec24 = step(QPSKdemod, rem41);    %Demodulación
directa para obtener los datos de U2
            dec24_remod = step(QPSKmod, dec24);
%Remodulacion de los datos de U2
            rem42 = rem41 - sqrt(s(2))*dec24_remod; %SIC para
remover los datos de U2
            dec34 = step(QPSKdemod, rem42);    %Demodulación
directa para obtener los datos de U3
            dec34_remod = step(QPSKmod, dec34);
%Remodulacion de los datos de U3
            rem43 = rem42 - sqrt(s(3))*dec34_remod; %SIC para
remover los datos de U3
            dec(:,j) = step(QPSKdemod, rem43);
%Demodulación directa para obtener los datos de U4
            elseif j == 5
                dec15 = step(QPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
                dec15_remod = step(QPSKmod, dec15);
%Remodulacion de los datos de U1
                rem51 = s_canal - sqrt(s(1))*dec15_remod;    %SIC
para remover los datos de U1
                dec25 = step(QPSKdemod, rem51);    %Demodulación
directa para obtener los datos de U2
                dec25_remod = step(QPSKmod, dec25);
%Remodulacion de los datos de U2
                rem52 = rem51 - sqrt(s(2))*dec25_remod; %SIC para
remover los datos de U2
                dec35 = step(QPSKdemod, rem52);    %Demodulación
directa para obtener los datos de U3
                dec35_remod = step(QPSKmod, dec35);
%Remodulacion de los datos de U3
                rem53 = rem52 - sqrt(s(3))*dec35_remod; %SIC para
remover los datos de U3
                dec45 = step(QPSKdemod, rem53);    %Demodulación
directa para obtener los datos de U4
                dec45_remod = step(QPSKmod, dec45);
%Remodulacion de los datos de U4

```

```

        rem54 = rem53 - sqrt(s(4))*dec45_remod; %SIC para
remover los datos de U4
        dec(:,j) = step(QPSKdemod, rem54);
%Demodulación directa para obtener los datos de U5
        elseif j == 6
            dec16 = step(QPSKdemod, s_canal);
%Demodulacion directa para obtener datos de U1
            dec16_remod = step(QPSKmod, dec16);
%Remodulacion de los datos de U1
            rem61 = s_canal - sqrt(s(1))*dec16_remod; %SIC
para remover los datos de U1
            dec26 = step(QPSKdemod, rem61); %Demodulación
directa para obtener los datos de U2
            dec26_remod = step(QPSKmod, dec26);
%Remodulacion de los datos de U2
            rem62 = rem61 - sqrt(s(2))*dec26_remod; %SIC para
remover los datos de U2
            dec36 = step(QPSKdemod, rem62); %Demodulación
directa para obtener los datos de U3
            dec36_remod = step(QPSKmod, dec36);
%Remodulacion de los datos de U3
            rem63 = rem62 - sqrt(s(3))*dec36_remod; %SIC para
remover los datos de U3
            dec46 = step(QPSKdemod, rem63); %Demodulación
directa para obtener los datos de U4
            dec46_remod = step(QPSKmod, dec46);
%Remodulacion de los datos de U4
            rem64 = rem63 - sqrt(s(4))*dec46_remod; %SIC para
remover los datos de U4
            dec56 = step(QPSKdemod, rem64); %Demodulación
directa para obtener los datos de U5
            dec56_remod = step(QPSKmod, dec56);
%Remodulacion de los datos de U5
            rem65 = rem64 - sqrt(s(5))*dec56_remod; %SIC para
remover los datos de U5
            dec(:,j) = step(QPSKdemod, rem65);
%Demodulación directa para obtener los datos de U6
        end
    end
case 3
    for j = 1:1:M
        if j == 1
            dec(:,j) = qamdemod(s_canal,4, 'OutputType', 'bit');
% Demodulacion directa para obtener datos de U1
            elseif j == 2
                dec12 = qamdemod(s_canal,4, 'OutputType', 'bit');
%Demodulacion directa para obtener datos de U1
                dec12_remod = qammod(dec12,4, 'InputType', 'bit');
%Remodulacion de los datos de U1
                rem2 = s_canal - sqrt(s(1))*dec12_remod; %SIC
para remover los datos de U1
                dec(:,j) = qamdemod(rem2,4, 'OutputType', 'bit');
%Demodulación directa para obtener los datos de U2
            elseif j == 3
                dec13 = qamdemod(s_canal,4, 'OutputType', 'bit');
%Demodulacion directa para obtener datos de U1

```

```

        dec13_remod = qammod(dec13,4,'InputType','bit');
%Remodulacion de los datos de U1
        rem31 = s_canal - sqrt(s(1))*dec13_remod;    %SIC
para remover los datos de U1
        dec23 = qamdemod(rem31,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U2
        dec23_remod = qammod(dec23,4,'InputType','bit');
%Remodulacion de los datos de U2
        rem32 = rem31 - sqrt(s(2))*dec23_remod; %SIC para
remover los datos de U2
        dec(:,j) = qamdemod(rem32,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U3
        elseif j == 4
            dec14 = qamdemod(s_canal,4,'OutputType','bit');
%Demodulacion directa para obtener datos de U1
            dec14_remod = qammod(dec14,4,'InputType','bit');
%Remodulacion de los datos de U1
            rem41 = s_canal - sqrt(s(1))*dec14_remod;    %SIC
para remover los datos de U1
            dec24 = qamdemod(rem41,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U2
            dec24_remod = qammod(dec24,4,'InputType','bit');
%Remodulacion de los datos de U2
            rem42 = rem41 - sqrt(s(2))*dec24_remod; %SIC para
remover los datos de U2
            dec34 = qamdemod(rem42,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U3
            dec34_remod = qammod(dec34,4,'InputType','bit');
%Remodulacion de los datos de U3
            rem43 = rem42 - sqrt(s(3))*dec34_remod; %SIC para
remover los datos de U3
            dec(:,j) = qamdemod(rem43,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U4
            elseif j == 5
                dec15 = qamdemod(s_canal,4,'OutputType','bit');
%Demodulacion directa para obtener datos de U1
                dec15_remod = qammod(dec15,4,'InputType','bit');
%Remodulacion de los datos de U1
                rem51 = s_canal - sqrt(s(1))*dec15_remod;    %SIC
para remover los datos de U1
                dec25 = qamdemod(rem51,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U2
                dec25_remod = qammod(dec25,4,'InputType','bit');
%Remodulacion de los datos de U2
                rem52 = rem51 - sqrt(s(2))*dec25_remod; %SIC para
remover los datos de U2
                dec35 = qamdemod(rem52,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U3
                dec35_remod = qammod(dec35,4,'InputType','bit');
%Remodulacion de los datos de U3
                rem53 = rem52 - sqrt(s(3))*dec35_remod; %SIC para
remover los datos de U3
                dec45 = qamdemod(rem53,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U4
                dec45_remod = qammod(dec45,4,'InputType','bit');
%Remodulacion de los datos de U4

```



```

        rem54 = rem53 - sqrt(s(4))*dec45_remod; %SIC para
remover los datos de U4
        dec(:,j) = qamdemod(rem54,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U5
        elseif j == 6
            dec16 = qamdemod(s_canal,4,'OutputType','bit');
%Demodulacion directa para obtener datos de U1
            dec16_remod = qammod(dec16,4,'InputType','bit');
%Remodulacion de los datos de U1
            rem61 = s_canal - sqrt(s(1))*dec16_remod; %SIC
para remover los datos de U1
            dec26 = qamdemod(rem61,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U2
            dec26_remod = qammod(dec26,4,'InputType','bit');
%Remodulacion de los datos de U2
            rem62 = rem61 - sqrt(s(2))*dec26_remod; %SIC para
remover los datos de U2
            dec36 = qamdemod(rem62,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U3
            dec36_remod = qammod(dec36,4,'InputType','bit');
%Remodulacion de los datos de U3
            rem63 = rem62 - sqrt(s(3))*dec36_remod; %SIC para
remover los datos de U3
            dec46 = qamdemod(rem63,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U4
            dec46_remod = qammod(dec46,4,'InputType','bit');
%Remodulacion de los datos de U4
            rem64 = rem63 - sqrt(s(4))*dec46_remod; %SIC para
remover los datos de U4
            dec56 = qamdemod(rem64,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U5
            dec56_remod = qammod(dec56,4,'InputType','bit');
%Remodulacion de los datos de U5
            rem65 = rem64 - sqrt(s(5))*dec56_remod; %SIC para
remover los datos de U5
            dec(:,j) = qamdemod(rem65,4,'OutputType','bit');
%Demodulación directa para obtener los datos de U6
        end
    end
end
for u = 1:1:M
    [nbit(i,u), ber(i,u)] = biterr(dec(:,u), x(:,u)); % Cálculo del BER
end
end

```

ANEXO B: Libros de código utilizado para la simulación de SCMA.

Los libros de códigos utilizados son propuestos por la empresa Altera, en la primera competencia de innovación de algoritmo SCMA para 5G.

SCMA Codebook index	SCMA codebook for each layer
CB_1	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ -0.1815 - 0.1318i & -0.6351 - 0.4615i & 0.6351 + 0.4615i & 0.1815 + 0.1318i \\ 0 & 0 & 0 & 0 \\ 0.7851 & -0.2243 & 0.2243 & -0.7851 \end{bmatrix}$
CB_2	$\begin{bmatrix} 0.7851 & -0.2243 & 0.2243 & -0.7851 \\ 0 & 0 & 0 & 0 \\ -0.1815 - 0.1318i & -0.6351 - 0.4615i & 0.6351 + 0.4615i & 0.1815 + 0.1318i \\ 0 & 0 & 0 & 0 \end{bmatrix}$
CB_3	$\begin{bmatrix} -0.6351 + 0.4615i & 0.1815 - 0.1318i & -0.1815 + 0.1318i & 0.6351 - 0.4615i \\ 0.1392 - 0.1759i & 0.4873 - 0.6156i & -0.4873 + 0.6156i & -0.1392 + 0.1759i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$
CB_4	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.7851 & -0.2243 & 0.2243 & -0.7851 \\ -0.0055 - 0.2242i & -0.0193 - 0.7848i & 0.0193 + 0.7848i & 0.0055 + 0.2242i \end{bmatrix}$
CB_5	$\begin{bmatrix} -0.0055 - 0.2242i & -0.0193 - 0.7848i & 0.0193 + 0.7848i & 0.0055 + 0.2242i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -0.6351 + 0.4615i & 0.1815 - 0.1318i & -0.1815 + 0.1318i & 0.6351 - 0.4615i \end{bmatrix}$
CB_6	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.7851 & -0.2243 & 0.2243 & -0.7851 \\ 0.1392 - 0.1759i & 0.4873 - 0.6156i & -0.4873 + 0.6156i & -0.1392 + 0.1759i \\ 0 & 0 & 0 & 0 \end{bmatrix}$

Estos libros de códigos se implementan en MATLAB de la siguiente manera:

```

%% Libros de códigos
CB(:, :, 1) = [ 0                0                0
0; ...
                -0.1815-1j*0.1318  -0.6351-1j*0.4615  0.6351+1j*0.4615
0.1815+1j*0.1318; ...
                0                0                0
0; ...
                0.7851            -0.2243            0.2243
-0.7851 ];

CB(:, :, 2) = [ 0.7851            -0.2243            0.2243
-0.7851; ...
                0                0                0
0; ...

```

```

-0.1815-1j*0.1318 -0.6351-1j*0.4615 0.6351+1j*0.4615
0.1815+1j*0.1318;...
0
0 ];
```

```

CB(:, :, 3) = [-0.6351+1j*0.4615 0.1815-1j*0.1318 -0.1815+1j*0.1318
0.6351-1j*0.4615;...
0.1392-1j*0.1759 0.4873-1j*0.6156 -0.4873+1j*0.6156
-0.1392+1j*0.1759;...
0
0;...
0 ];
```

```

CB(:, :, 4) = [ 0 0 0
0;...
0
0;...
0.7851 -0.2243 0.2243
-0.7851;...
-0.0055-1j*0.2242 -0.0193-1j*0.7848 0.0193+1j*0.7848
0.0055+1j*0.2242 ];
```

```

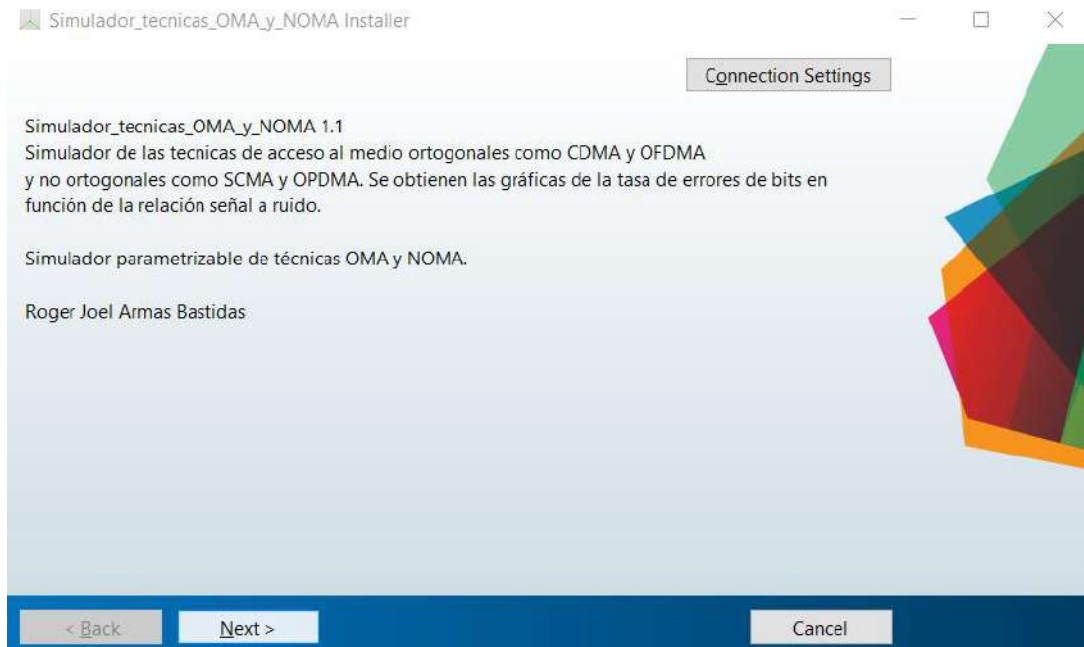
CB(:, :, 5) = [-0.0055-1j*0.2242 -0.0193-1j*0.7848 0.0193+1j*0.7848
0.0055+1j*0.2242;...
0
0;...
0
0;...
-0.6351+1j*0.4615 0.1815-1j*0.1318 -0.1815+1j*0.1318
0.6351-1j*0.4615 ];
```

```

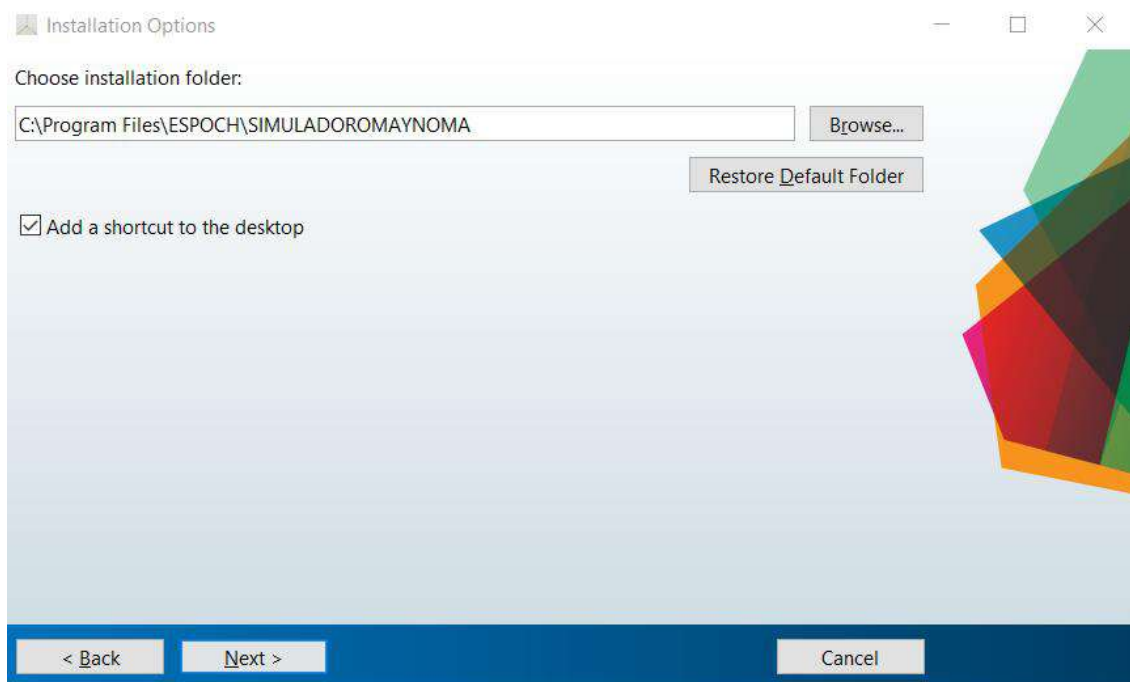
CB(:, :, 6) = [ 0 0 0
0;...
0.7851 -0.2243 0.2243
-0.7851;...
0.1392-1j*0.1759 0.4873-1j*0.6156 -0.4873+1j*0.6156
-0.1392+1j*0.1759;...
0
0 ];
```

ANEXO C: Instalación del simulador de interfaz gráfica.

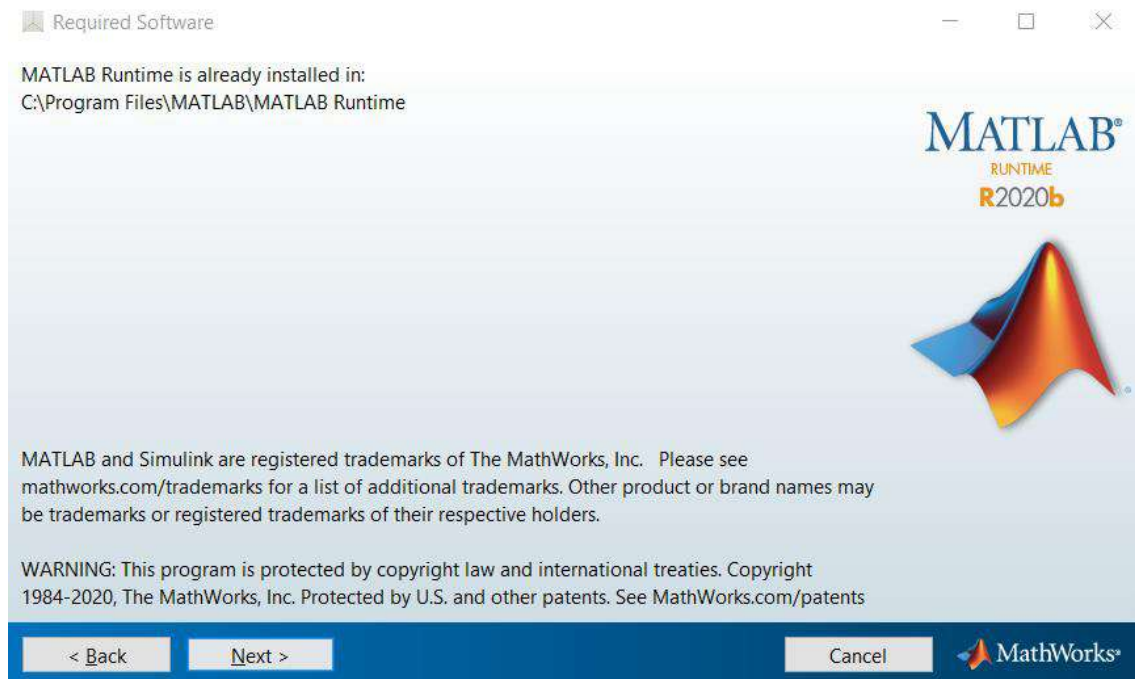
1. Se procede a descomprimir los archivos necesarios del instalador MyAppInstaller. Copiar el archivo vcruntime140.dll y pegarlo en el siguiente directorio: C:/Windows/System32.
2. Ejecutar como administrador el instalador MyAppInstaller.exe.



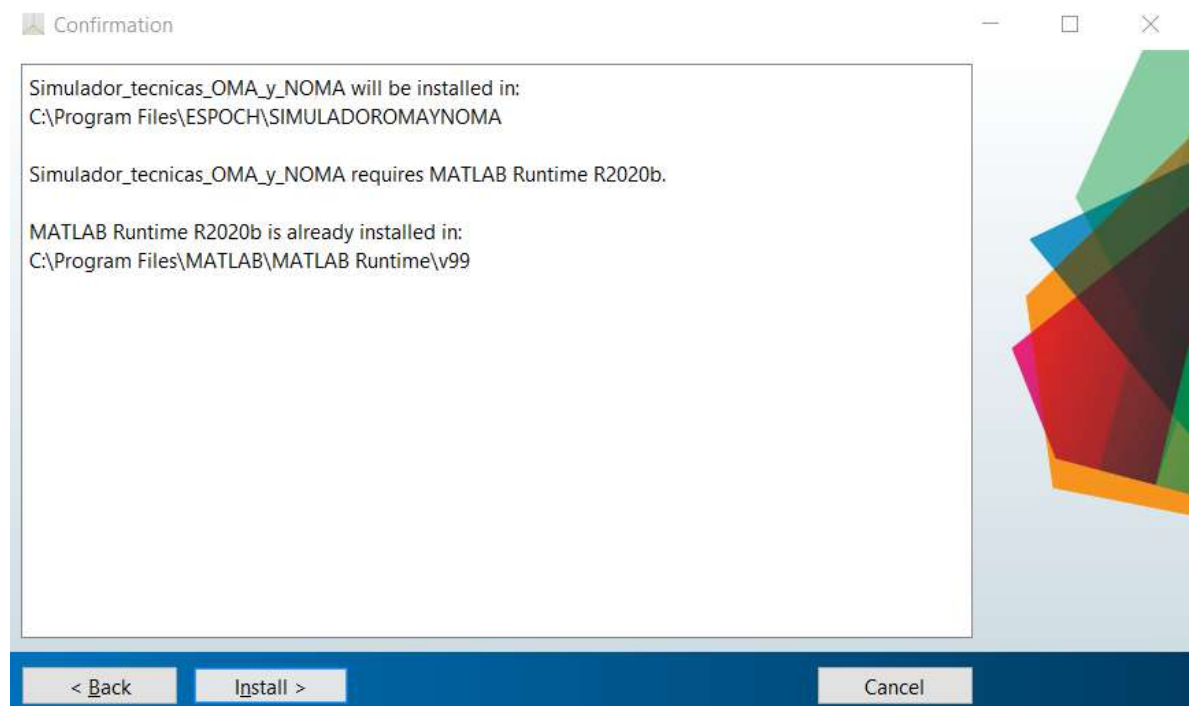
3. Dar clic en siguiente, y marcar el casillero para añadir el icono de la aplicación en el escritorio del computador.



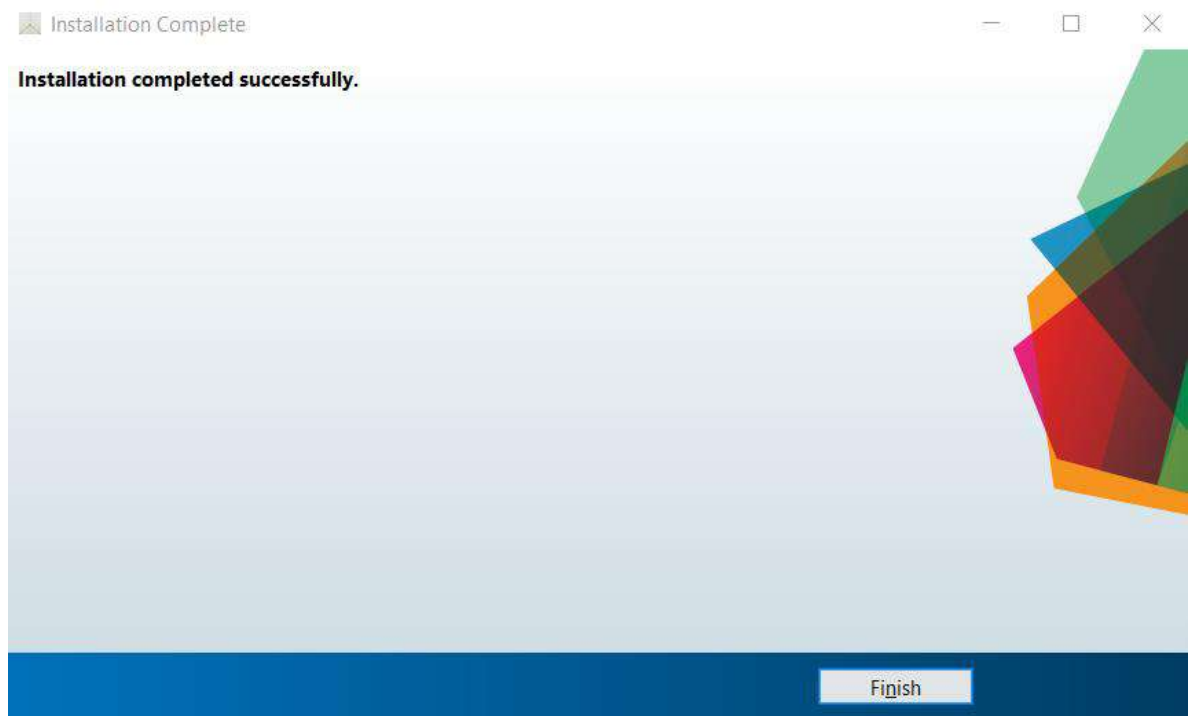
4. Si no se tiene instalado MATLAB en el computador y es la primera vez instalando el simulador, se debe aceptar los términos para la instalación de las librerías Runtime que permiten la ejecución del simulador. En caso de tener instalado, dar clic en siguiente.



5. Verificar las direcciones donde se ubicará el simulador dentro del computador y dar clic en Instalar.



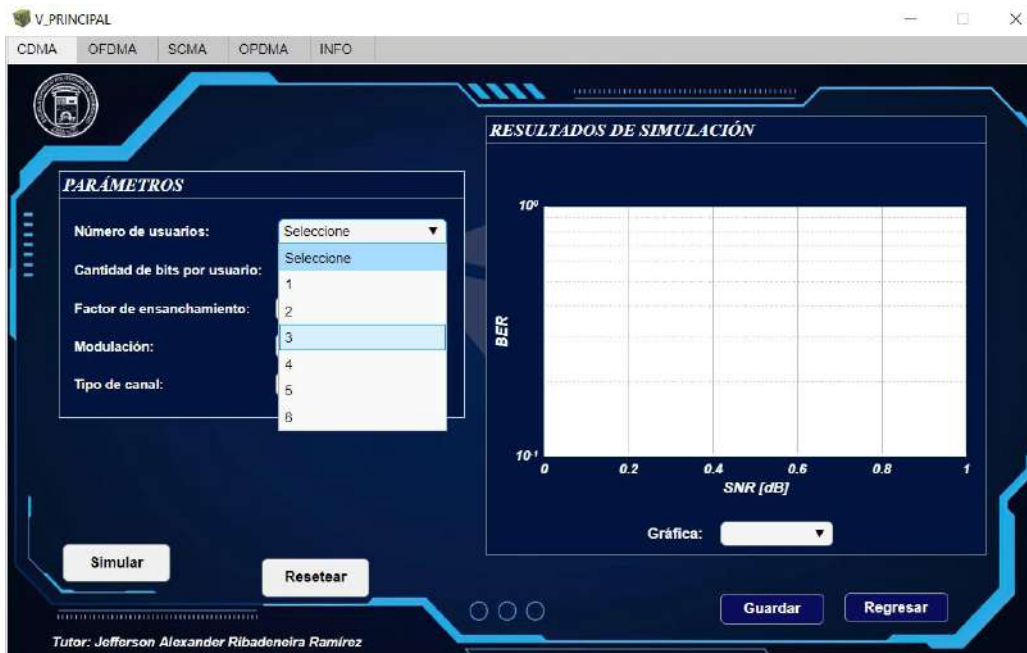
6. Por último, se procede a finalizar la instalación dando clic en el botón Finish,



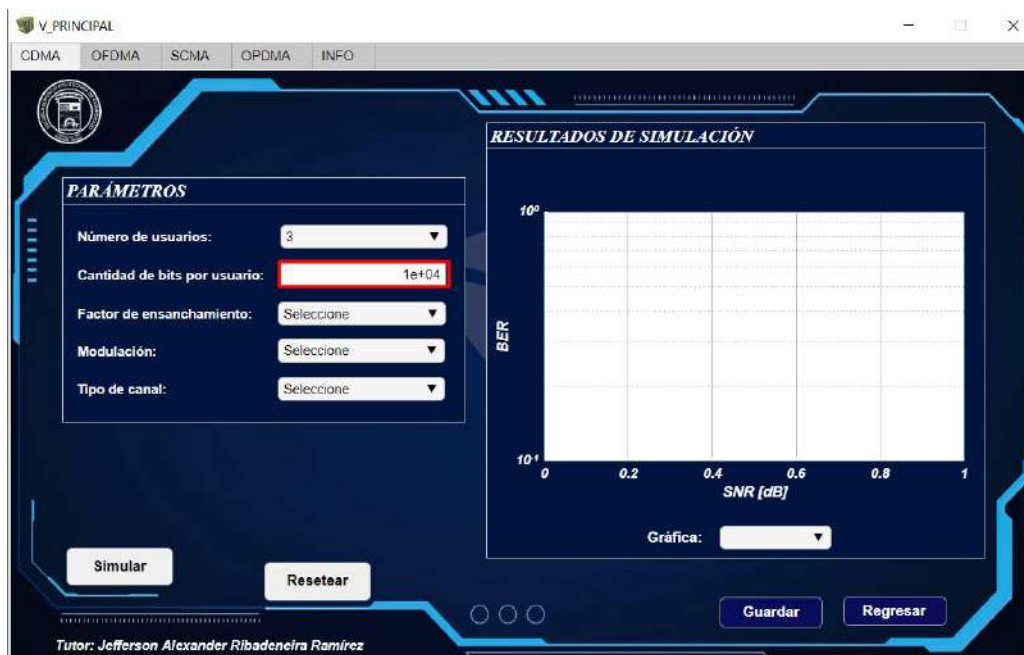
ANEXO D: Manual de simulación para cada técnica.

- CDMA

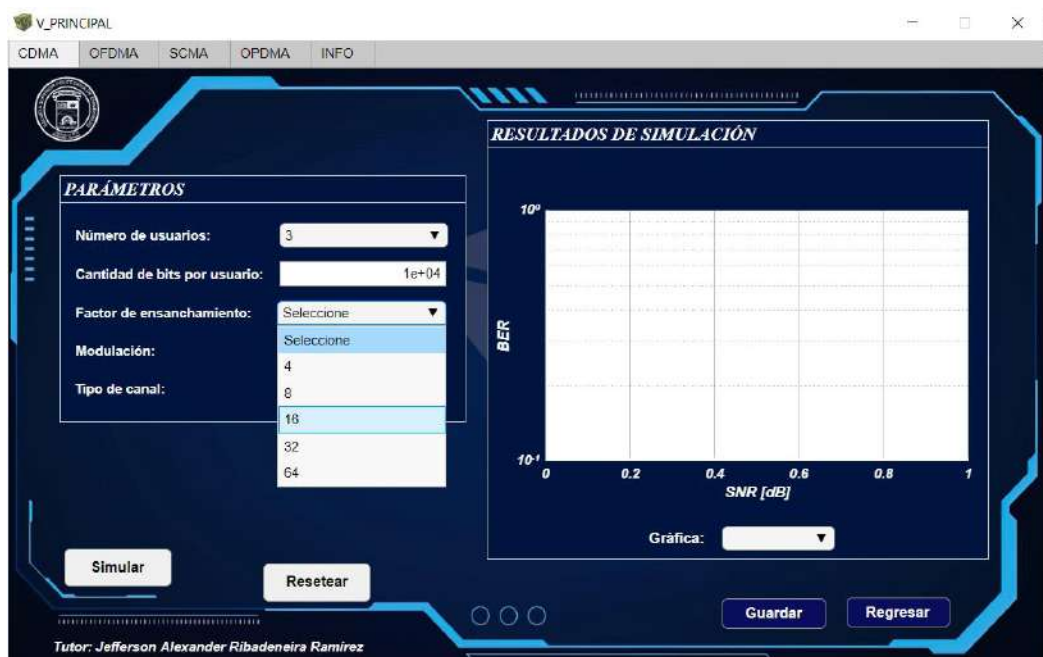
1. Seleccionar el número de usuarios a simular.



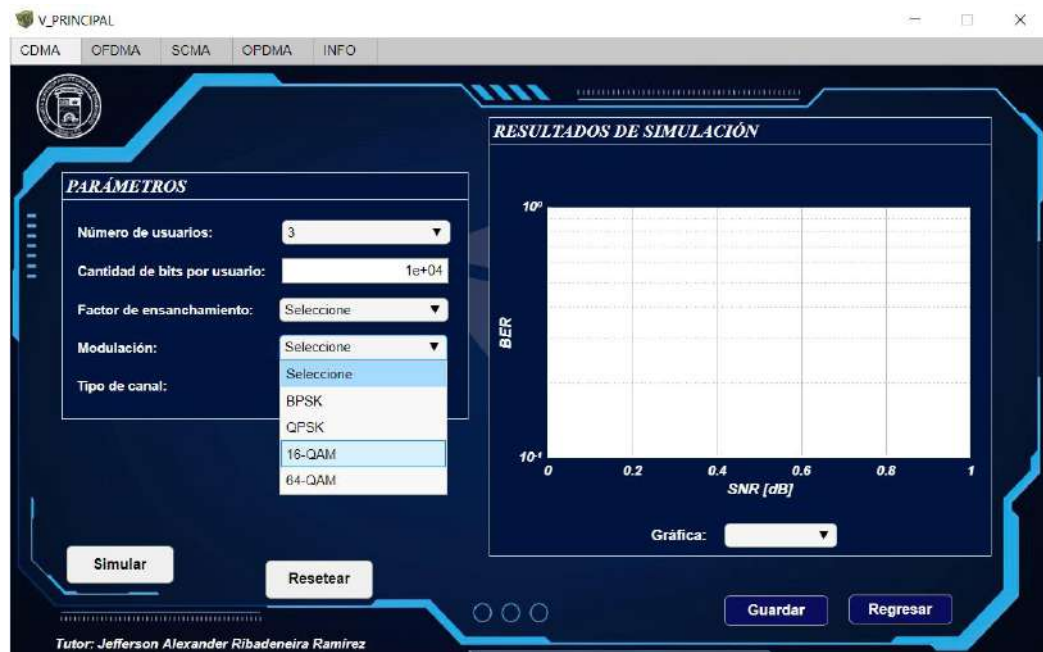
2. Ingresar la cantidad de bits a analizar. Si en el escenario propuesto a simularse, se escoge un esquema de modulación M-QAM, se debe ingresar una cantidad de bits múltiplo de $\log_2(M)$.



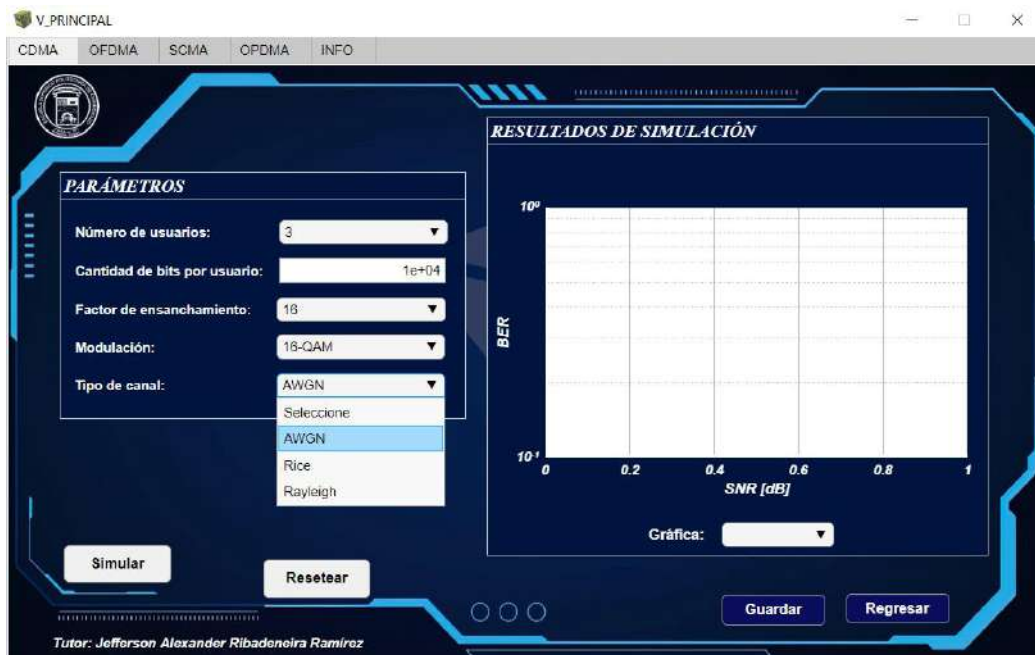
3. Seleccionar el factor de ensanchamiento en base a la tasa de transferencia de datos.



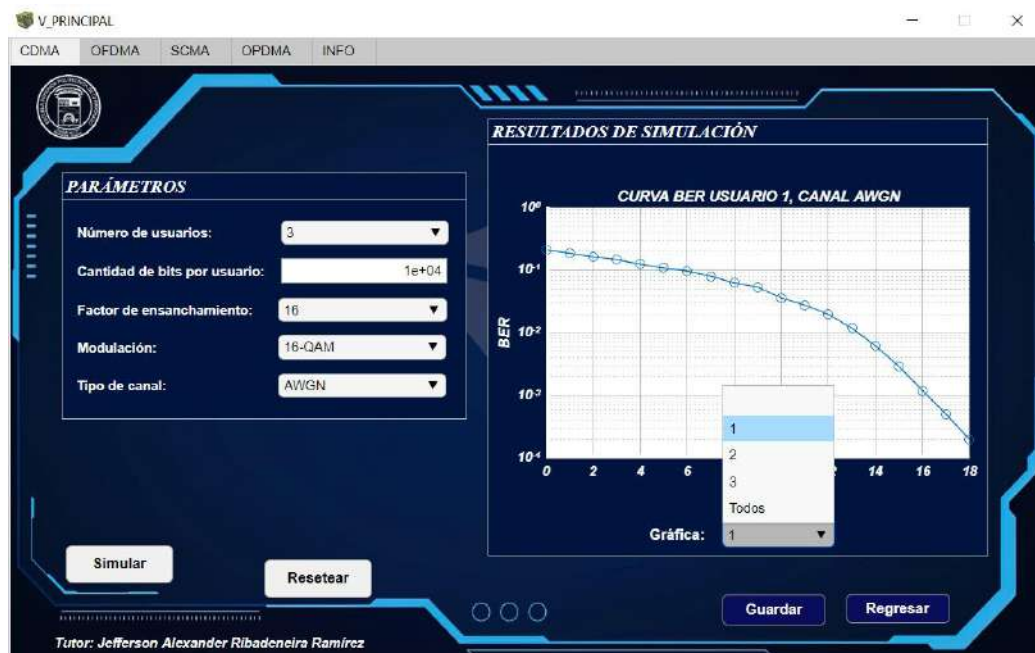
4. Seleccionar el esquema de modulación.



5. Seleccionar el canal inalámbrico por el cual pasará la información.

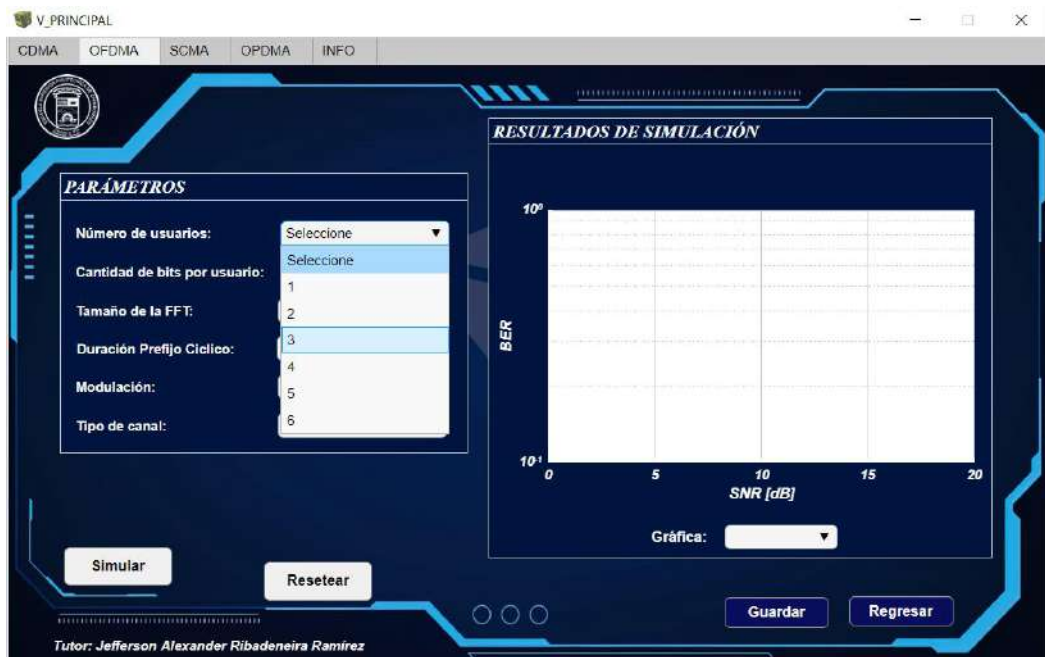


6. Dar clic en Simular para que el simulador empiece a realizar los cálculos. Aparecerá una ventana de espera y al final una ventana de aviso cuando la simulación esté lista. Para observar las gráficas seleccionamos en el menú desplegable Gráfica el número de usuario a visualizar.

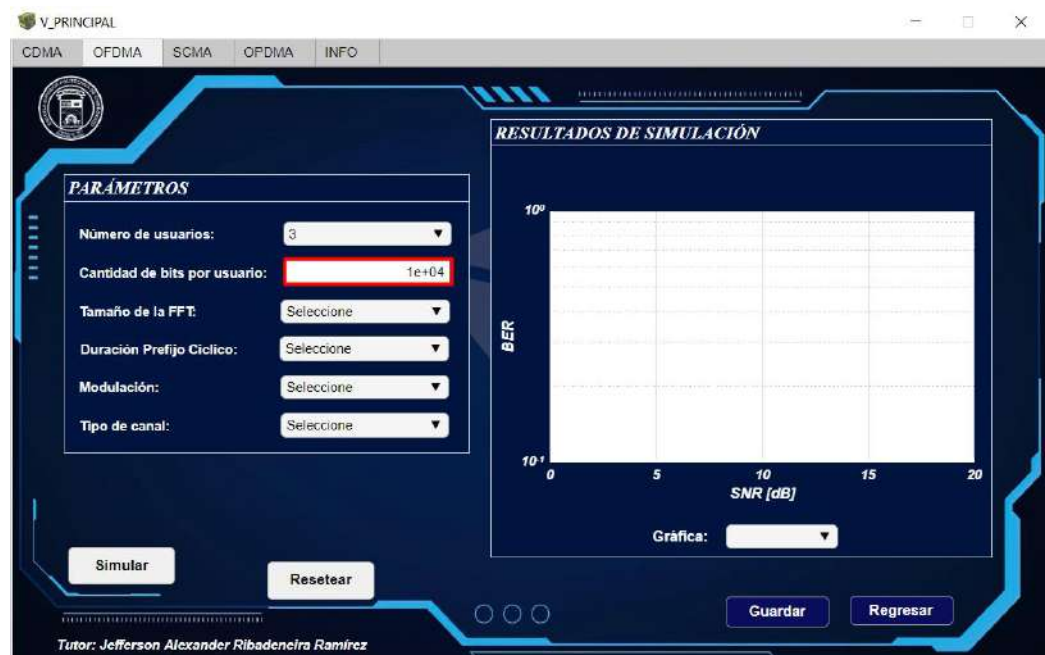


- OFDMA

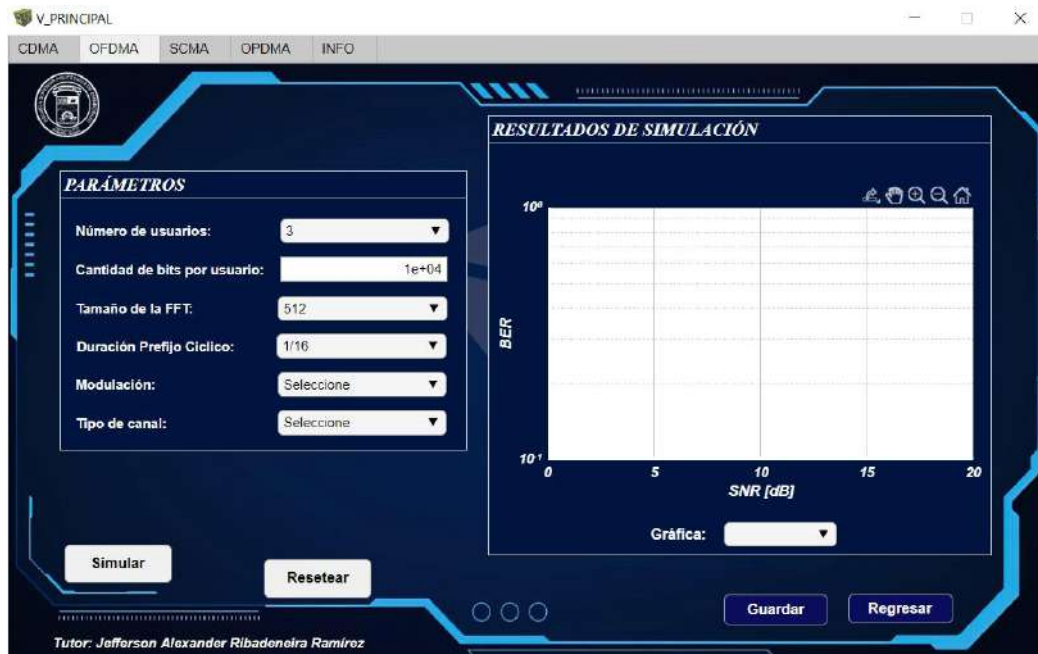
1. Seleccionar el número de usuario a simularse.



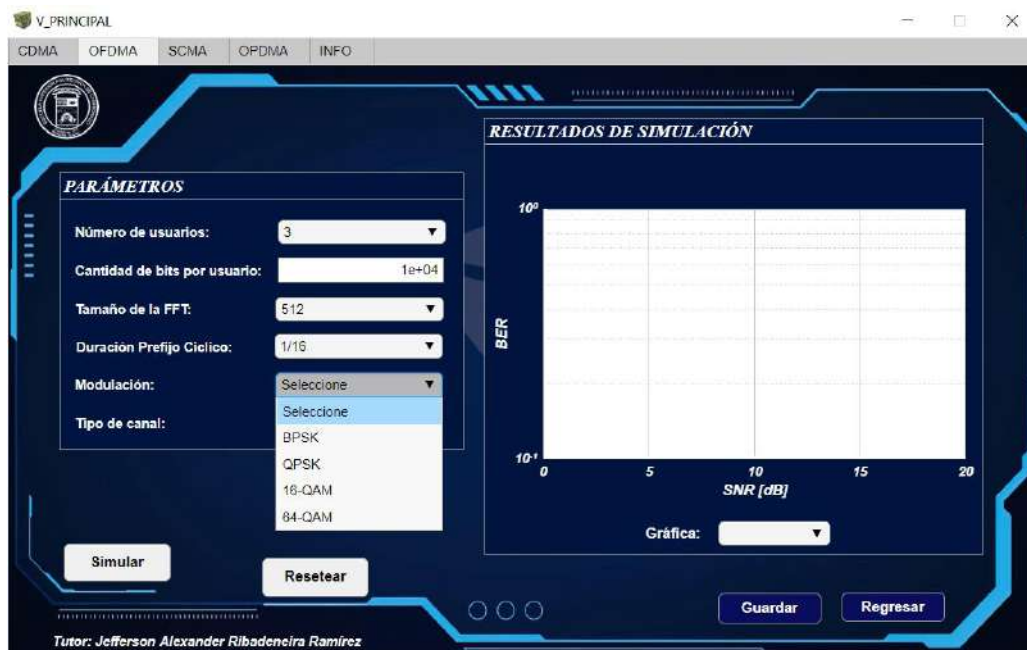
2. Ingresar la cantidad de bits a generarse para cada usuario seleccionado. Si en el escenario propuesto a simularse, se escoge un esquema de modulación M-QAM, se debe ingresar una cantidad de bits múltiplo de $\log_2(M)$.



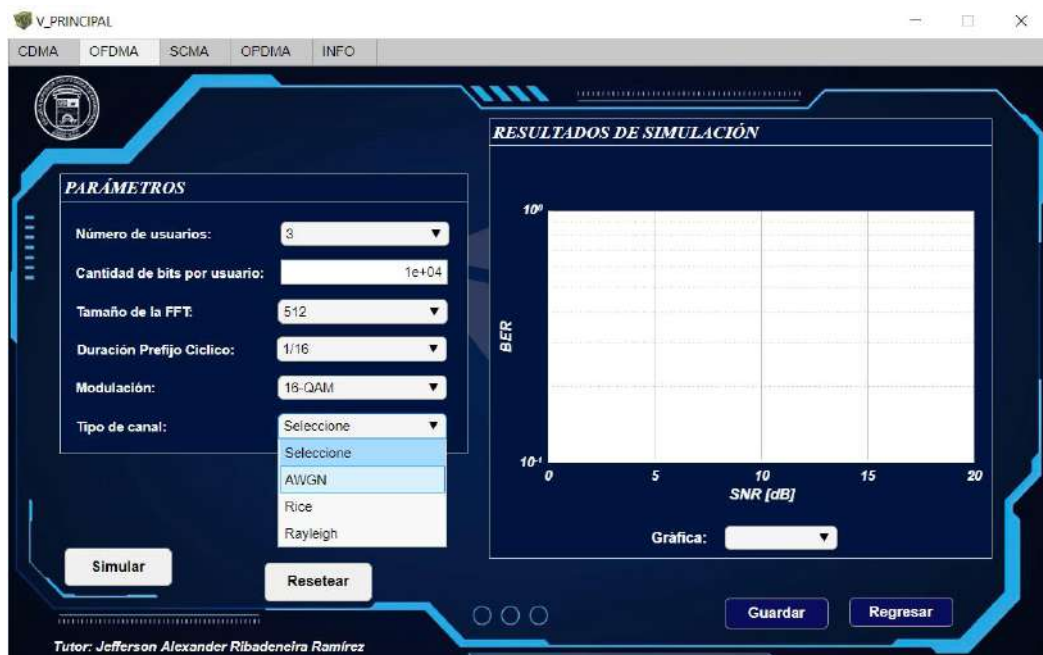
3. Seleccionar los parámetros característicos de esta técnica como el tamaño de la IFFT, este parámetro se ingresa en base al ancho de banda de transmisión. La duración del prefijo cíclico se encuentra en fracciones de un símbolo OFDM, el cual será ingresado al inicio de cada símbolo de la trama a transmitir en esta técnica.



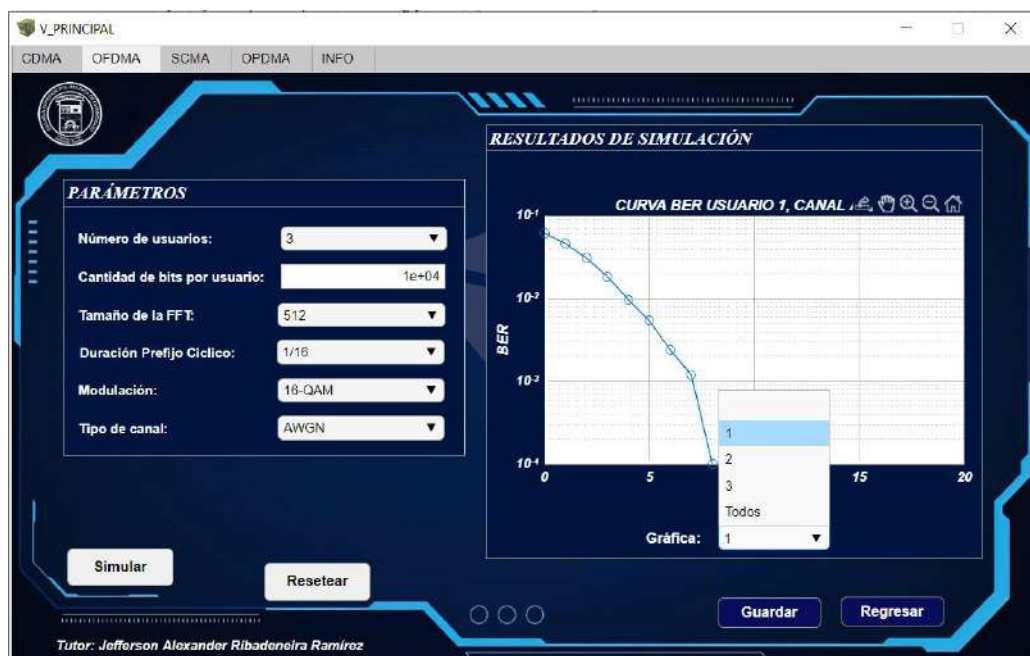
4. Seleccionar el esquema de modulación a simular.



5. Seleccionar el canal inalámbrico sobre el cual la información se verá afectada.

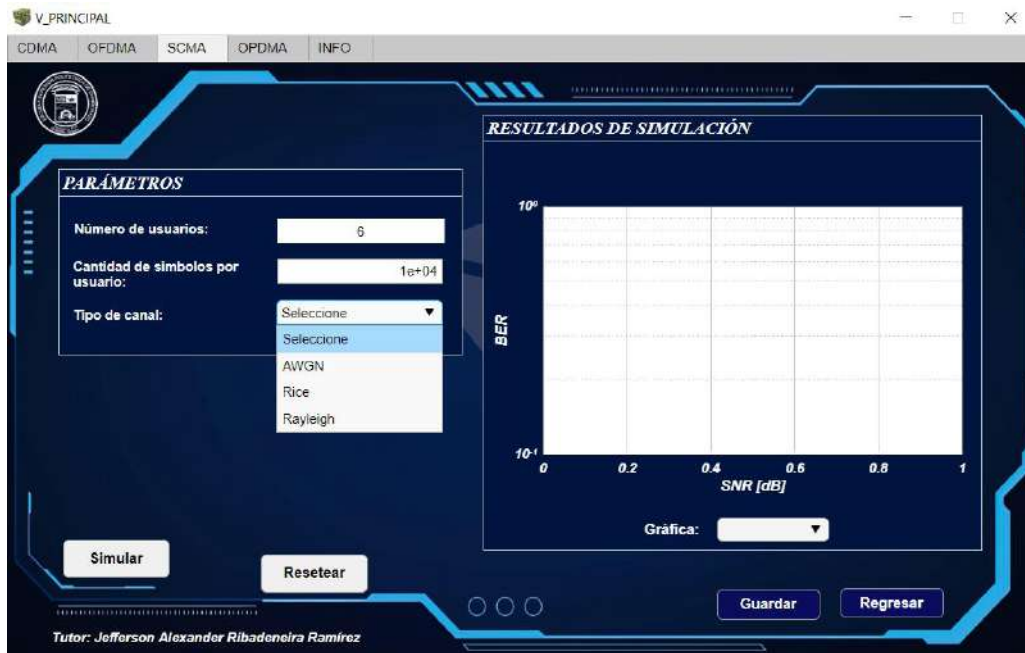


6. Una vez ingresado todos los parámetros, se procede a dar clic en el botón Simular. Para visualizar las gráficas del BER seleccionamos un usuario en el menú desplegable Gráfica.

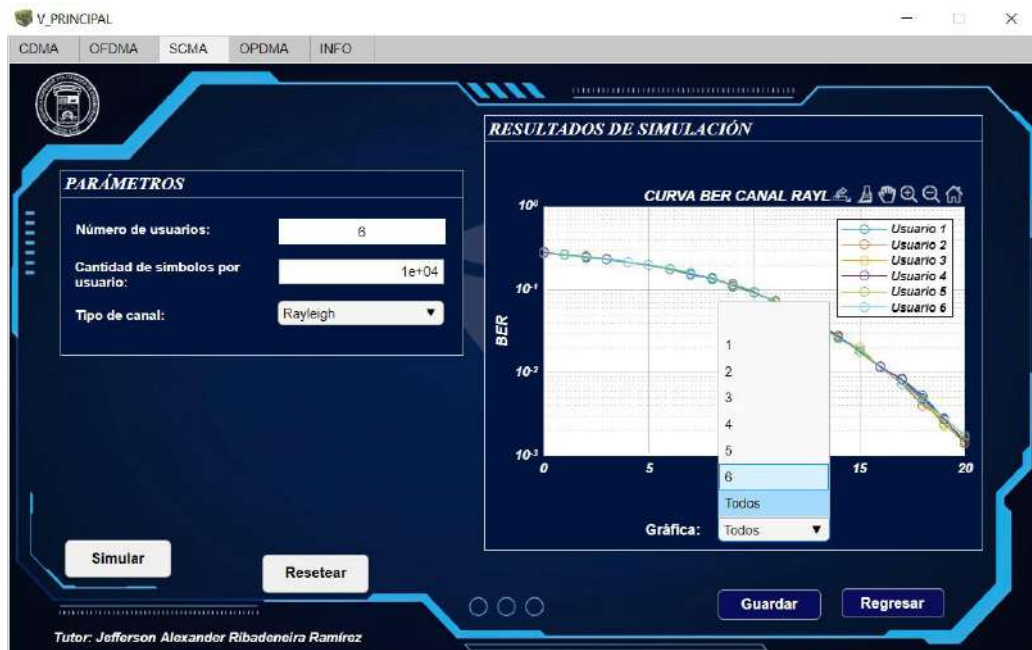


- SCMA

1. En el simulador de esta técnica solo se puede cambiar los valores de símbolos a generar para los 6 usuarios y el tipo de canal inalámbrico por el cual pasará la señal SCMA.

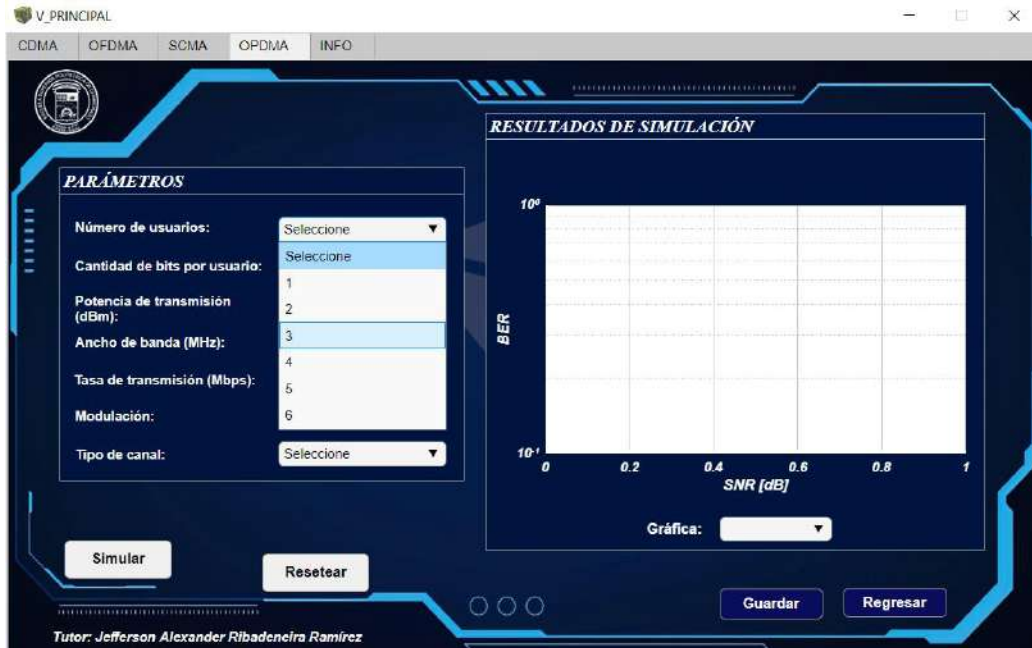


2. Luego se procede a dar clic en Simular y seleccionar el usuario del que se requiere observar la curva BER respectiva.

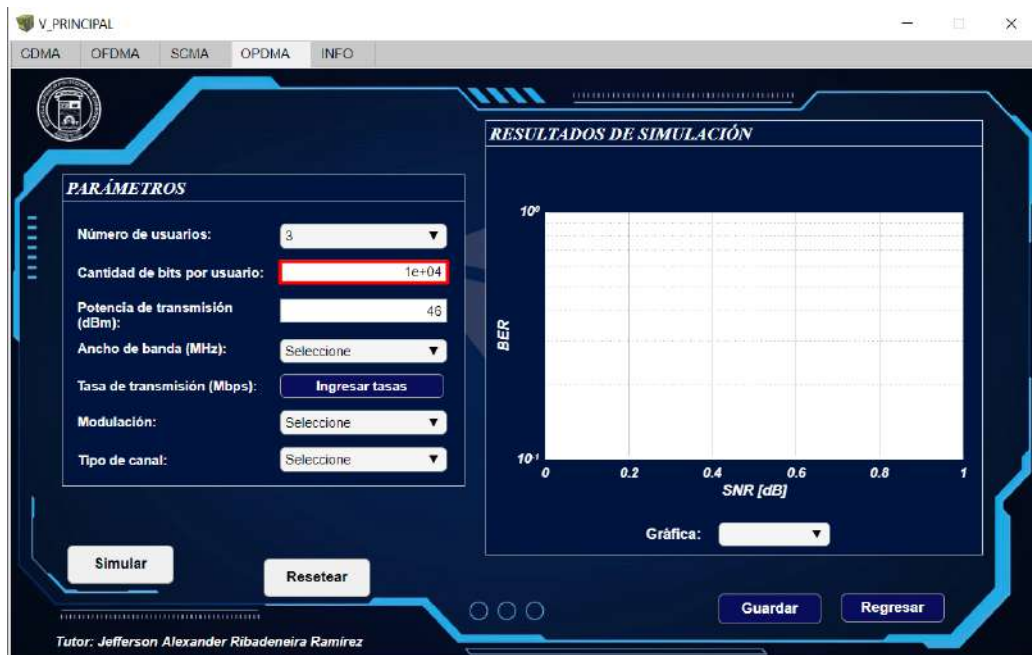


- OPDMA

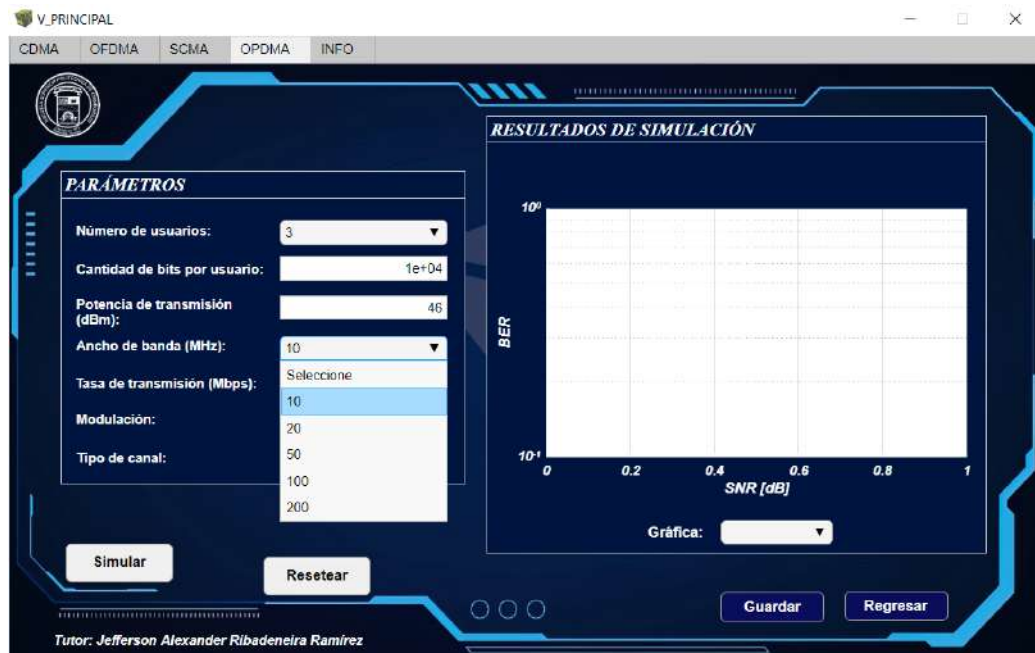
1. Seleccionar el número de usuario a simular.



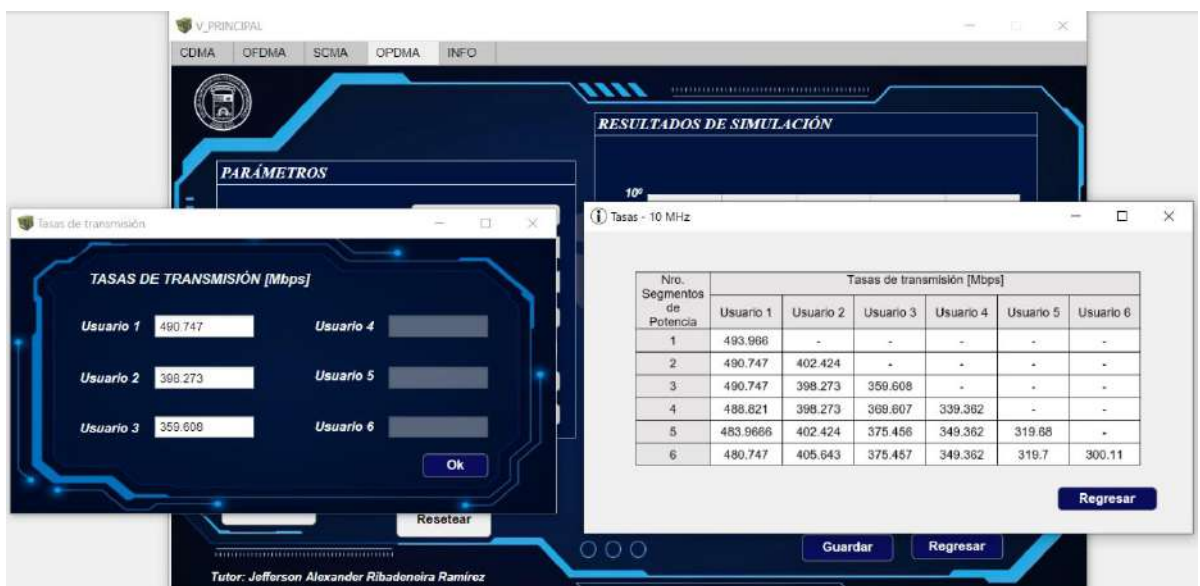
2. Ingresar la cantidad de bits a generar para cada usuario. Esta técnica no implica restricciones en cuanto a la cantidad debido al esquema de modulación seleccionado.



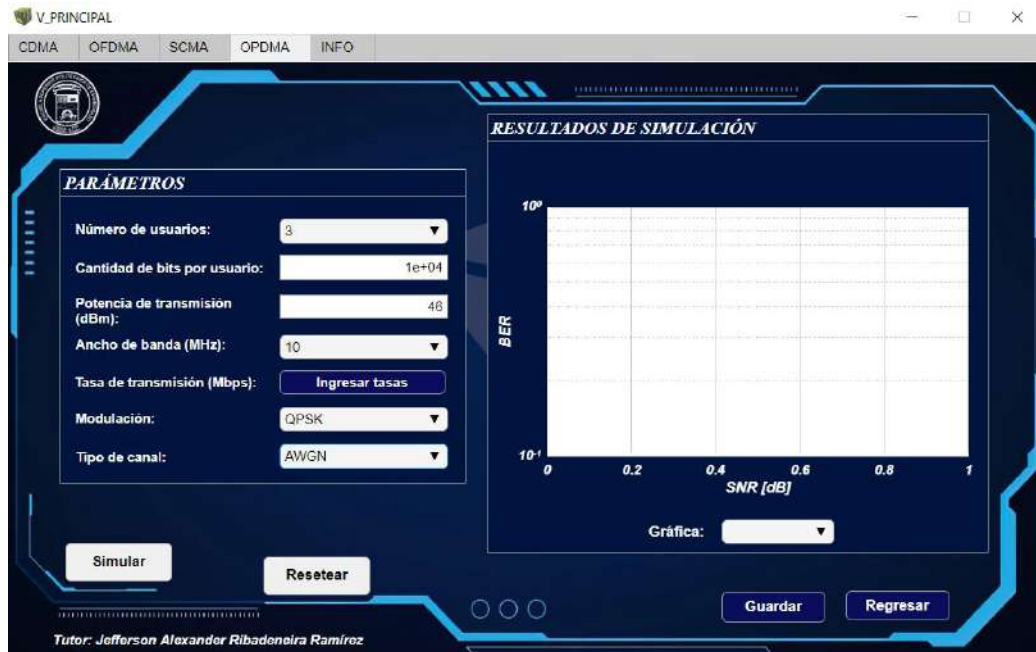
3. Ingresar el valor de potencia de transmisión en dBm, cuyo valor está establecido en 46 dBm por defecto. También se debe seleccionar el ancho de banda para la transmisión de los bits sobre el canal.



4. Para ingresar las tasas de transmisión de datos, se debe dirigir a la pestaña de INFO y dar clic en el botón tasas. Ahí se desplegará una ventana donde se puede seleccionar el ancho de banda que se seleccionó anteriormente. Al dar clic en un botón, se muestra otra ventana donde se visualiza una tabla de las tasas de transmisión en Mbps para cada usuario en función al número de segmentos de potencia calculados. Presionando el botón ingresar tasas aparecerá una ventana con los campos de texto para ingresar los valores de tasas mostrados en la tabla.



5. Seleccionar el esquema de modulación y el tipo de canal por el cual las señales serán transmitidas.

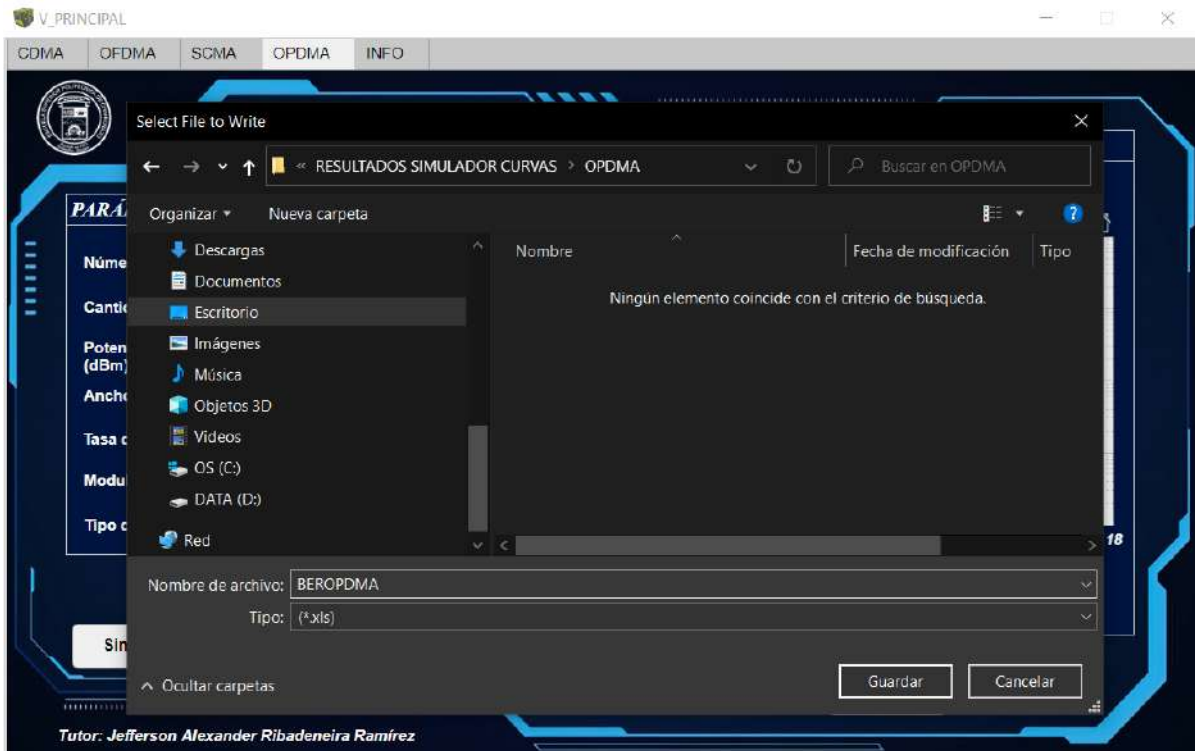


6. Por último se procede a dar clic en el botón Simular y se selecciona el número de usuario en el menú desplegable Gráfica ubicado en la parte inferior de la gráfica de curva BER.



- Exportar resultados en formato de hoja de calculo

1. Dar clic en el botón Guardar, una vez realizado esto, se desplegará el explorador de archivos de Windows para seleccionar la ubicación del archivo en formatos compatibles con Excel.



2. Ubicar el archivo en formato .xls y verificar los resultados.

The image shows an Excel spreadsheet with the following data:

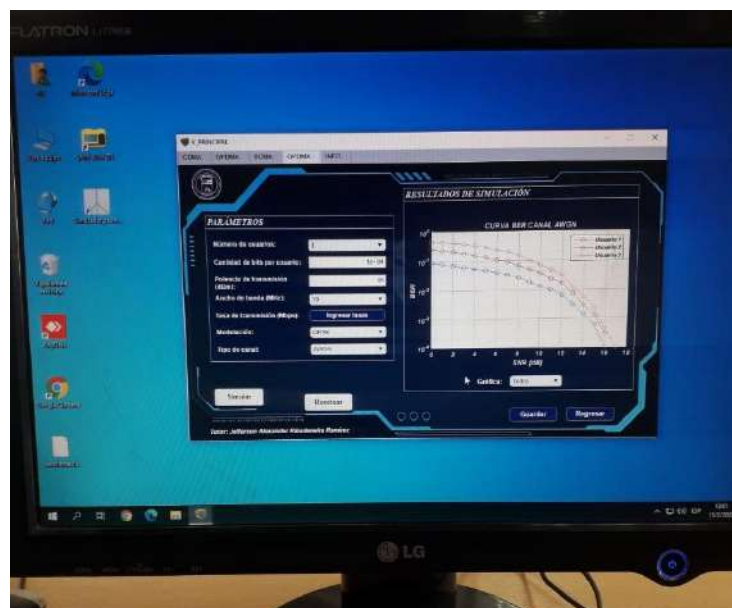
SNR (dB)	BER U1	BER U2	BER U3
0	0,0895	0,2579	0,4417
1	0,0797	0,2321	0,4256
2	0,0676	0,2022	0,3966
3	0,0568	0,1762	0,3734
4	0,0546	0,1589	0,33
5	0,0476	0,132	0,2959
6	0,039	0,111	0,2416
7	0,036	0,0937	0,1931
8	0,0272	0,0731	0,1533
9	0,0199	0,0518	0,1074
10	0,0125	0,0352	0,0722
11	0,0085	0,0235	0,047
12	0,0052	0,0132	0,0253
13	0,0023	0,002	0,0117
14	0,0017	0,0029	0,0063
15	0,0004	0,0008	0,0013
16	0,0002	0,0004	0,0006
17	0,0002	0,0003	0,0005
18	0	0	0
19	0	0	0
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0
25	0	0	0
26	0	0	0
27	0	0	0
28	0	0	0
29	0	0	0

ANEXO E: Implementación del simulador en el Laboratorio de comunicaciones de la FIE y comprobación del funcionamiento.

Una vez instalado el simulador se procede a realizar una simulación de prueba con el objetivo de comprobar el funcionamiento de este.



Al dar clic en simular se obtiene los resultados de la simulación correspondiente al BER en función del SNR.





epoch

**Dirección de Bibliotecas y
Recursos del Aprendizaje**

**UNIDAD DE PROCESOS TÉCNICOS Y ANÁLISIS BIBLIOGRÁFICO Y
DOCUMENTAL**

REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 13 / 06 / 2022

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: ROGER JOEL ARMAS BASTIDAS
INFORMACIÓN INSTITUCIONAL
Facultad: INFORMÁTICA Y ELECTRÓNICA
Carrera: TELECOMUNICACIONES
Título a optar: INGENIERO EN TELECOMUNICACIONES
f. Analista de Biblioteca responsable: Lcdo. Holger Ramos, MSc.

0966-DBRA-UPT-2022