

PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA INFORMATICA

**REDES DE KOHONEN PARA REDUCIR LA DISTORSION NO
LINEAL EN SISTEMAS INALAMBRICOS**

MARCELO ANDRES OYANADEL VELIZ

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

DICIEMBRE 2007

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**REDES DE KOHONEN PARA REDUCIR LA DISTORSION
NO LINEAL EN SISTEMAS INALAMBRICOS**

MARCELO ANDRES OYANADEL VELIZ

Profesor Guía: **Nibaldo Rodríguez Agurto**

Profesor Co-referente: **Broderick Crawford Labrín**

Carrera: **Ingeniería Civil en Informática**

Diciembre 2007

Agradecimientos
A mis padres y
mi profesor Dr. Nivaldo Rodríguez

Dedicado
A mi Madre, mi Padre y familia

Resumen

Esta memoria presenta la técnica de las redes neuronales de Kohonen para reducir la distorsión no lineal en sistemas inalámbricos basados en OFDM, para lo cual se hace un estudio completo de los tópicos para poder implementar un sistema OFDM elaborado en lenguaje JAVA que logre ratificar lo expuesto. Se logra componer un Sistema OFDM que minimiza la distorsión no lineal generada por los Amplificadores de Potencia SSPA y el Ruido Gaussian durante la transmisión de la señal, sistema que utiliza una red neuronal de Kohonen para lograr el objetivo. Luego se elabora un sistema en lenguaje JAVA basado en el algoritmo no supervisado de VQTAM, obteniendo satisfactorios resultados que algunos casos alcanzaron los 6 db para sistemas OFDM de 16QAM, sistema que utiliza la técnica de los Mínimos Cuadrados del algoritmo VQTAM.

Abstract

This memory displays the technique by means of the use of the neuronal networks of Kohonen to reduce the distortion nonlinear in wireless systems based on OFDM, for which a complete study made of the topics to be able to implement a OFDM system elaborated in language JAVA that manages to ratify the exposed thing. To be able to compose a System OFDM that minimize the distortion non-linear generated by the SSPA Power Amplifiers and the Gaussian Noise during the transmission of the signal, system that uses a neuronal network of Kohonen to achieve to got the objective. Later, a system in language JAVA based on the non-supervised algorithm of VQTAM obtained satisfactory results that some of cases reached 6 db for OFDM-16QAM systems, system that uses the technique of the Squared Minimums of algorithm VQTAM.

Indice

Indice	vii
Glosario de Términos	ix
Lista de Abreviaturas o siglas	x
Indice de Ilustraciones	xii
Lista de Tablas	xiv
CAPITULO 1: Introducción	1
1.1 Objetivo General	2
1.2 Objetivos Específicos	2
1.3 Organización de Texto	3
CAPITULO 2: Estado del Arte	4
2.1 Estándares de Transmisión Inalámbrica	4
2.1.1 WIMAX, Definición	4
2.1.2 Características de WIMAX	6
2.2 Distorsión No Lineal	7
2.2.1 Distorsión de Intermodulación	8
2.2.2 Distorsión sobre la Amplitud y la Fase	9
2.2.3 Distorsión no lineal en amplificadores	10
2.2.4 Descripción Paso Banda	12
2.3 Redes Neuronales	14
2.3.1 Características de las Redes Neuronales	16
2.3.2 Descripción Histórica de las Redes Neuronales	17
2.4 Tecnicas para Reducir la Distorsión no Lineal	18
CAPITULO 3: Modulación Multicarrier OFDM, Amplificadores de Potencia, y Redes de Kohonen	21
3.1 Modulación Multicarrier OFDM	21
3.1.1 Características de la modulación OFDM	22
3.1.2 Descripción Histórica de Modulación OFDM	22

3.1.3¿Cómo funciona OFDM?	23
3.1.4 Representación matemática	25
3.1.5 Transmisión OFDM	25
3.1.5.1 Modulación por Desplazamiento de Fase	27
3.1.5.2 Modulación de amplitud en Cuadratura	27
3.2 Amplificadores de Potencia	28
3.2.1Traveling Wave Tube (TWT)	28
3.2.2Amplificador de Potencia de Estado Sólido (SSPA)	29
3.3 Redes de Kohonen	28
3.3.1 Redes de Aprendizaje no Supervisado	30
3.3.2 Reseña Historia de Redes Kohonen	30
3.3.3 Funcionamiento de Redes de Kohonen	31
3.3.4 Algoritmo de Kohonen	32
3.3.5 Memoria vector-cuantificada temporal asociativa	34
3.3.6 Descomposición QR	36
3.3.7 Utilizando Descomposición QR en Mínimos Cuadrados	38
CAPITULO 4: Sistema de Transmisión OFDM	39
4.1 Entrada (INPUT)	40
4.2 Modulador de amplitud en cuadratura de 16 estados (16QAM)	40
4.3 Modulador Banda Base	43
4.4 SOM, Red Neuronal	45
4.5 Amplificador de Potencia	48
4.6 Canal Gaussian	49
4.7 Demodulador Banda Base	50
4.8 Salida (OUTPUT)	52
CAPITULO 5: Evaluación de Resultados	53
CAPITULO 6: Conclusión	61
CAPITULO 7: Bibliografía	63
APENDICE: Código Fuente	66

Glosario de Términos

Banda Base: Se refiere a la banda de frecuencias producida por un dispositivo generador de señales, antes de sufrir modulación alguna.

Broadcasting: Es la distribución de señales transmitidos por un canal.

Demodulación: Es el proceso inverso a la modulación.

Distancia Euclídea: Es la distancia entre dos puntos del plano.

Intermodulación: es el resultado de la interacción entre dos o más señales en el interior de un dispositivo no lineal.

Klystron: Es un amplificador de válvula de vacío de electrones en la cual una modulación inicial de velocidad impartida a los electrones, da una modulación del haz de electrones.

Kohonen: Red neuronal no supervisada, competitiva, distribuida de forma regular en dimensiones, cuyo fin es descubrir la estructura subyacente de los datos introducidos en ella.

Modulación: Es una técnica para transportar información sobre una onda portadora con la intención transmitir más información en forma simultánea, protegiéndola de posibles interferencias y ruidos.

Ortogonalidad: Es cuando dos vectores dan un producto escalar igual a cero.

Radio Frecuencia: Es la porción menos energética del espectro electromagnético, situada entre unos 3 Hz y unos 300 GHz. Las ondas electromagnéticas de esta región del espectro se transmiten aplicando la corriente alterna originada en un generador a una antena.

Regresión: Es un término estadístico que modela una variable independiente Y con respecto a otras variable independiente X. El primer método de regresión es el Método de los mínimos Cuadrados creado en XIX.

Ruido Gaussian: Es el ruido cuya densidad de probabilidad responde a una distribución normal (o distribución de Gauss).

Lista de abreviaturas o Siglas

16QAM: Modulación de Amplitud en Cuadratura de 16 estados

64QAM: Modulación de Amplitud en Cuadratura de 64 estados

ART: Redes de Resonancia Adaptativa.

ASK: Modulación por desplazamiento de Amplitud

BER: Razón de Error de Bits.

BPSK: Modulación Binaria de Desplazamiento de Fase

BMU: Neurona Ganadora en SOM

EVM: Magnitud del Vector de Error

FFT: Transformada de Fourier.

HPA: Amplificador de Potencia.

ICI: Interferencia entre Subcarriers.

IFFT: Inversa de la Transformada de Fourier.

JAVA: Lenguaje de Programación de Sun Microsystems orientada a objetos

OFDM: Modulación Multiplexada por División de Frecuencia Ortogonal.

PSK: Modulación por Desplazamiento de Fase

QAM: Modulación de Amplitud en Cuadratura.

QPSK: Modulación por Cuadratura de Desplazamiento de Fase

RF: Radio Frecuencia.

RNA: Red Neuronal Artificial.

SER: Razón de Error de Símbolo.

SSPA: Amplificador de Potencia de Estado Sólido

SOM: Mapas Auto organizativos.

SNR: Relación Señal Ruido

TWTA: Amplificador de Tubo de Onda Progresiva

VQTAM: Memoria Vector-Cuantificada Temporal Asociativa

Índice de Ilustraciones

Figura 2.1. Evolución de los accesos a redes WIMAX en los últimos 2 años	5
Figura 2.2 Intermodulación de segundo orden	8
Figura 2.3. Productos de Intermodulación de tercer orden	9
Figura 2.4 Respuesta típica de un amplificador junto con sus parámetros de distorsión asociados	11
Figura 2.5 Filtros de Bandpass y LowPass frente a una señal de entrada	12
Figura 2.6 No linealidad en el dominio BandPass	12
Figura 2.7 No linealidad en el dominio Lowpass	13
Figura 2.8. Comparación de rendimiento BER v/s SNR en RTRL	18
Figura 2.9 BER vs SNR para 64QAM-64OFDM con canal AWGN, SSPA AM/AM	19
Figura 2.10. BER vs SNR para 16QAM-64OFDM con canal AWGN, SSPA AM/AM	20
Figura 3.1. Ejemplos del espectro de OFDM. a) Un sólo subcarrier, b) 5 portadores	21
Figura 3.2. a) Técnica multiportadora convencional, b) Modulación con portadoras ortogonales	23
Figura 3.3. Diagrama de bloque de un transmisor OFDM simple	24
Figura 3.4. Diagrama de un sistema basado en OFDM	26
Figura 3.5 Arquitectura típica de un mapa autoorganizativo de Kohonen	33
Figura 3.6 Teoría de VQTAM de forma grafica	34
Figura 4.1 Diagrama del Sistema de Transmisión OFDM	39
Figura 4.2 Ejemplo de 16QAM	41
Figura 4.3 Constelación Ideal de 16QAM según Formula matemática obtenida del sistema OFDM con SOM implementada	42
Figura 4.4 Constelación Ideal de 64QAM según Formula matemática obtenida del sistema OFDM con SOM implementada	43
Figura 4.5 Esquema de módulos del modulador OFDM	44
Figura 4.6 Ejemplo de Curva del Polinomio, Inversa y compuesta, la curva es una simulación realizada con el sistema implementado bajo 10000 datos de entrada	46
Figura 4.7 Amplitud versus Valores del Polinomio	49
Figura 4.8 Salida del demodulador sin amplificador	50
Figura 4.9 Salida del demodulador con amplificador	51
Figura 5.1 Magnitud del Vector de Error (EVM)	53

Figura 5.2 Constelación 16QAM de la salida del demodulador con un sistema que obtiene sus pesos de salida por medio de Mínimos Cuadrados	54
Figura 5.3 Constelación 16QAM de la salida del demodulador con un sistema que obtiene sus pesos de salida por medio del algoritmo no supervisado VQTAM	55
Figura 5.4 Constelación 16QAM para la salida del demodulador dada por OFDM + HPA + W	55
Figura 5.5 Curva BER vs SNR. Sistema sin Corrector Neuronal	56
Figura 5.6 Constelación 64QAM para la salida del demodulador dada por OFDM + HPA + W	56
Figura 5.7 Curva BER vs SNR. Sistema sin Corrector Neuronal	57
Figura 5.8 Curva BER vs SNR. Sistema con Corrector Neuronal	58
Figura 5.9 Curva BER vs SNR, ganancia obtenida frente a 10240 datos de entrada en 16QAM	59
Figura 5.10 Curva BER vs SNR. Sistema con Corrector Neuronal	59
Figura 5.11 Curva de Amplitud vs valor de los Polinomios, a) Polinomio del HPA, b) Compuesta entre a) y c), y c) Inversa de a)	60

Lista de Tablas

Tabla 4.1 Algoritmo utilizado para implementar 16QAM	41
Tabla 4.2 Algoritmo utilizado para implementar la IFFT	45
Tabla 4.3 Tabla de coeficientes de 5° orden	48
Tabla 5.1 Datos para obtener la Curva BER vs SNR de la Figura 5.8	58

Capítulo 1

Introducción

En la actualidad los sistemas de comunicación son parte importante en el desarrollo y diario vivir de la sociedad, por lo mismo, la demanda en servicios y calidad ha crecido exponencialmente en los últimos años. La tecnología de sistemas de comunicación de banda ancha han entrado con fuerza, y cada vez los usuarios requieren mayor disponibilidad y calidad de servicio, para lo cual los proveedores alimentan estas exigencias con tecnologías ADSL y Wi-Fi. Sin embargo, con cada nuevo acierto, surgen nuevos problemas, como interferencias, costos de instalación, etc.

Los sistemas de transmisión inalámbrica son por hoy las ansiadas soluciones de conectividad que requiere el mercado, un sistema de comunicación inalámbrico basado en OFDM (Modulación Multiplexada por División de Frecuencia Ortogonal) [CHAN01], de largo alcance, gran potencia, y alta velocidad, tecnología que cada vez gana más adeptos, empresas que quieren tener un sistema estable para toda su zona de trabajo, ya sean computadores, iPod o teléfonos móviles. Sin embargo, esta tecnología inalámbrica tiene sus desventajas, como la interferencia de señal, y la distorsión propia que se genera al usar la técnica de modulación multicarrier OFDM en la eficiencia de potencia, y que es propia de los amplificadores de potencia que utilizan modulación OFDM.

Y es ahí donde recae la importancia de estudios que ofrezcan a los desarrolladores de tecnologías nuevas alternativas para mejorar las técnicas y equipos que se construyen con el fin de lograr cada vez sistemas de información más estables y potentes.

Este documento establece las bases de conocimientos para poder implementar una técnica de linealización para esta distorsión. Son muchas las técnicas usadas, pero aquí se enfocará en la técnica de redes neuronales usando el mapa autoorganizativo de Kohonen [CAST01], también llamado Redes de Kohonen. Por lo tanto, en este escrito se conocerá de que tratan los temas fundamentales para poder reducir la distorsión no lineal en amplificadores de potencia, los cuales son: modulación OFDM, Redes de Kohonen, Distorsión no lineal y amplificadores de potencia.

Este proyecto trata sobre implementar una solución basada en las redes neuronales para mitigar la distorsión no lineal en los amplificadores de potencia de las antenas de los sistemas inalámbricos, la cual será modelada y evaluada por un sistema computacional desarrollado en JAVA. Se ha elegido las Redes de Kohonen por su calidad para clasificar información o el agrupamiento de patrones por tipo o clases, lo cual permitirá reconocer los bits que vengan dañados debido al proceso de amplificación de la señal.

1.1 Objetivo General

Desarrollar e implementar un sistema basado en mapas autoorganizativos para reducir la distorsión no lineal en sistemas de transmisión inalámbrica OFDM.

1.2 Objetivos Específicos

- Investigar las técnicas: Modulación QAM, OFDM, redes autoorganizativas y algoritmos de aprendizaje.
- Diseñar un esquema de linealización viable para reducir la distorsión no lineal basada en mapas autoorganizativos, usando algoritmos no supervisados y supervisados.
- Implementación computacional de los modelos sugeridos con 16QAM-OFDM y 64QAM-OFDM.
- Evaluar el rendimiento los modelos propuestos y contrastar ambos resultados.

1.3 Organización del texto

El resto de esta memoria de título se organiza de la siguiente manera:

- En el capítulo 2 se presenta el estado del arte de las técnicas utilizadas en transmisión inalámbrica tales como WIMAX, además se presenta la problemática de esta transmisión, y todos los tópicos relacionados con la distorsión no lineal, ya sea distorsión sobre amplitud, fase y la generada en los amplificadores. Finalmente, se presenta una descripción con técnicas utilizadas para reducir la distorsión no lineal.
- El capítulo 3 muestra un estudio del esquema de modulación OFDM, y de la modulación por desplazamiento de fase y de amplitud de cuadratura, también, se presentan una descripción de los modelos de amplificadores de potencia, y, por último, una completa revisión de las Redes Neuronales y de Kohonen, y la teoría detrás del algoritmo VQTAM.

- El capítulo 4 presenta la descripción del sistema OFDM planteado desde la entrada de datos hasta la salida, describiendo cada paso dentro del sistema, ya sea, los modulares QAM y de banda base para procesar los datos, la red neuronal que corrige los datos previniendo la distorsión generada en el amplificador, el canal gaussian y, finalmente, el demodulador que arroja los datos sin distorsión alguna.
- Por último, el capítulo 5, presenta la resolución de resultados del sistema OFDM presentado en el capítulo 4 e implementado en un sistema computacional construido en JAVA.

Estado del Arte

2.1 Estándares de Transmisión Inalámbrica

OFDM es base de muchos actuales sistemas de comunicación inalámbrica la cual varían en su forma de trabajo, sus protocolos, dispositivos, etc. los cuales han sido elaborados para poder aprovechar el ancho de banda al usar modulación multicarrier.

Todos estos estándares tienen la misma estructura física, una estación que recibe la señal (ya sea fija o móvil), una antena que emite la señal, y un amplificador de potencia de la señal. Entre los estándares mas conocidos que actualmente operan están:

- **Televisión Digital Terrestre (DVB-T o TDT):** es la aplicación de las tecnologías del medio digital a la transmisión de contenidos a través de una antena convencional (aérea) o de conexión por cable o satélite. Aplicando la tecnología digital se consiguen mayores posibilidades, como proveer de un mayor número de canales, mejor calidad de imagen o imagen en alta definición (HD o High Definition en inglés) y mejor calidad de sonido (empleando sistemas como AC3, Dolby Digital) [TRAN01].
- **WI-FI:** estándar de protocolo de comunicaciones del IEEE (802.11) que define el uso de los dos niveles más bajos de la arquitectura OSI (capas física y de enlace de datos), especificando sus normas de funcionamiento en una WLAN, la cual esta basada en OFDM [TRAN01].
- **WIMAX:** estándar de transmisión inalámbrica impulsada por el WIMAX Forum, el cual ocupa 256 portadoras, la cual puede ofrecer alcances de hasta 48 kilómetros [TRAN01].

Para comprender bien la operación de un sistema OFDM y la distorsión de los sistemas Inalámbricos, se profundizará en el estándar WIMAX con el objetivo de poder palpar mas claramente la problemática y la solución planteada.

2.1.1 WIMAX, Definición

WiMAX (del inglés **Worldwide Interoperability for Microwave Access**, Interoperabilidad

Mundial para Acceso por Microondas) es un estándar de transmisión inalámbrica de datos basada en modulación OFDM(802.MAN) proporcionando accesos concurrentes en áreas de hasta 48 kilómetros de radio y a velocidades de hasta 70 Mbps, utilizando tecnología que no requiere visión directa con las estaciones base. Este estándar es un conjunto de normas técnicas compartidas por distintos fabricantes y proveedores que aseguran la compatibilidad de sus productos, donde el nombre de su especificación técnica es IEEE 802.16-2004 + ETSI HyperMAN.

En la actualidad, existen dos tipos principales de tecnología WiMAX: WiMAX fija (802.16d-2004) y WiMAX móvil (802.16e-2005). La WiMAX fija es una tecnología de punto a varios puntos, en tanto que la WiMAX móvil es una tecnología de varios puntos a varios puntos, que se asemeja a la de una infraestructura celular. Estas dos soluciones fueron diseñadas para brindar servicios inalámbricos ubicuos de banda ancha de alto rendimiento a costos reducidos.

Los impulsores de WIMAX son un grupo de empresas conocido como el WiMax Forum. Fundado en junio de 2001, que ha saltado de 67 adherentes iniciales a más de 368, lo que le asegura una buena base de apoyo, en especial al contar entre sus filas con nombres tan prestigiados como Intel, Fujitsu, AT&T, Samsung o Siemens. El WiMAX Forum es un consorcio de empresas dedicadas a diseñar los parámetros y estándares de esta tecnología, y a estudiar, analizar y probar los desarrollos implementados. En principio se podría deducir que esta tecnología supone una grave amenaza para el negocio de tecnologías inalámbricas de acceso de corto alcance en que se basan muchas empresas, pero hay entidades muy importantes detrás del proyecto. Las principales firmas de telefonía móvil también están desarrollando terminales capaces de conectarse a estas nuevas redes. En efecto, ya se esta comenzando a ofrecer servicios de conexión a Internet a 4 Mbps, incorporando WiMAX a los computadores portátiles y PDA.

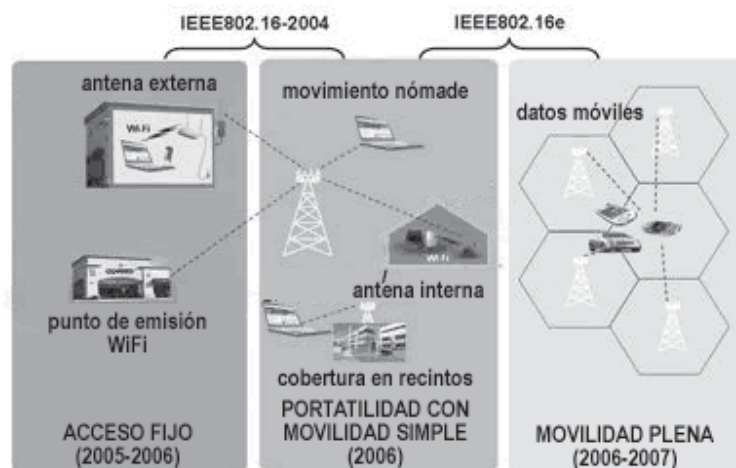


Figura 2.1. Evolución de los accesos a redes WIMAX en los últimos 2 años.

En Sudamérica, la primera red WIMAX se realizó en Argentina, en la ciudad de San

Miguel del Monte, donde la empresa bonarense MILICOM en conjunto con Intel implementó una antena en el corazón de esta ciudad y provee acceso inalámbrico de banda ancha a Internet a toda la localidad. Por otra parte, debido a la noticia de que los grandes productores de teléfonos móviles como Siemens, Samsung, Nokia, etc, grandes empresas proveedoras de telefonía móvil comienzan en el año 2005 ha implementar redes WIMAX complementando, en muchos casos, sus actuales redes Wi-Fi, es el caso de TELMEX que planea para comienzos del año 2008 tener cobertura WIMAX en 91% de las grandes ciudades.

2.1.2 Características de WIMAX

El estándar WIMAX estaba basado en modulación OFDM con 256 subportadoras. La definición de MAC (Control de Acceso al Medio) es muy potente y gestiona diferentes tipos de calidad del servicio (QoS) a la vez. La definición de PHY (Capa física) contiene un buen número de opciones y capacidades, entre los que se encuentran modulación y codificación adaptativa, diversidad STC, codificación MIMO y otras técnicas avanzadas. Aunque la 802.16 define una amplia tecnología, el WiMAX Forum ha seleccionado un subconjunto de esta norma, OFDM-256, PMP (punto multipunto), 3,5 GHz con anchos de banda 3,5 y 7 MHz o bandas de 5,8 GHz [ALON01].

Si bien para WIMAX Fijo (802.16d) se basa en OFDM con 256 subportadores, 802.16e utiliza OFDMA (Acceso Múltiple por División ortogonal de frecuencia) que esta basada en OFDM-256, OFDMA permite que muchas estaciones móviles accedan asignándole a cada uno un cierto número de subcarriers. También introduce la tecnología TDMA que asigna diferentes segmentos de tiempo a diferentes grupos de usuario. Todos los subcarriers OFDMA se dividen en diversos grupos de subcarriers en dominios de frecuencia, cada uno de los cuales se denomina subcanal. Un usuario puede ocupar uno o más subcanales. En ámbitos temporales, muchos equipos móviles en un segmento de tiempo dado pueden transmitir datos simultáneamente a través de diferentes subcanales. Asimismo, el OFDMA presenta un mecanismo de asignación flexible que hace posible asignar subcarriers dinámicamente dependiendo del tráfico, mientras distintos modos de modulación y potencias de transmisión son aplicados en diferentes subcarriers lo que resulta en niveles más altos de utilización de espectro. Las principales características de OFDMA son [ALON01]:

- Operación con TDMA y multiplexado OFDM.
- Bandas de frecuencia de 100 KHz para las señales OFDM como recurso básico de transmisión.
- Adaptación de velocidades de transmisión a partir de la asignación múltiple de bandas para la creación de señales de banda ancha.
- Diversidad en transmisión a través de la división de la información en diferentes

bandas sobre la misma portadora.

El principal punto débil de la implementación práctica del sistema esta en el diseño de los amplificadores para el enlace ascendente. Esto es debido a las variaciones de la envolvente de la señal, que complica la obtención de amplificadores lineales eficientes.

2.2 Distorsión no lineal

El estudio de los efectos de la distorsión no lineal y su compensación ha sido desde hace ya muchos años objeto de atención para investigadores de diversas áreas. Hoy, en particular, este estudio sigue siendo fundamental ya que se encuentra directamente implicado en el desarrollo de tecnologías de última generación en el área de las comunicaciones. Los nuevos sistemas de transmisión digital, en especial aquellos basados en OFDM, son capaces de ofrecer altos niveles de eficiencia espectral utilizando modulaciones lineales multinivel sobre un numeroso conjunto de subportadoras, que al ser ortogonales en frecuencia, pueden ser ubicadas en un ancho de banda muy reducido permitiendo así transmitir elevadas tasas de información por segundo y por ancho de banda. Sin embargo, y a consecuencia de esto, problemas como las interferencias por canal adyacente o la presencia de una distorsión no lineal en la cadena de transmisión afectan de manera crítica las prestaciones de estos sistemas imponiendo severos límites a su viabilidad [SIEG01].

En este contexto, el comportamiento no lineal de los amplificadores de alta potencia utilizados en transmisión de radio frecuencia, constituye el principal obstáculo para el buen funcionamiento de los sistemas de comunicación digital basados en OFDM.

Diversos estudios demuestran que los sistemas basados en OFDM presentan una altísima vulnerabilidad ante pérdidas de sincronismo y, en especial, ante la presencia de la distorsión no lineal típicamente introducida por los amplificadores de Radio Frecuencia. En el proceso de transmisión. Esto es debido a que la señal OFDM está caracterizada por una gran variabilidad en su envolvente, lo cual es comúnmente expresado como un alto nivel de PAPR (Peak-to-Average Power Ratio), parámetro que relaciona los valores máximo y promedio de la señal temporal. A consecuencia de esto, problemas como la interferencia por canal adyacente e intersimbólica afectan de manera crítica las prestaciones de estos sistemas. Por otra parte, la solución trivial de utilizar sólo parte del margen dinámico de los amplificadores (back-off), a fin de evitar zonas de distorsión y saturación, resulta especialmente inapropiada en el campo de las comunicaciones móviles, satelitales y, particularmente, en las nuevas aeroplataformas de radiocomunicación HAPS, donde la eficiencia de potencia en transmisión es primordial para lograr la máxima autonomía del equipamiento y de la nave de soporte.

Por lo tanto, el comportamiento no lineal de los amplificadores de alta potencia o HPAs (High Power Amplifiers), constituye uno de los principales obstáculos para el funcionamiento de los sistemas de comunicación digital basados en OFDM. Los HPA's introducen dos tipos de

distorsión no lineal AM/AM y AM/PM, sobre la amplitud y fase de la señal transmitida respectivamente. En las condiciones de ancho de banda de las señales comúnmente utilizadas en sistemas OFDM, ambos tipos de distorsión pueden ser caracterizados ya sea como funciones de transferencia o ganancia no lineal, sin memoria y dependientes de la amplitud de entrada al HPA.

2.2.1 Distorsión de Intermodulación

En los amplificadores de potencia se pueden generar dos distorsiones importantes, la distorsión de intermodulación, y la distorsión armónica, sin embargo, esta última distorsión no es relevante para este estudio, debido a que no influye en la señal OFDM (es ruido sonoro). La distorsión de intermodulación es el resultado de la interacción entre dos o más señales en el interior de un dispositivo no lineal, produciendo a su salida otras señales no deseadas. Estas señales adicionales que aparecen a la salida del dispositivo se conocen con el nombre de productos de intermodulación. Considerando que a la entrada del dispositivo no lineal tenemos dos tonos de radiofrecuencia f_1 y f_2 , a su salida se producirán productos de intermodulación a frecuencias suma y diferencia de múltiplos de las frecuencias originales, es decir:

$$f_{salida} = mf_1 \pm nf_2 \tag{2.1}$$

donde m y n son números enteros mayores o iguales que la unidad.

El orden de la distorsión no lineal queda determinado por la suma de m y n .

En la figura siguiente se representan las diferentes frecuencias de los productos de intermodulación de segundo y tercer orden que se generan a la salida de un dispositivo no lineal en el caso particular de que a su entrada se introduzcan dos tonos a frecuencias $f_1 = 100\text{MHz}$ y $f_2 = 110\text{MHz}$ [RAMO01].

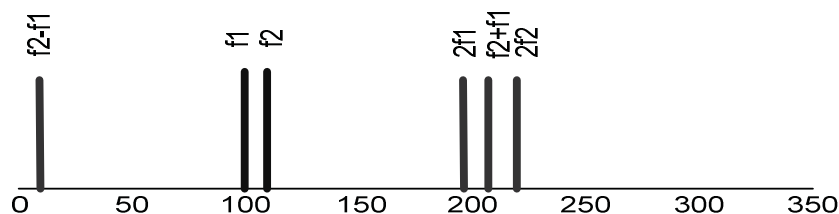


Figura 2.2 Intermodulación de segundo orden

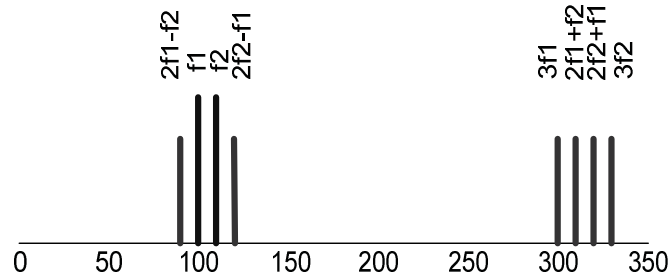


Figura 2.3. Productos de Intermodulación de tercer orden.

Como puede observarse en la figura 2.3, las distorsiones de segundo orden caen bastante alejadas de la banda de interés donde se encuentran las señales deseadas y, por lo tanto, generalmente no suelen causar problemas. En cambio, los productos de intermodulación de tercer orden del tipo $2f_1 - 2f_2 = 90\text{MHz}$ y $2f_2 - f_1 = 120\text{MHz}$ son los más perjudiciales, ya que se generan muy cerca de las señales de interés y son difíciles de filtrar. Todavía resulta peor el caso de una señal multicanal formada por diversas portadoras de radiofrecuencia, ya que la mayoría de los productos de intermodulación de este tipo caerán encima de los canales y serán imposibles de filtrar. Si además la banda de canales ocupa más de una octava de frecuencia, entonces ocurrirá lo mismo con la distorsión de segundo orden. Los restantes productos de intermodulación de tercer orden $2f_1 + f_2 = 310\text{MHz}$ y $2f_2 + f_1 = 320\text{MHz}$ no suelen representar un problema en la práctica [RAMO01].

En general, conforme aumenta el orden de la distorsión, el nivel de los productos de intermodulación a la salida del dispositivo disminuye. Sin embargo, los productos de intermodulación de tercer orden tienen una amplitud proporcional al cubo de la señal de entrada, mientras que los productos de segundo orden son proporcionales al cuadrado de la señal de entrada. De este modo, si se aumenta en 1 dB la potencia de cada una de las señales de entrada, los productos de intermodulación de segundo y tercer orden a la salida del dispositivo no lineal aumentarán en 2 y 3 dB, respectivamente. Por lo tanto, para potencias elevadas puede ser que la distorsión de tercer orden supere a la de segundo orden.

2.2.2 Distorsión sobre la Amplitud y la Fase

El comportamiento no lineal de un amplificador de potencia altera la forma original de la señal de entrada produciendo una señal de salida distorsionada. La respuesta no lineal de un amplificador de potencia se suele caracterizar mediante las funciones de conversión AM / AM (Amplitude Modulation / Amplitude Modulation) y AM / PM (Amplitude Modulation / Phase Modulation) según las cuales la modulación de amplitud de la señal de entrada produce, a su vez, una modulación modificada (no lineal) de amplitud y una modulación no lineal de fase, respectivamente, en la señal de salida.

El efecto conjunto de ambas funciones produce distorsión dentro y fuera de la banda de frecuencias de la señal original. Mientras la distorsión dentro de banda puede corromper la señal haciendo imposible la extracción de información, la distorsión fuera de banda puede hacer lo propio con otras señales ubicadas en las bandas adyacentes, lo cual, aparte de regulaciones y normativas, es particularmente crucial en los sistemas multiseñal.

En los sistemas multiportadoras, la no-linealidad del amplificador produce, además, intermodulación entre los diferentes tonos [RAMO01].

2.2.3 Distorsión no lineal en amplificadores

Los amplificadores están generalmente preparados para trabajar en un régimen de funcionamiento lineal, el cual se caracteriza por un aumento de la potencia de salida proporcional al incremento de la potencia de entrada. La diferencia en dB entre ambos niveles de potencia constituye la ganancia del amplificador. Sin embargo, conforme aumenta la potencia de señal a la entrada, llega un momento en que el amplificador comienza a saturarse y la potencia a la salida ya no aumenta proporcionalmente (comportamiento no lineal) [RAMO01].

En la figura 2.2 y 2.3 se puede observar el comportamiento anteriormente comentado y el siguiente. Dependiendo del tipo de amplificador con el que estemos trabajando, la saturación o el comportamiento no lineal surgirá para potencias más bajas o más altas (representado por la línea azul continua).

Para caracterizar las prestaciones de un amplificador en este sentido se utiliza un parámetro conocido como punto de saturación a 1 dB. El punto de saturación a 1 dB se define como aquel punto en que la potencia a la salida del amplificador se encuentra 1 dB por debajo del comportamiento lineal (ideal, graficado mediante la línea recta azul punteada), y puede referirse tanto a la entrada como a la salida del dispositivo.

En el caso de un amplificador de potencia utilizado en transmisión se tendrán puntos de saturación a 1 dB más elevados, mientras que en el caso de un amplificador de bajo ruido se tendrán valores más reducidos, ya que este último suele utilizarse en recepción donde los niveles de señal son más pequeños.

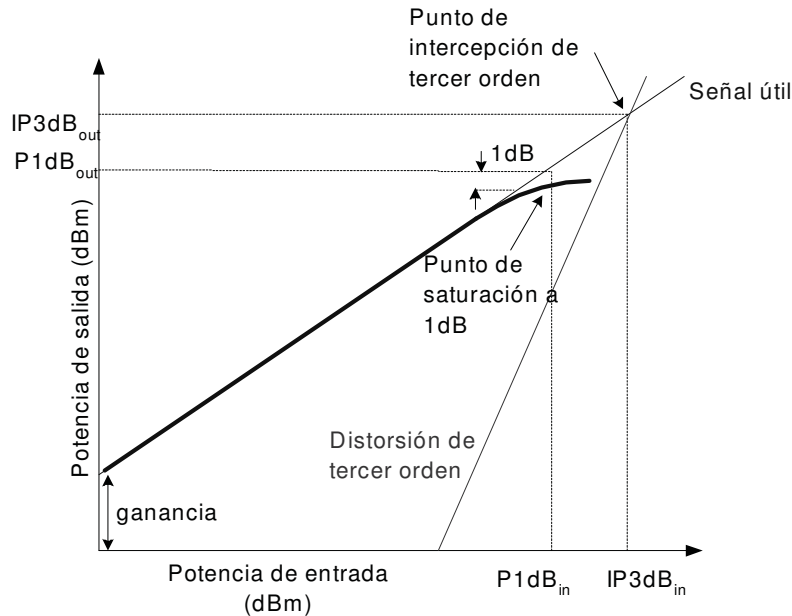


Figura 2.4 Respuesta típica de un amplificador junto con sus parámetros de distorsión asociados [RAMO01].

La saturación del amplificador implica que nos encontramos en una zona de trabajo donde la distorsión no lineal presenta valores importantes. Normalmente suele representarse en el mismo gráfico de potencia de salida frente a potencia de entrada el nivel de la distorsión, y más concretamente, el nivel de los productos de intermodulación de tercer orden $2f1 - f2$ ó $2f2 - f1$ que se ha visto que son los más perjudiciales. Estos productos de intermodulación presentarán una pendiente de orden 3 tal y como se representa en la figura 3.1.

Una manera de cuantificar la estimulación ejercida sobre la no-linealidad del amplificador consiste en medir la elongación de la envolvente de la señal de entrada mediante la relación valor peak / valor promedio (PAPR), de su potencia, expresada por:

$$PAPR = 10 \log(a_{max}^2 / a^2) dB \tag{2.2}$$

Donde,

a es la amplitud de la señal de entrada.

a_{max} es el máximo valor de amplitud de la señal de entrada.

Mientras mayor es el resultado de la relación mayor es la distorsión.

Para caracterizar la distorsión no lineal de un amplificador suele emplearse un parámetro conocido como punto de intercepción de tercer orden (IP3) y que se define como el punto donde se interceptan las rectas teóricas de potencia de señal útil y de potencia de intermodulación.

Al igual que antes, este parámetro puede referirse tanto a la entrada como a la salida del dispositivo. Evidentemente, cuanto más elevado sea el valor de IP3 más lineal será nuestro

dispositivo. Como regla práctica, el valor de IP3 suele estimarse 10 dB por encima del punto de saturación a 1 dB para sistemas que trabajan a frecuencias altas frente a 15 dB para sistemas que trabajan a frecuencias bajas [RAMO01].

2.2.4 Descripción Paso Banda

La distorsión no-lineal de las señales OFDM causan la pérdida de ortogonalidad de los subcarriers, y por consiguiente interferencia interportador. El resultado de esto es un aumento del SER (Razón de Error de Símbolo) cuando las señales de OFDM son descubiertas por un juego de filtros en el receptor. El paso banda (BandPass) deja pasar un determinado rango de frecuencias de una señal y atenúa el paso del resto. Estos filtros son complementados por Pasa Bajo (LowPass) los cuales se caracterizan por permitir el paso de frecuencias más bajas y atenuar las frecuencias más altas.

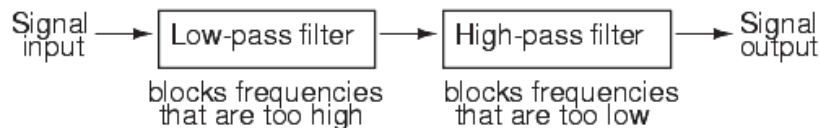


Figura 2.5 Filtros de Bandpass y LowPass frente a una señal de entrada

Una transmisión lineal de un sistema opera en frecuencias banda paso las que también pueden ser descritas en el equivalente dominio de Paso Bajo. Posiblemente un valor complejo de señal Bajo Paso puede ser denotado como LP relacionado para su señal banda paso BP a través de [SIEG02]:

$$S_{BP}(t) = R\{S_{SP}(t)e^{i2\pi f_0 t}\} \quad (2.3)$$

Donde $R\{\cdot\}$ denota la parte real, y f_0 es la frecuencia central del Bandpass.

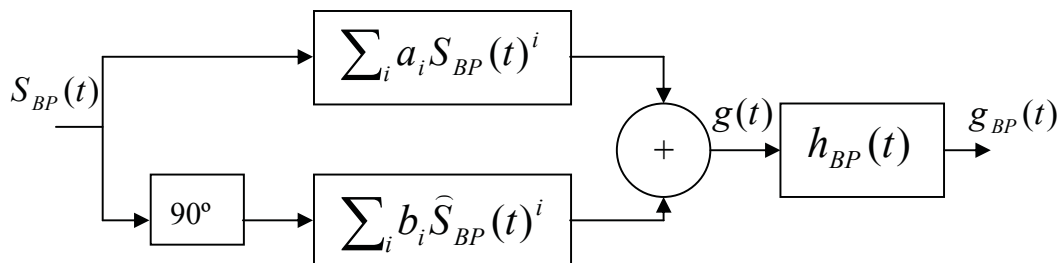


Figura 2.6 No linealidad en el dominio BandPass

Como se ve en la figura 3.5 $S_{BP}(t)$ es la entrada no linealizada, donde las características de

la Bandpass no linealizada, a veces también llamadas características instantáneas, están dadas por el polinomio.

$$g(t) = a_1 S_{BP}(t) + a_3 S_{BP}(t)^3 + a_5 S_{BP}(t)^5 + b_1 \widehat{S}_{BP}(t) + b_3 \widehat{S}_{BP}(t)^3 + b_5 \widehat{S}_{BP}(t)^5 \quad (2.4)$$

Donde $\widehat{S}_{BP}(t)$ es la transformada de Hilbert (rotación de fase de 90°) de $S_{BP}(t)$. Notar que $g(t)$ no es, estrictamente hablando, una señal Bandpass, entonces la operación del polinomio produce salidas espectrales no solo alrededor de la frecuencia f_0 , sino que también otras armónicas $3f_0$ o $5f_0$. Solo después de aplicar un filtro BandPass, $h_{BP}(t)$, llamada zona de filtros que elimina esas otras armónicas, se tiene una señal Banpass $g_{BP}(t)$.

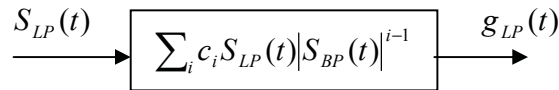


Figura 2.7 No linealidad en el dominio Lowpass

El equivalente a lo anteriormente enunciado, se refleja en la figura 3.6, entonces la descripción Lowpass es el polinomio [SIEG02]:

$$g_{LP}(t) = f(S_{LP}(t)) = c_1 S_{LP}(t) + c_3 S_{LP}(t) |S_{LP}(t)|^2 + c_5 S_{LP}(t) |S_{LP}(t)|^4 \quad (2.5)$$

Los coeficientes c_i del polinomio corresponden a los coeficientes reales de Bandpass a_i y b_i , dados por

$$c_i = \frac{1}{2^{i-1}} \binom{i}{(i-1)/2} (a_i + j b_i) \quad (2.6)$$

Si el amplificador no introduce modulación de fase en su señal de salida, entonces todos los b_i son cero y los c_i son valores reales. Por lo tanto, el poder de la señal de Bandpass es:

$$P_S = \overline{|S_{BP}(t)|^2} = \frac{1}{2} \overline{|S_{LP}(t)|^2} = \lim_{T \rightarrow \infty} \frac{1}{4T} \int_{-T}^T |S_{LP}(t)|^2 dt \quad (2.7)$$

donde la notacion sobrelinea significa que se hace un promedio sobre el tiempo.

2.3 Redes Neuronales

Las redes de neuronas (denominadas habitualmente como RNA) son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Se trata de un sistema de interconexión de neuronas en una red que colabora para producir un estímulo de salida.

Una red neuronal consiste en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, un computador o un conjunto de válvulas). El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz el cerebro que se caracterizan por su generalización y su robustez. Una red neuronal se compone de unidades llamadas neuronas. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Esta salida viene dada por tres funciones:

- Una función de propagación (también conocida como función de excitación), que por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina Excitatoria; si es negativo, se denomina Inhibitoria.
- Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida la misma función de propagación.
- Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que queramos darle a dichas salidas. Algunas de las más utilizadas son la Sigmoide (para obtener valores en el intervalo $[0,1]$) y la Hiperbólica-Tangente (para obtener valores en el intervalo $[-1,1]$).

Existe una serie de modelos que aparecen en la mayoría de estudios académicos y la bibliografía especializada.

- Perceptrón
- Adaline
- Perceptrón multicapa
- Memorias asociativas
- Máquina de Bolzman
- Máquina de Cauchy
- Redes de Elman
- Redes de Hopfield
- Red de contrapropagación
- Redes de neuronas de base radial
- Redes de neuronas de aprendizaje competitivo
- Redes de Kohonen o mapas autoorganizados

- Crecimiento dinámico de células
- Gas Neuronal Creciente
- Redes ART (Adaptative Resonance Theory)

Una primera clasificación de las redes de neuronas artificiales que se suele hacer es en función del patrón de conexiones que presenta.

- Dos tipos de redes de propagación hacia delante o acíclicas en las que todas las señales van desde la capa de entrada hacia la salida sin existir ciclos, ni conexiones entre neuronas de la misma capa.
 - Monocapa. Ejemplos: perceptrón, Adaline.
 - Multicapa. Ejemplos: perceptrón multicapa.
- Las redes recurrentes que presentan al menos un ciclo cerrado de activación neuronal. Ejemplos: Elman, Hopfield, máquina de Bolzman

Una segunda clasificación que se suele hacer es en función del tipo de aprendizaje de que es capaz (si necesita o no un conjunto de entrenamiento supervisado). Para cada tipo de aprendizaje encontramos varios modelos propuestos por diferentes autores:

- Aprendizaje supervisado: necesitan un conjunto de datos de entrada previamente clasificado o cuya respuesta objetivo se conoce. Ejemplos de este tipo de redes son: el perceptrón simple, la red Adaline, el perceptrón multicapa y la memoria asociativa bidireccional.
- Aprendizaje no supervisado o autoorganizado: no necesitan de tal conjunto previo. Ejemplos de este tipo de redes son: las memorias asociativas, las redes de Hopfield, la máquina de Bolzman y la máquina de Cauchy, las redes de aprendizaje competitivo, las redes de Kohonen o mapas autoorganizados y las redes de resonancia adaptativa (ART)
- Redes híbridas: son un enfoque mixto en el que se utiliza una función de mejora para facilitar la convergencia. Un ejemplo de este último tipo son las redes de base radial.

Finalmente también se pueden clasificar las RNA según sean capaces de procesar información de distinto tipo en:

- Redes analógicas: procesan datos de entrada con valores continuos y, habitualmente, acotados. Ejemplos de este tipo de redes son: Hopfield, Kohonen y las redes de aprendizaje competitivo.
- Redes discretas: procesan datos de entrada de naturaleza discreta; habitualmente valores lógicos booleanos. Ejemplos de este segundo tipo de redes son: las máquinas de Bolzman y Cauchy, y la red discreta de Hopfield.

2.3.1 Características de las Redes Neuronales

Al igual que el cerebro, una RNA se compone de un conjunto masivamente paralelo de unidades de proceso muy simples y es en las conexiones entre estas unidades donde reside la inteligencia de la red. Sin embargo, en términos de escala, un cerebro es muchísimo mayor que cualquier RNA creada hasta la actualidad, y las neuronas artificiales también son más simples que su contrapartida animal. Biológicamente, un cerebro aprende mediante la reorganización de las conexiones sinápticas entre las neuronas que lo componen. De la misma manera, las RNA tienen un gran número de procesadores virtuales interconectados que de forma simplificada simulan la funcionalidad de las neuronas biológicas. En esta simulación, la reorganización de las conexiones sinápticas biológicas se modela mediante un mecanismo de pesos, que son ajustados durante la fase de aprendizaje. En una RNA entrenada, el conjunto de los pesos determina el conocimiento de esa RNA y tiene la propiedad de resolver el problema para el que la RNA ha sido entrenada [FREE01].

Por otra parte, en una RNA, además de los pesos y las conexiones, cada neurona tiene asociada una función matemática denominada función de transferencia. Dicha función genera la señal de salida de la neurona a partir de las señales de entrada. La entrada de la función es la suma de todas las señales de entrada por el peso asociado a la conexión de entrada de la señal. Algunos ejemplos de funciones de transferencia son la función escalón, la lineal o mixta, la sigmoideal y la función gaussiana.

Las Redes de Neuronas Artificiales (RNA) tienen muchas ventajas al momento de crear maquinas inteligentes debido a que están basadas en la estructura del sistema nervioso, principalmente el cerebro, creando una acción y reacción, que se logra mediante un periodo de entrenamiento.

- **Aprendizaje:** Las RNA tienen la habilidad de aprender mediante una etapa que se llama etapa de aprendizaje. Esta consiste en proporcionar a la RNA datos como entrada a su vez que se le indica cuál es la salida (respuesta) esperada.
- **Auto organización:** Una RNA crea su propia representación de la información en su interior, descargando al usuario de esto.
- **Tolerancia a fallos.** Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo aceptablemente aún si se daña parcialmente.
- **Flexibilidad:** Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (ej. si la información de entrada es la imagen de un objeto, la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente).
- **Tiempo real:** La estructura de una RNA es paralela, por lo cuál si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real

2.3.2 Descripción Histórica de las Redes Neuronales

En 1936 Alan Turing fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación. Sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas. Ellos modelaron una red neuronal simple mediante circuitos eléctricos. Posteriormente en 1949 Donald Hebb fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría. Aún hoy, este es el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal. Su idea fue que el aprendizaje ocurría cuando ciertos cambios en una neurona eran activados. También intentó encontrar semejanzas entre el aprendizaje y la actividad nerviosa. Los trabajos de Hebb formaron las bases de la Teoría de las Redes Neuronales. Ya en la década de los 50' Karl Lashley en su serie de ensayos, encontró que la información no era almacenada en forma centralizada en el cerebro sino que era distribuida encima de él [CAST01].

Pero todos estos hallazgos no podían quedar en el anonimato, es por eso que en 1956 se celebró el Congreso de Dartmouth, dando paso al nacimiento de la inteligencia artificial.

Estando aún en los años 50, Frank Rosenblatt en 1957, comenzó el desarrollo del Perceptron. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como identificador de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado en el entrenamiento. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente. Luego en 1959 el mismo Frank Rosenblatt escribió: "Principios de Neurodinámica", en este libro confirmó que, bajo ciertas condiciones, el aprendizaje del Perceptron convergía hacia un estado finito (Teorema de Convergencia del Perceptron).

En 1960 Bernard Widroff y Marcian Hoff desarrollaron el modelo Adaline (ADaptative LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas. Un año más tarde, en 1961, Karl Steinbeck realiza: "Die Lernmatrix", una red neuronal para simples realizaciones técnicas (memoria asociativa).

Ocho años después, en 1969 casi se produjo la "muerte abrupta" de las Redes Neuronales, ya que Marvin Minsky y Seymour Papert probaron (matemáticamente) que el Perceptron no era capaz de resolver problemas relativamente fáciles, tales como el aprendizaje de una función no-lineal. Esto demostró que el Perceptron era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real. Pero en 1974 Paul Werbos desarrolló la idea básica del algoritmo de aprendizaje de propagación hacia atrás

(backpropagation), cuyo significado quedó definitivamente aclarado en 1985.

En 1977 Stephen Grossberg desarrolla la “Teoría de Resonancia Adaptada (TRA)”. Esta Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo.

Pasaron casi diez años, para que en 1985 John Hopfield provocara el renacimiento de las redes neuronales con su libro: “Computación neuronal de decisiones en problemas de optimización”. Posteriormente en 1986 David Rumelhart y G. Hinton redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation).

2.4 Técnicas para Reducir la Distorsión no Lineal

La solución propuesta en este documento para mitigar el efecto de la distorsión no lineal en sistemas inalámbricos no es la única, en efecto existen numerosos estudios en diferentes partes del globo que plantean soluciones a este problema, muchos de los cuales son actualmente la solución implementada por las organizaciones.

Es el caso de compensadores neuronales recurrentes con el algoritmo Real Time Recurrent Learning (RTRL), propuesto por Williams/Zipser en el año 1989, el cual computa las derivaciones de estados y rendimientos con respecto a todos los pesos como la secuencia de procesos de la red.

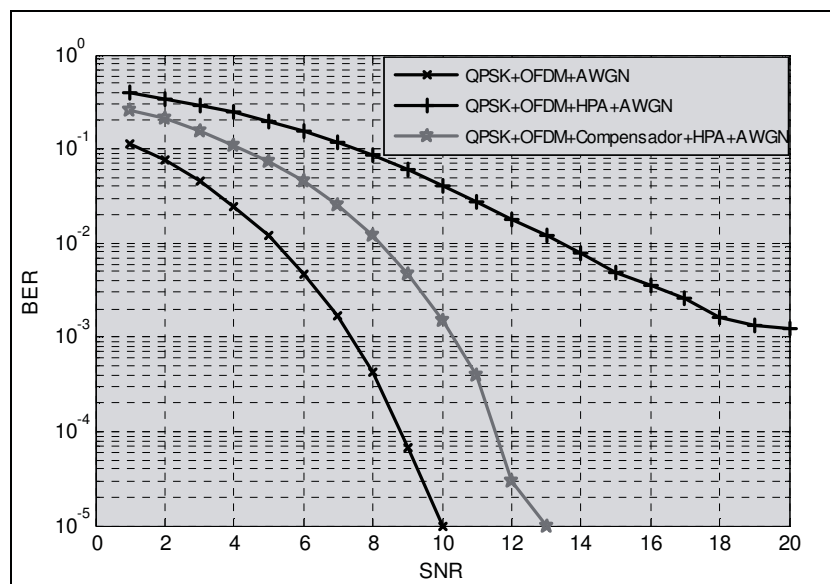


Figura 2.8. Comparación de rendimiento BER v/s SNR en RTRL

Según la Figura 2.8 de [ROJA01] es posible obtener una ganancia de 6db, para un sistema en

simulación computacional con 100240 bits de entrada y con 1024 canales.

Esta alternativa de solución muestra un enorme grado de satisfacción y reduce notablemente la distorsión no lineal generada por un amplificador de potencia en un sistema OFDM que para este caso se implemento con QPSK.

Otra alternativa que se puede destacar para reducir la distorsión no lineal es La optimización por enjambre de partículas (PSO) perteneciente a las técnicas estocásticas de cálculo evolutivo, considerada unas de las representantes de la rama de la inteligencia de enjambre (Swarm Intelligence).

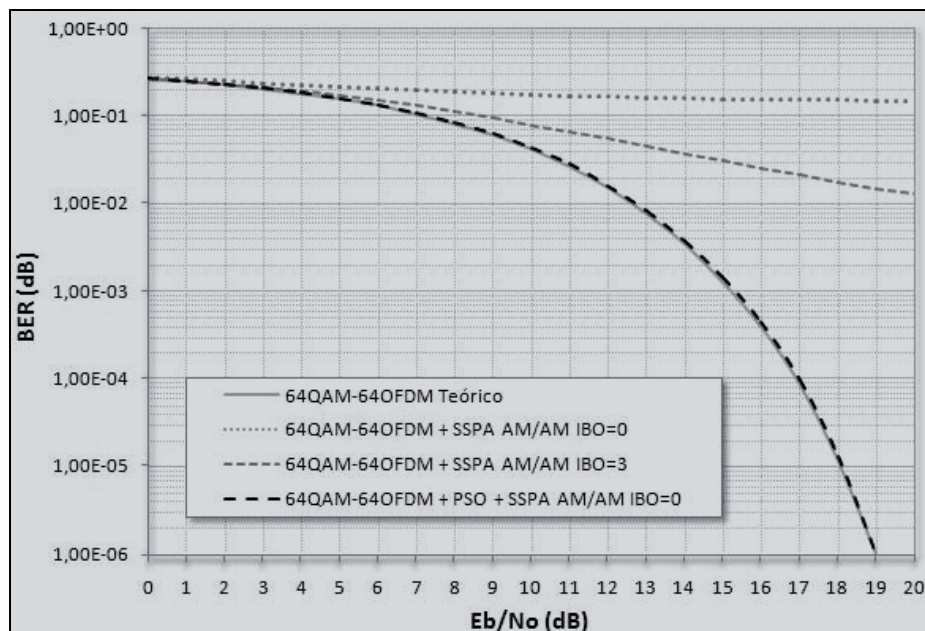


Figura 2.9 BER vs SNR para 64QAM-64OFDM con canal AWGN, SSPA AM/AM

En las figuras 2.9 y 2.10 [DURA01] se presenta el rendimiento del BER vs SNR para los esquemas 16QAM-64OFDM y 64QAM-64OFDM respectivamente, utilizando el modelo de 3er orden para SSPA con distorsión de amplitud, con y sin compensador PSO (CPSO). Para el esquema 16QAM el sistema alcanza una reducción de E_b/N_0 de 5.5 dB cuando se ocupa CPSO. En el caso de 64QAM se obtiene una reducción de E_b/N_0 7.5 dB cuando se ocupa CPSO.

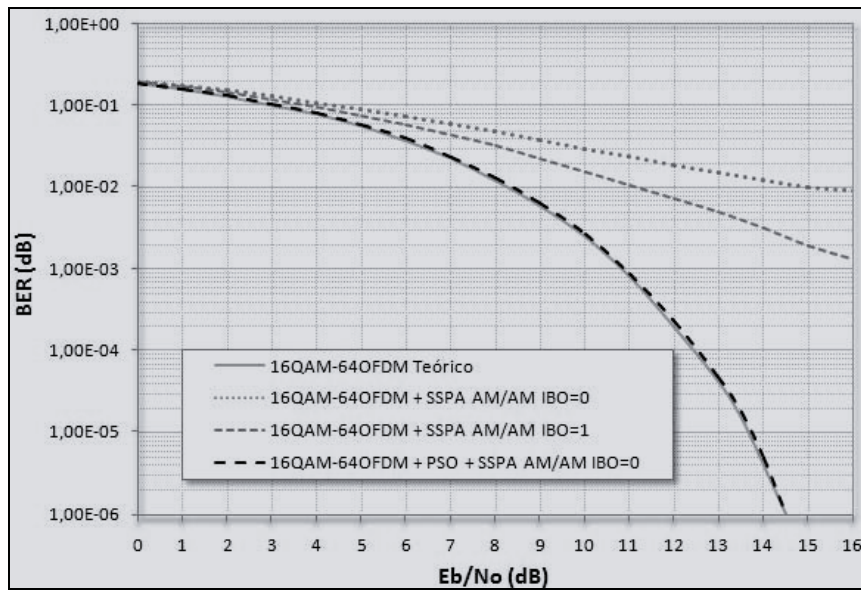


Figura 2.10. BER vs SNR para 16QAM-64OFDM con canal AWGN, SSPA AM/AM

Para la simulación computacional de PSO se utilizaron 125 muestras con una entrada de 102.400 bits para 16QAM y 153.600 bits para 64QAM distribuidos de forma uniforme, en un sistema OFDM de 64 canales.

Capítulo 3

Modulación Multicarrier OFDM, Amplificadores de Potencia, y Redes de Kohonen

3.1 Modulación Multicarrier OFDM

En telecomunicación el término modulación engloba el conjunto de técnicas para transportar información sobre una onda portadora, típicamente una onda senoidal (señal análoga). Estas técnicas permiten un mejor aprovechamiento del canal de comunicación lo que posibilita transmitir más información en forma simultánea, protegiéndola de posibles interferencias y ruidos. Básicamente, la modulación consiste en hacer que un parámetro de la onda portadora cambie de valor de acuerdo con las variaciones de la señal moduladora, que es la información que se quiere transmitir.

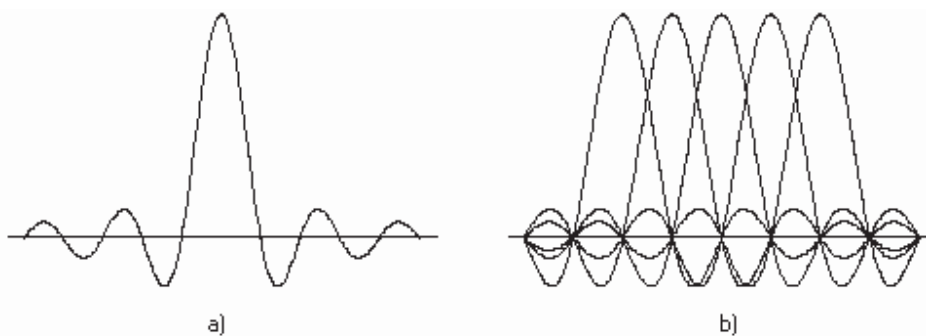


Figura 3.1. Ejemplos del espectro de OFDM. a) Un sólo subcarrier, b) 5 portadores.

La modulación por división ortogonal de frecuencia, en inglés Orthogonal Frequency Division Multiplexing (OFDM), también llamada modulación por multitono discreto es una modulación que consiste en enviar la información modulando en QAM (modulación de amplitud en cuadratura) o en PSK (Modulación por desplazamiento de fase) un conjunto de portadoras de diferente frecuencia [LUE01].

Normalmente se realiza la modulación OFDM tras pasar la señal por un codificador de canal con el objetivo de corregir los errores producidos en la transmisión, entonces esta modulación se

denomina COFDM del inglés Coded OFDM. Debido al problema técnico que supone la generación y la detección en tiempo continuo de los cientos, o incluso miles, de portadoras equiespaciadas que forman una modulación OFDM, los procesos de modulación y demodulación se realizan en tiempo discreto mediante la utilización de la función Transformada de Fourier.

3.1.1 Características de la modulación OFDM

La modulación OFDM es muy robusta frente al multicamino, que es muy habitual en los canales de radiodifusión, frente a los desvanecimientos selectivos en frecuencia y frente a las interferencias de radiofrecuencia. Debido a las características de esta modulación, las distintas señales con distintos retardos y amplitudes que llegan al receptor contribuyen positivamente a la recepción, por lo que existe la posibilidad de crear redes de radiodifusión de frecuencia única sin que existan problemas de interferencia.

Entre los sistemas que usan la modulación OFDM destacan:

- La televisión digital terrestre DVB-T, también conocida como TDT.
- La radio digital DAB.
- La radio digital de baja frecuencia DRM.
- El protocolo de enlace ADSL.
- El protocolo de red de área local IEEE 802.11a/g, también conocido como Wireless LAN (WLAN).
- El sistema de transmisión inalámbrica de datos WiMAX (802.16e)

3.1.2 Descripción Histórica de Modulación OFDM

Los orígenes de OFDM datan de finales de los años 1950 cuando se introdujo la Multiplexación por División de Frecuencia (FDM). En 1966 el Sr. Chang patentó la estructura OFDM y publicó el concepto de señales multi-tono con solapamiento ortogonal. Las señales ortogonales son aquellas que son mutuamente independientes unas de otras. La ortogonalidad permite que múltiples señales sean transmitidas sobre el mismo canal sin que se interfieran entre sí. En 1971 el Sr. Weinstein [10] introdujo la idea de usar la Transformada discreta de Fourier (DFT) para implementar la generación y recepción de señales OFDM. Luego, con el uso de la Transformada Rápida de Fourier (FFT), que es más eficiente que DFT, se logró la implementación de OFDM más fácil aún. Esto sugirió que una implementación más fácil de

OFDM se logra con el Procesamiento de Señales Digitales (DSP), el que puede implementar algoritmos FFT. Sólo recientemente el uso de tecnologías con circuitos integrados ha hecho que la implementación de OFDM sea costo-efectiva. La poca confianza en DSP frenó el avance de OFDM durante sus primeros años de desarrollo.

Sólo después de los años 1980 OFDM se comenzó a desarrollar comercialmente con la introducción de sistemas Digital Audio Broadcasting (DAB). Posteriormente OFDM se siguió implementando en otros sistemas, como por ejemplo, Digital Video Broadcasting (DVB) iniciado en 1993.

3.1.3 ¿Cómo funciona OFDM?

La idea de OFDM es dividir el ancho de banda disponible en varios subcarriers o subportadoras. Todos los subcarriers de OFDM son sincronizados en tiempo y frecuencia, para controlar la interferencia entre subcarriers (ICI) y mantener la ortogonalidad de la modulación. Cada uno de los subcarriers se modula con un esquema de modulación arbitrario (por ejemplo BPSK, QPSK, M-QAM, etc.) [LUNA01].

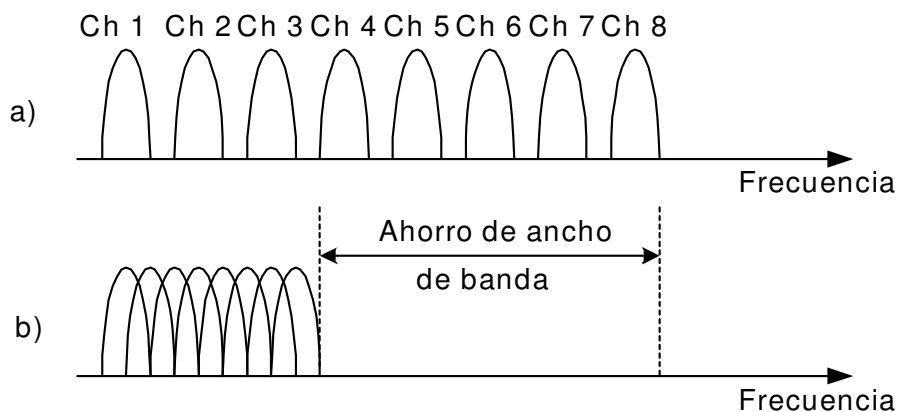


Figura 3.2. a) Técnica multiportadora convencional, b) Modulación con portadoras ortogonales.

OFDM logra la ortogonalidad localizando cada señal en un subcarrier diferente. Las señales OFDM son una suma de sinusoides, cada una correspondiente a un subcarrier (Figura 2.1). La frecuencia base de cada subcarrier es un múltiplo entero del inverso del tiempo del símbolo, dando origen a que todos los subcarriers tiene un número entero de ciclos por símbolo. Otra forma de ver la ortogonalidad en OFDM es a través de su espectro. En el dominio de la frecuencia OFDM cada subcarrier tiene una respuesta de frecuencia. Esto es un resultado del tiempo del símbolo correspondiente al inverso del espacio del carrier. El espacio del carrier está

determinado por la tasa de muestreo y por el tamaño de FFT usado para analizar la señal.

Las señales digitales OFDM son convertidas por el transmisor para ser enviadas, el que mapea la amplitud y fase de la señal. Luego la señal se transforma a un dominio de tiempo usando IFFT. Para transmitir la señal OFDM, el dominio de tiempo de la señal calculado se mezcla para la frecuencia requerida. Una vez que la señal ha sido transformada, se le inserta el período de guarda y el frame de sincronización.

El período de guarda está formado por una copia de la última parte del símbolo OFDM activo o símbolo útil y se inserta como prefijo del símbolo activo. El intervalo de guarda también se conoce con el nombre de prefijo cíclico (CP). Los sistemas de comunicación que usan intervalo de guarda generan una pérdida de eficiencia espectral proporcional al tamaño de la longitud del período de guarda. El tamaño del intervalo de guarda se recomienda que sea mayor a la longitud de la respuesta de impulso del canal de transmisión para mantener la ortogonalidad de los portadores y eliminar el efecto negativo de interferencia entre símbolos (ISI) e interferencia entre subcarriers (ICI). El efecto ISI se elimina debido a la incorporación de la banda de guarda entre los símbolos OFDM e ICI se elimina porque el prefijo cíclico mantiene la ortogonalidad de los portadores. Finalmente la señal se modula y se envía (Figura 3.2).

El receptor debe desarrollar la operación inversa al transmisor, demodulando la señal recibida y extrayendo el frame de sincronización y período de guarda. Luego, y con el uso de FFT, se analiza la señal en el dominio de la frecuencia. Una vez obtenida la fase y la amplitud del subcarrier, la señal se convierte en un dato digital.

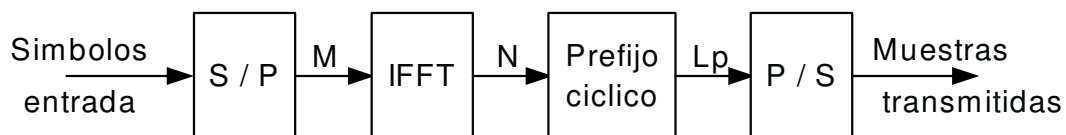


Figura 3.3. Diagrama de bloque de un transmisor OFDM simple.

La señal a transmitir normalmente es un flujo de datos seriales. En OFDM, cada símbolo normalmente transmite entre 40 y 4000 bits, por lo que la señal serial debe ser convertida a una señal paralela para poder ser transmitida en símbolos OFDM. Los datos de cada símbolo dependen del esquema de modulación usado y del número de subcarriers. Por ejemplo, para el esquema de modulación de subcarriers 16-QAM, cada subcarrier contiene 4 bits de datos, por lo que si se usan 100 subcarriers el número de bits por símbolo será de 400. Los subcarriers no usados son seteados a cero.

La ventaja de esta técnica es su eficiencia para compensar la distorsión de interferencia

entre símbolos (ISI) y la interferencia entre subcarriers (ICI), ya que al ser las señales ortogonales, es decir, independientes unas de las otras, permite que múltiples señales sean transmitidas por el mismo canal sin que se interfieran entre sí [CHAN01].

La desventaja se denomina “razón de potencia peak versus potencia promedio (PAPR)”, es decir, cuando el valor de PAPR es grande provoca que se pierda la ortogonalidad de las portadoras lo cual implica que la señal recibida sea con un alto grado de interferencia entre símbolos e interferencia entre subcarriers. Causando además sensibilidad a la no-linealidad de un amplificador de alto poder de un transmisor.

3.1.4 Representación matemática

OFDM transmite un gran número de portadores, estrechamente espaciados en el dominio de frecuencia. Para evitar un gran número de modulador-demoduladores, filtros al transmisor, filtros complementarios y demoduladores al receptor, es deseable poder usar modernas técnicas de proceso de señales digitales, como la Transformada Rápida de Fourier (FFT). Matemáticamente, cada portador puede describirse como una onda compleja:

$$S_c(t) = A_c(t) \exp(j[\omega_c(t) + \Phi_c(t)]) \quad (3.1)$$

La señal real es la parte real de $S_c(t)$. Ambos $A_c(t)$ y $\Phi_c(t)$, la amplitud y fase del portador, puede variar en un símbolo por símbolo base. Los valores de los parámetros son constantes por sobre el período de duración de símbolo t . OFDM consiste en muchos portadores. Así la señal compleja se representa por:

$$S_s(t) = \frac{1}{N} \sum_{n=0}^{N-1} A_n(t) \exp(j[\omega_n(t) + \Phi_n(t)]) \quad (3.2)$$

Donde, $\omega_n = \omega_0 + n\Delta\omega$

3.1.5 Transmisión OFDM

Un carrier de señal OFDM es la suma de numero de subcarriers ortogonales (cada tono es ortogonal con cada otro tono), con la información de la banda base en cada subcarriers comúnmente moduladas independientemente usando algún tipo de Modulación de amplitud en Cuadratura (QAM) o Modulación por desplazamiento de Fase (PSK).

La información es modulada en un tono ajustando la fase del tono, la amplitud, o ambos. En la forma más básica, un tono puede estar presente o no indicado con un 1 ó 0 en el bit de información, sin embargo, típicamente se emplea (para la modulación banda base) la modulación de amplitud en cuadratura (QAM).

Un sistema OFDM toma una corriente de datos y la divide en N corrientes de datos paralelos, cada uno en un valor de 1/N del valor original y con una duración del símbolo de TS. Cada corriente entonces es trazada en un mapa de a un tono en una frecuencia única y combinada usando la Transformada de Fourier Rápida Inversa (IFFT) para permitir la forma de onda de dominio de tiempo a ser transmitida.

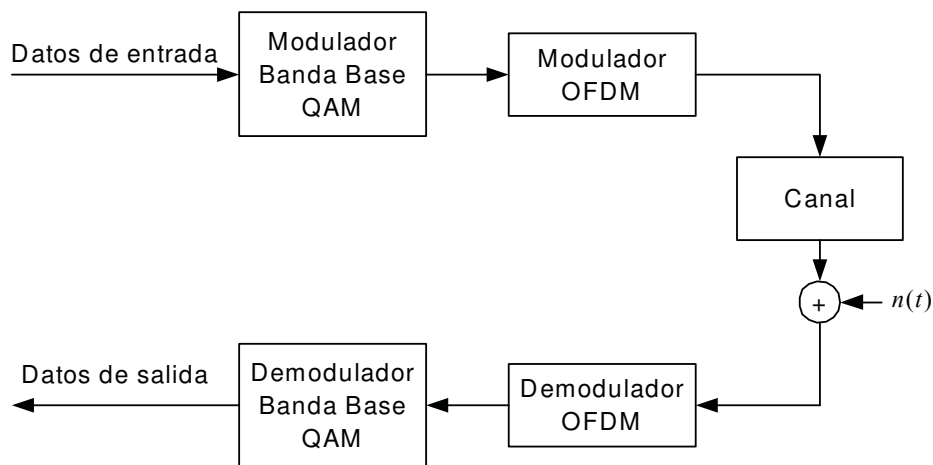


Figura 3.4. Diagrama de un sistema basado en OFDM.

Como se nota en la figura anterior cada señal modulada es procesada por un modulador y demodulador OFDM, por ejemplo, si se usara un sistema de 100 tonos, una sola corriente de datos con un valor de 1 Megabyte por segundo (Mbps) sería convertida en 100 corrientes de 10 kilobytes por segundo (kbps). Creando pequeñas corrientes de datos paralelos, la eficacia del ancho de banda del símbolo de modulación es disminuida en un factor de 100, o, equivalentemente, la duración del símbolo de modulación es aumentada en un factor de 100.

La selección apropiada de parámetros del sistema, como el número de tonos y el espaciado de tono, puede reducir, o aún eliminar, la distorsión de interferencia entre símbolos (ISI), porque la extensión típica de retraso de multicamino representa una muy pequeña proporción del tiempo de símbolo alargado. Visto de otra forma, la coherencia de amplitud de banda del canal puede ser mucho más pequeña, porque se ha reducido la amplitud de banda del símbolo.

3.1.5.1 Modulación por Desplazamiento de Fase

La modulación por desplazamiento de fase o PSK (Phase Shift Keying) es una forma de modulación angular consistente en hacer variar la fase de la portadora entre un número de valores discretos. La diferencia con la modulación de fase convencional (PM) es que mientras en ésta la variación de fase es continua, en función de la señal moduladora, en la PSK la señal moduladora es una señal digital y, por tanto, con un número de estados limitado.

Dependiendo del número de posibles fases a tomar, recibe diferentes denominaciones. Dado que lo más común es codificar un número entero de bits por cada símbolo, el número de fases a tomar es una potencia de dos. Así tendremos BPSK con 2 fases (equivalente a PAM), QPSK con 4 fases (equivalente a QAM), 8-PSK con 8 fases y así sucesivamente. A mayor número de posibles fases, mayor es la cantidad de información que se puede transmitir utilizando el mismo ancho de banda, pero mayor es también su sensibilidad frente a ruidos e interferencias.

Las modulaciones BPSK y QPSK son óptimas desde el punto de vista de protección frente a errores. Conceptualmente hablando, la diferencia entre distintos símbolos (asociados a cada fase) es máxima para la potencia y ancho de banda utilizados.

3.1.5.2 Modulación de amplitud en Cuadratura

La modulación QAM consiste en modular en Modulación por desplazamiento de Amplitud (ASK) de forma independiente, dos portadores que tienen la misma frecuencia pero que están desfasadas entre sí 90°. La señal modulada QAM es el resultado de sumar ambas señales ASK. Estas pueden operar por el mismo canal sin interferencia mutua porque sus portadoras están en cuadratura.

La fórmula matemática de una señal modulada en QAM es:

$$a \cos wt + b \sin wt \quad (3.3)$$

Las amplitudes de las dos señales moduladas en ASK (a y b), toman de forma independiente los valores discretos a_n y b_n correspondientes al total de los "N" estados de la señal moduladora codificada en banda base multinivel $N = n \times m$.

3.2 Amplificadores de Potencia

Los dispositivos comerciales de amplificación de potencia en RF utilizan comúnmente dos tecnologías diferentes. Por una parte, están los HPA basados en tubos de ondas progresivas (TWTA) y, por otra, aquellos basados en dispositivos de estado sólido (SSPA). La ventaja principal de los últimos, en términos de comportamiento no lineal, es que introducen un nivel de distorsión AM/PM muy inferior. Sin embargo, aspectos como la respuesta de frecuencia y los retardos electrónicos (memoria) aun mantienen a los TWT (traveling-wave tubes) en una posición predominante en aplicaciones de alta frecuencia, especialmente en sistemas de microondas.

3.2.1 Traveling Wave Tube (TWT)

El tubo de onda progresiva TWT (Traveling Wave Tube) [TRAN01] es una de las variantes de amplificador que se utiliza en las estaciones para comunicaciones satelitales. El TWT es un amplificador de gran ancho de banda (hasta una octava) y una ganancia de potencia de 25 a 50 dB. La eficiencia, entre el 20 y 40%, es función del ancho de banda. Este amplificador consiste en un generador de haz electrónico y una estructura de enfoque magnético. Una estructura en forma de hélice facilita la interacción entre el campo de microondas y el haz electrónico. La velocidad de los electrones se ajusta para que sea igual a la velocidad de fase de las microondas.

Los TWT típicos en general funcionan con una tensión de colector inferior al ánodo y cátodo. La amplificación propiamente dicha se produce en la estructura de enfoque e interacción. En la medida que la onda a amplificar viaja en la estructura de hélice el campo electromagnético modula la velocidad de los electrones en ondas periódicas aproximadamente en fase con el campo. La mayoría de los electrones desaceleran y entregan energía al campo produciendo la amplificación. Se caracteriza a este proceso por una ganancia proporcional a la longitud de la zona de interacción. La estructura de onda lenta es una hélice de alambre de tungsteno o molibdeno sujeta a una varilla de cerámica (óxido de aluminio o berilio) que la aíslan de la estructura metálica envolvente. La selección de materiales influye en la capacidad de potencia de salida y la eficiencia del TWT.

En las estaciones de comunicaciones por satélite se recurre al Klystron o al TWT debido a las exigencias de potencia de emisión. En cambio, en los enlaces terrestres se recurre exclusivamente a amplificadores de potencia de estado sólido con transistores. En muy pocos casos se utilizó en el pasado TWT para obtener ganancias adicionales cercanas a 5 dB. El TWT tiene prestaciones inferiores al Klystron en cuanto hace a la linealidad de fase. El nivel de ruido es menor en el TWT, -64 dBm/kHz con respecto a -58 dBm/kHz en el Klystron. En los TWT se trabaja con un back-off (IBO) de 7 dB para reducir los productos de intermodulación y muchas veces con linealizadores o predistorsionadores.

Los amplificadores de tubo permiten producir grandes potencias. La elección entre ambos

tipos depende del ancho de banda requerido y del coste (son más baratos los Klystron). Los amplificadores de potencia son dispositivos no lineales, de manera que a medida que la potencia de la señal de entrada aumenta, hay una región de funcionamiento casi-lineal a partir de la cual se alcanza un punto de saturación en que la salida ya no sigue a la entrada de forma lineal. La máxima potencia a la salida se corresponde con el punto de saturación.

3.2.2 Amplificador de Potencia de Estado Sólido (SSPA)

Los amplificadores más interesantes por la relación entre el costo, consumo, tamaño, reproductividad y distorsiones son los realizados mediante transistores SSPA (Solid State Power Amplifier) [TRAN01].

En los amplificadores de potencia de estado sólido el nivel máximo de potencia de salida es de 10 watts en las bandas de 4/6 GHz y de 2,5 w en 11/14 GHz. Tienen por ello una potencia de salida limitada frente a los amplificadores tradicionales usados en estaciones terrenas.

Los SSPA tienen una disipación mayor que obliga a ocupar un volumen físico también mayor, consumiendo más potencia que los enlaces radioeléctricos para señales analógicas de capacidad equivalente. El volumen físico ocupado también está determinado por el límite de consumo de potencia eléctrica, que en las instalaciones normales es de 400 watt/m² tanto para el consumo desde la red de distribución como para el cálculo de calorías del aire acondicionado.

Según [SIEG02] la señal de salida de estos amplificadores puede ser aproximada a un polinomio:

$$Z_L(n) = c_1 A + c_3 A^3 + c_5 A^5 \quad (3.4)$$

Donde los coeficientes c_i son coeficientes reales si el amplificador sólo agrega distorsión de amplitud, y coeficientes complejos si el amplificador agrega distorsión de amplitud y de fase.

El modelo planteado en [SIEG02] puede ser adaptado a cualquier amplificador SSPA según el estudio planteado por el autor, solo basta adaptar los coeficientes del polinomio por medio de los parámetros obtenidos con la ganancia, punto de intercepción de tercer orden, y salida de ganancia con 1 db de compresión.

3.3 Redes de Kohonen

Una red de Kohonen, o SOM (Self-Organizing Map, o Mapa Autoorganizado) es una red neuronal de aprendizaje no supervisado, competitiva, distribuida de forma regular en una rejilla de, normalmente, dos dimensiones, cuyo fin es descubrir la estructura subyacente de los datos introducidos en ella. A lo largo del entrenamiento de la red, los vectores de datos son

introducidos en cada neurona y se comparan con el vector de peso característico de cada neurona. La neurona que presenta menor diferencia entre su vector de peso y el vector de datos es la neurona ganadora (o BMU) y ella y sus vecinas verán modificados sus vectores de pesos [CAST01].

3.3.1 Redes de Aprendizaje no Supervisado

Este modelo neuronal utiliza una estrategia de aprendizaje que los humanos utilizamos frecuentemente, el llamado aprendizaje no supervisado. Si el aprendizaje supervisado se asemeja al profesor que enseña y corrige al alumno, el aprendizaje no supervisado o autoorganizado es semejante al alumno que aprende por sí mismo, sin la ayuda de un profesor, pero disponiendo de un material docente, libros, etc.

Este tipo de aprendizaje no necesita que le mostremos los patrones objetivo para salida, ya que el algoritmo y la regla de modificación de las conexiones producen patrones de salida consistentes. Cuando la red procesa patrones con bastante grado de similitud, da la misma salida para ambos patrones, esto es, clasifica los patrones en categorías de patrones parecidos.

En el proceso de aprendizaje, la red mide cuanto se parecen el patrón que le llega y los que tiene almacenados y en función de ello los agrupa en una categoría o otra, aunque en un principio no sepa que salida corresponderá a cada tipo o grupo de patrones de entrada, ni que atributos usará para clasificarlos.

En general, los métodos de aprendizaje no supervisado usan representaciones modélicas de los objetos a reconocer y a clasificar. Por ejemplo, una aplicación de reconocimiento de caras se puede pasar la fotografía como un mapa de bits, pero esto sería muy costoso computacionalmente, pero, sin embargo, si se pasa una serie de valores como anchura de ojos, anchura de boca, tamaño de frente, etc., la red neuronal podría clasificar la cara en función de sus parecidos.

3.3.2 Reseña Historia de Redes Kohonen

Los Mapas autoorganizativos de aprendizaje no supervisado (SOM) fue creado por Teuvo Kohonen, profesor de la academia de Finlandia, y es en honor a él el porque se le denominan Redes Kohonen.

En la década del 70, el profesor Kohonen tenía una teoría: “Existen evidencias que demuestran que en el cerebro existen neuronas que se organizan en muchas zonas, de forma que las informaciones captadas del entorno a través de los órganos sensoriales se representan internamente en forma de capas bidimensionales. Por ejemplo, en el sistema visual se han

detectado mapas del espacio visual en zonas de córtex (capa externa del cerebro). También en el sistema auditivo se detecta organización según la frecuencia a la que cada neurona alcanza la mayor respuesta (organización tonotópica). Aunque en gran medida esta organización neuronal está predeterminada genéticamente, es probable que de ella se origine mediante el aprendizaje. Esto sugiere, por tanto, que el cerebro podría poseer la capacidad inherente de formar mapas topológicos de las informaciones recibidas del exterior. Esta teoría podría explicar su poder de operar con elementos semánticos: algunas áreas del cerebro simplemente podrían crear y ordenar neuronas especializadas o grupos con características de alto nivel y sus combinaciones. Se trataría, en definitiva, de construir mapas espaciales para atributos y características”.

A partir de estas ideas, Teuvo Kohonen presentó en 1982 un sistema con un comportamiento semejante. Se trataba de un modelo de red neuronal con capacidad para formar mapas de características de manera similar a como ocurre en el cerebro. El objetivo de Kohonen era demostrar que en un estímulo externo (información de entrada) por si solo, suponiendo una estructura propia y una descripción funcional del comportamiento de la red, era suficiente para forzar la formación de mapas. En la actualidad, esta técnica es usada, generalmente, para clasificación automática de documentos, reconocimiento de patrones físicos, entre otros.

3.3.3 Funcionamiento de Redes de Kohonen

Un mapa de Kohonen es un conjunto de vectores de dimensión n distribuidos habitualmente en una retícula bidimensional (aunque nada impide que se puedan distribuir en 3 o más dimensiones). Para cada vector (nodo o neurona) se define un vecindario: cada vector puede tener 8 vecinos (es decir, una retícula rectangular) o 6 (retícula hexagonal). Hay razones teóricas para preferir una u otra; mientras que la más popular es la rectangular, la hexagonal tiene una base teórica más sólida [FYFE02].

La vecindad tiene otro parámetro: la forma de la función por la que se cambian los valores de los vectores que hay en ella: una forma tipo Gaussiana hará que el cambio de valores disminuya con la distancia, mientras que una forma tipo Burbuja cambiará de la misma forma todos los vectores que pertenezcan a la vecindad.

Para entrenar un mapa de Kohonen, primero hay que tener un conjunto de datos, que se dividirá en tres conjuntos: entrenamiento, prueba y validación. El de entrenamiento servirá precisamente para eso, el de prueba para seleccionar un mapa entre varios, y el de validación, posteriormente, para establecer el error final, que es el que vale. Esta es solo una de las formas posibles de entrenar un mapa. Para establecer el error total de un mapa con unos parámetros determinados, se puede usar otra metodología diferente llamada Leave-k-out: se divide el conjunto de datos en k partes, se usan $k-1$ para entrenamiento, y una para prueba; el procedimiento se repite para las k partes en las que se ha dividido el conjunto de entrenamiento [SERR01].

Normalmente, previamente al entrenamiento, hay que realizar algún tipo de reprocesamiento, para que todas las variables tengan aproximadamente el mismo rango y la misma desviación estándar. La forma más segura es hacer lo siguiente [SEERR01]:

- Convertir variables categóricas de n categorías en n variables (también se pueden convertir en $n-1$ variables). Los valores de estas variables dependerán de cómo hayamos preprocesado el resto del archivo: normalmente se pondrá al valor máximo en caso de que la variable corresponda a esa categoría y en el valor mínimo en el caso de que no. Por ejemplo, en una variable categórica con 3 categorías, la segunda categoría se codificaría como [min, max, min], que podría ser [-1, 1, -1].
- Normalizar el resto de las columnas, restando la media y dividiendo por la desviación estándar. Así todas las variables tendrán un rango entre -1 y 1.

Otros preprocesos posibles son aplicar logaritmos, en caso de que el rango de variación pase por varios órdenes de magnitud, o restar el mínimo y dividir por el rango de variación, para dar diferentes variables en el rango [0,1].

Aún así, cuando se trata de variables con distribución muy desigual, los modelos que se van a hallar no son excesivamente buenos, y habría que someterlos a algún preproceso adicional, que será totalmente heurístico (es decir, con un alto porcentaje de cocina), y que incluirá eliminación de algunas variables, proceso estadístico usando alguno de los componentes, anteriores y ejecuciones preliminares del algoritmo.

Lo importante del preproceso es no desvirtuar las relaciones métricas entre los diferentes valores de los componentes, que pueden dar lugar a artefactos en los resultados del algoritmo. Por ejemplo, en variables categóricas, si las categorías son totalmente diferentes el vector que represente cada categoría tendrá que hacerse de forma que la distancia a todas las demás sea la misma. Sin embargo, con variables que tengan una distancia "natural", por ejemplo, variables del tipo mediano, pequeño, grande, habrá que convertirlas a números enteros, o a puntuaciones que guarden esa misma relación.

Luego, dependiendo de la aplicación, se hacen más análisis sobre los resultados. Una de las herramientas de análisis que se puede aplicar se denomina mapa de Sammon, una herramienta de proyección bidimensional que permite proyectar un conjunto de muchas dimensiones en dos dimensiones [SEERR01].

3.3.4 Algoritmo de Kohonen

En una SOM la idea principal es encontrar regularidades o clases en los datos de entrada, y modificar sus pesos para ser capaz de reconocer estas regularidades o clases. La arquitectura típica de estos mapas autoorganizativo es:

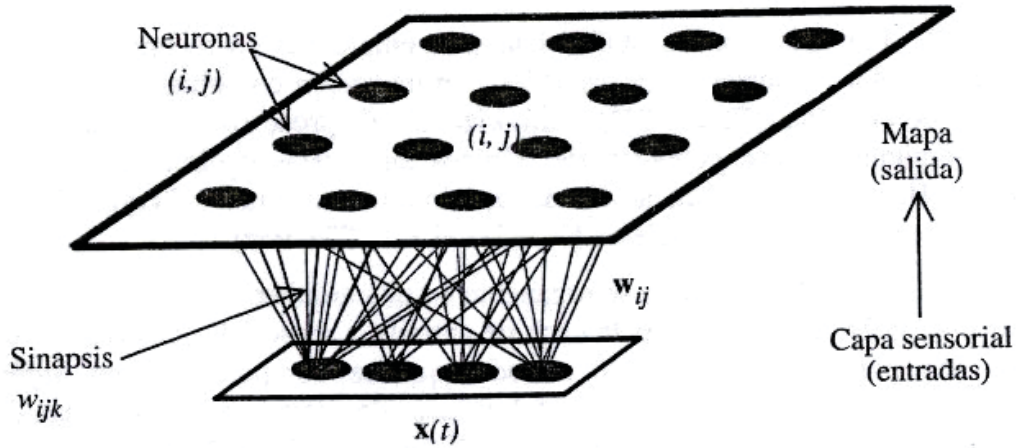


Figura 3.5 Arquitectura típica de un mapa autoorganizativo de Kohonen

Como se aprecia es una red de tipo unidireccional. La red se organiza en dos capas, siendo la primera capa la formada por las neuronas de entrada. La segunda capa consiste en un arreglo de neuronas de dos dimensiones. Como se necesitan dos índices para etiquetar cada neurona, los pesos sinápticos asociados a cada neurona tendrán tres índices (i,j,k) donde (i,j) indican la posición de la neurona en la capa y k, la componente o conexión con cierta neurona de entrada. Cada neurona utiliza como regla de propagación una distancia de su vector de pesos sinápticos al patrón de entrada.

Bajo estos conceptos, el algoritmo de Kohonen que se describe como sigue:

1. Inicialización de los pesos w_{ijk} .
2. Elección de un patrón de entre el conjunto de patrones de entrenamiento.
3. Para cada neurona del mapa, calcular la distancia Euclídea entre el patrón de entrada x y el vector de pesos sinápticos. Matemáticamente la distancia Euclídea entre el vector sináptico y la entrada, se representa como:

$$d^2(W_{ij}, X) = \sum_k (w_{ijk} - x_k)^2 \quad (3.5)$$

4. Evaluar la neurona ganadora, es decir aquella cuya distancia es la menor de todas
5. Actualizar los pesos sinápticos de la neurona ganadora y de sus vecinas según la siguiente regla:

$$\delta w_{ijk}(t) = \alpha(t) \cdot h(|i - g|, t) \cdot (x_k(t) - w_{ijk}(t)) \quad (3.6)$$

Donde $\alpha(t)$ es un factor llamado ritmo de aprendizaje que da cuenta de la importancia que

la diferencia entre el patrón y los pesos tiene en el ajuste de los mismos a lo largo del proceso de aprendizaje. Hay varias posibilidades para esta función, desde un constante hasta algún tipo de función monótona decreciente con el tiempo. h es una función de vecindad que indica en que medida se modifican los pesos de las neuronas vecinas. Con esto se quiere decir que cuando la neurona ganadora modifica sus pesos, la vecindad de esta neurona lo hace también, en mayor o menor medida según sea la forma funcional de h . En general, las funciones empleadas para h tienen un máximo en $|i - j| = 0$ y decrecen más o menos rápido a medida que esta distancia aumenta.

6. Finalmente, lo usual es fijar un número de iteraciones antes de comenzar el aprendizaje. Si no se llegó al número de iteraciones establecido previamente, se vuelve al paso 2.

3.3.5 Memoria vector-cuantificada temporal asociativa

Esta técnica de identificación y control de sistemas dinámicos se basa en usar mapas auto organizativos, tal como las redes de Kohonen, la técnica logra la minimización del error sobre una variedad de tareas complejas, entre las que se pueden mencionar [BARR01]:

- Predicción de Series de Tiempo
- Eficiencia de Modelos de Predicción Lineal a través de VQTAM

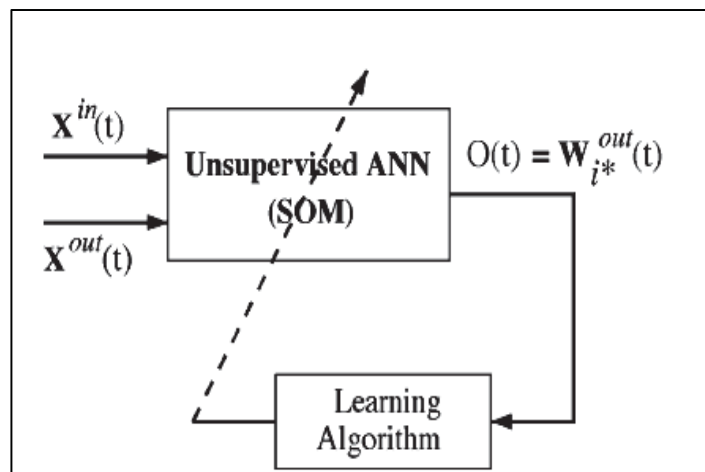


Figura 3.6 Teoría de VQTAM de forma grafica

VQTAM esta basada en las Redes de Kohonen, pero que la optimiza al permitir trabajar con entradas y salidas dinámicas solo con modificaciones menores del algoritmo real de SOM. Una de la forma de trabajar las Redes autoorganizativas, es utilizando algoritmos no supervisados para encontrar los vectores de pesos finales de cada neurona.

Durante la etapa de entrenamiento, ajeno a las entradas se tienen los siguientes parámetros:

- Numero de Neuronas de la red neuronal
- Cantidad de Iteraciones durante el entrenamiento
- Taza de aprendizaje de las neuronas
- Desviación Estándar de la función de vecindad entre neuronas

Luego de determinar el valor para cada parámetro, se inicializa la red neuronal, en la cual hay para cada entrada X , su correspondiente matriz de pesos, dada por un peso de entrada y un peso de salida, entonces, la red neuronal tiene como entrada:

$$x(t) = \begin{pmatrix} x^{in}(t) \\ x^{out}(t) \end{pmatrix} \quad (3.7)$$

Donde X_{in} es la entrada que se tiene naturalmente, y X_{out} es el valor que se debería obtener, la red neuronal

Cada neurona tendrá un vector de pesos de entrada y salida, dado por:

$$w_i(t) = \begin{pmatrix} w_i^{in}(t) \\ w_i^{out}(t) \end{pmatrix} \quad (3.8)$$

Con las variables, los vectores de pesos, y las entradas inicializadas, el proceso de entrenamiento de la red neuronal se da comienzo, el cual se da por un número de T de iteraciones donde primero se elige la neurona ganadora, y luego, se actualizan los pesos, y así, sucesivamente, hasta completar el número de T iteraciones establecidas.

De todas las neuronas, se elige una neurona ganadora, por medio de la siguiente regla:

$$i^*(t) = \arg \min_{i \in A} \{ \| x^{in}(t) - w_i^{in}(t) \| \} \quad (3.9)$$

Luego de elegir a la neurona ganadora, se actualizan los pesos de todas las neuronas de la red, donde cada peso estará determinado por la diferencia de la entrada X y el peso anterior de la neurona, la función de vecindad Gausianna, y la tasa de aprendizaje de la red neuronal.

$$\Delta w_i^{in}(t) = \alpha(t) h(i^*, i; t) [x^{in}(t) - w_i^{in}(t)] \quad (3.10)$$

$$\Delta w_i^{out}(t) = \alpha(t) h(i^*, i; t) [x^{out}(t) - w_i^{out}(t)] \quad (3.11)$$

Finalizado la etapa de entrenamiento, se obtiene los pesos de salida de cada neurona, que generan la siguiente formula:

$$Z(r) = \sum w_i^{out} r^{i+1} \quad (3.12)$$

Donde $i=1,2,\dots$ hasta N, donde N es el numero total de neuronas.

Sin embargo, se puede usar un modelo de autoregresión lineal para obtener los vectores de pesos deseados [BARR01]. Para entender bien esta técnica se debe comprender varias temáticas de algebra lineal, en este documento analizaremos la descomposición QR y la técnica de optimización matemática de los mínimos cuadrados.

El proceso para poder lograr los coeficientes del polinomio trabaja del mismo modo a como se obtiene en la técnica descrita en la Figura 4.2 y el valor de los pesos para este caso es el resultado obtenido en la ecuación (3.26).

3.3.6 Descomposición QR

En álgebra lineal, la Descomposición QR (también llamada factorización QR) de una matriz es una descomposición matricial como un producto de una matriz ortogonal y una triangular, habitualmente es usada esta factorizacion para resolver problemas de Mínimos Cuadrados [GARC01].

La descomposición QR de una matriz cuadrada real A es:

$$A = QR \quad (3.13)$$

Donde Q es una matriz ortogonal $Q^T Q = I$ y R es matriz triangular superior.

Existen diversas formas de obtener QR, esta el método de Ortogonalización de Gram-Schmidt, Reflexiones de Householder, y el uso de series de las Rotaciones de Givens. En este capitulo sólo se abarcará el método de Ortogonalización de Gram-Schmidt, por ser el más usado y el que mejores resultados ha generado.

Utilizando este método para obtener A tenemos:

$$A = (a_1 \mid \dots \mid a_n) \quad (3.14)$$

Entonces

$$u_k = a_k - \sum_{j=1}^{k-1} \text{proj}_{e_j} a_k \quad (3.15)$$

$$e_k = \frac{u_k}{\|u_k\|} \quad (3.16)$$

Utilizamos los a_i s de la formula ecuación (3.14) para obtener:

$$a_k = \sum_{j=1}^{k-1} \text{proj}_{e_j} a_k + e_k \|u_k\| \quad (3.17)$$

Ahora estas ecuaciones pueden ser escritas en forma matricial de esta manera:

$$(e_1 | \dots | e_n) \begin{pmatrix} \|u_1\| & \langle e_1, a_2 \rangle & \langle e_1, a_3 \rangle & \dots \\ 0 & \|u_2\| & \langle e_2, a_3 \rangle & \dots \\ 0 & 0 & \|u_3\| & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \quad (3.18)$$

De este modo se tiene que

$$Q = (e_1 | \dots | e_n) \quad (3.19)$$

Finalmente, R se obtiene como sigue:

$$R = Q^T A = \begin{pmatrix} \langle e_1, a_1 \rangle & \langle e_1, a_2 \rangle & \langle e_1, a_3 \rangle & \dots \\ 0 & \langle e_2, a_2 \rangle & \langle e_2, a_3 \rangle & \dots \\ 0 & 0 & \langle e_3, a_2 \rangle & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix} \quad (3.20)$$

3.3.7 Utilizando Descomposición QR en Mínimos Cuadrados

La técnica de Eficiencia de Modelos de Predicción Lineal a través de VQTAM está basada íntegramente en el método de los Mínimos Cuadrados, el cual necesita de la Descomposición QR para poder realizar la tarea.

Los Mínimos cuadrados se puede definir como una técnica de optimización matemática que, dada una serie de mediciones, intenta encontrar una función que se aproxime a los datos (un "mejor ajuste"). Intenta minimizar la suma de cuadrados de las diferencias ordenadas (llamadas residuos) entre los puntos generados por la función y los correspondientes en los datos. La técnica de mínimos cuadrados se usa comúnmente en el ajuste de curvas. Muchos otros problemas de optimización pueden expresarse también en forma de mínimos cuadrados, minimizando la energía o maximizando la entropía [IBOR01].

Entonces suponiendo que dada una matriz A de columnas independientes, si tenemos [GARC01]:

$$Ax = b \quad (3.21)$$

Utilizando la Descomposición QR para los mínimos cuadrados . Donde b es la matriz con los resultados que se gustaría obtener, por la formula (3.14) se puede escribir (3.21) de la forma:

$$Rx = Q^T b \quad (3.22)$$

Esto porqué se puede reemplazar (3.21) utilizando las matrices Q y R de la descomposición QR se tiene:

$$A^T Ax = A^T b \quad (3.23)$$

Luego, tomando en cuenta a las ecuaciones en (3.14), (3.23) se puede escribir de la forma:

$$(QR)^T QRx = (QR)^T b \quad (3.24)$$

Del mismo modo se puede reescribir (4.26) y obtener:

$$R^T Q^T QRx = R^T Q^T b \quad (3.25)$$

Finalmente, se obtiene:

$$Rx = Q^T b \quad (3.26)$$

Donde la matriz x equivale a los coeficientes para formular el monomio.

Capítulo 4

Sistema de Transmisión OFDM

En este capítulo se presenta el sistema de transmisión OFDM el cual opera en todas sus partes, el cual posee una red neuronal implementada que pretende reducir la distorsión no lineal que genera el amplificador a la señal.

Generado el sistema OFDM, este se convertirá en dos sistemas, en los cuales ambos varía por la forma de encontrar el vector de pesos de salida de las neuronas, por un lado se ocupa un algoritmo no supervisado, explicado en el capítulo 4.5, el otro ocupa los mínimos cuadrados para obtener ese mismo vector dentro de la SOM.

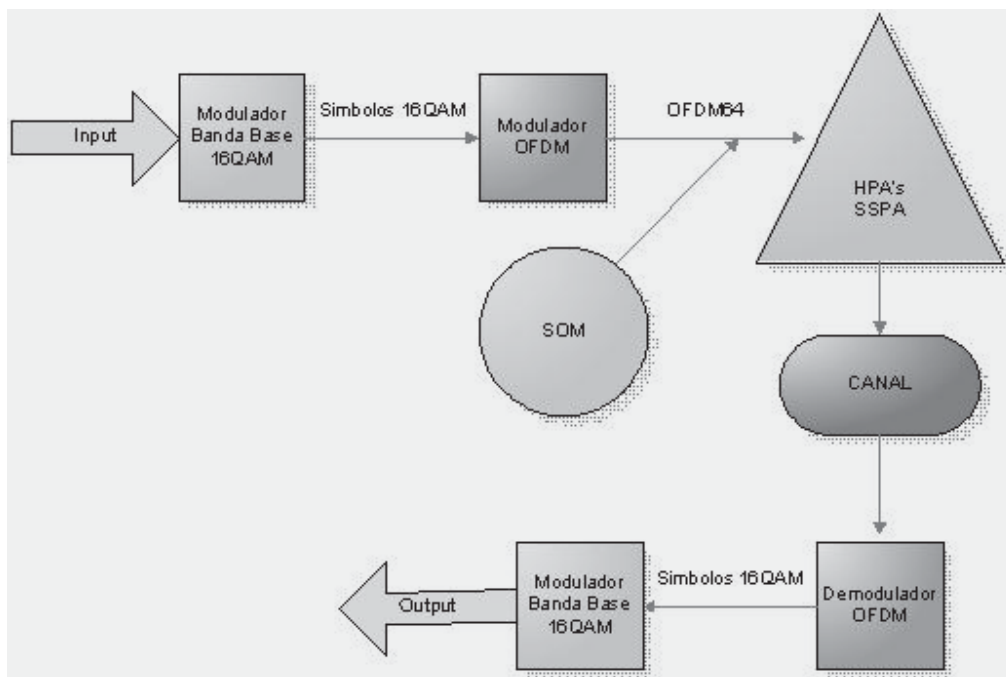


Figura 4.1 Diagrama del Sistema de Transmisión OFDM

En la figura 5.1 se puede apreciar las estructuras más importantes dentro de este sistema, los cuales serán explicados en los puntos siguientes. Para así, poder llegar a lo que será la implementación de un polinomio que entrega resultados lineales producto de la aplicación de coeficientes de la SOM.

En este Capítulo se mostrará la implementación de la entrada, modulador 16QAM, modulador OFDM y Amplificador de Potencia, Red Neuronal, y el demodulador OFDM.

Sin embargo, cabe destacar dos puntos:

- Se implementó una versión del sistema con 64QAM, pero en este capítulo sólo se presentarán los resultados, y no la forma en que se realizó la implementación.
- Se aplicaron dos tipos de sistemas de transmisión basado en SOM, las cuales corresponden a las dos técnicas explicadas en el capítulo 4.5

4.1 Entrada (INPUT)

La entrada al sistema es un grupo de bits, en nuestro caso, se trabajará con 100.000 bits aleatorios, generados con anterioridad, los cuales han sido registrados en un archivo. Por lo mismo, sólo basta leer el archivo desde el programa para poder visualizar los bits de entrada de bits aleatorios.

También es factible hacer un módulo que se encargue automáticamente de generar los bits necesarios para realizar el experimento, sin embargo, no es tan importante, lo realmente importante es tener la entrada de 100.000 bits aleatorios. Este tipo de entrada de archivos estáticos, es elaborado con la intención de que al momento de evaluar el comportamiento de la red neuronal, la entrada se mantenga constante, y no sea una variable, por esto, los parámetros que solamente iremos ajustando serán solo los valores propios de la red neuronal.

4.2 Modulador de amplitud en cuadratura de 16 estados (16QAM)

Aquí el flujo de datos de entrada (100.000 bits) es dividido en grupos de 4 bits para lograr una cuadratura de 16 valores (Figura 5.2).

Por ejemplo, si los valores de entrada fueran 101001010101001110101010, nos quedarían 6 grupos tal como siguen 1010, 0101, 0101, 0011, 1010, 1010.

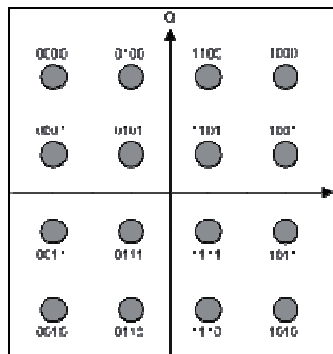


Figura 4.2 Ejemplo de 16QAM

El algoritmo en pseudo código utilizado para formar cada cuadratura puede ser analizado en la tabla 4.1:

```

Arreglo [4] [2500];
i++;
j++;
Mientras (Entrada sea valida)
{
    Desde (i=0; i++; i<4)
    Arreglo [i] [j] = entrada
    j++;
}
    
```

Tabla 4.1 Algoritmo utilizado para implementar 16QAM

Entonces, se presenta la formulación matemática del modulador QAM:

$$q_k = a + bj \tag{4.1}$$

Donde se tiene que:

- $a, b = 1, 3, -1, -3$
- k representa el cuadrante de los 16 puntos originales, es decir, $k = 1, \dots, 16$.

Considerando los valores de a , b y k , se tienen los siguientes pares ordenados originales:

$$\begin{aligned}
 q_1 &= 1 + 1j \\
 q_2 &= 1 + 3j \\
 q_3 &= 3 + 1j \\
 q_4 &= 3 + 3j \\
 q_5 &= 1 - 1j \\
 q_6 &= 1 - 3j \\
 q_7 &= 3 - 1j \\
 q_8 &= 3 - 3j \\
 q_9 &= -1 + 1j \\
 q_{10} &= -1 + 3j \\
 q_{11} &= -3 + 1j \\
 q_{12} &= -3 + 3j \\
 q_{13} &= -1 - 1j \\
 q_{14} &= -1 - 3j \\
 q_{15} &= -3 - 1j \\
 q_{16} &= -3 - 3j
 \end{aligned}$$

Por lo tanto, quedaría una constelación ideal como sigue en la Figura 5.3:

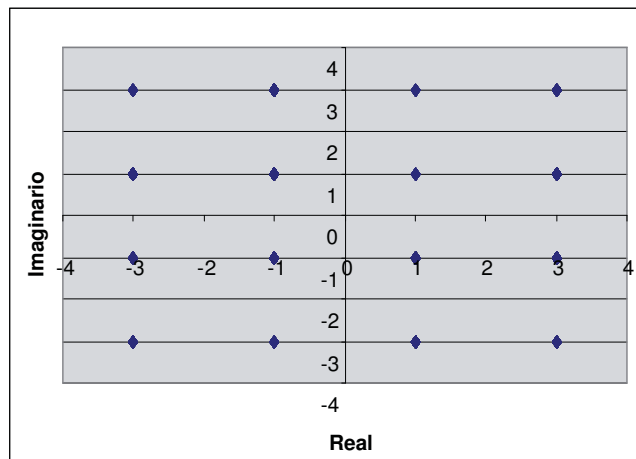


Figura 4.3 Constelación Ideal de 16QAM según Formula matemática obtenida del sistema OFDM con SOM implementada.

En el caso de una constelación de 64QAM, dado por $2^6 = 64$, entonces en vez de ser grupos de 4 bits, aquí sería grupos de 6 bits. De este modo, se obtiene una grafica como la que muestra el Figura 5.4.

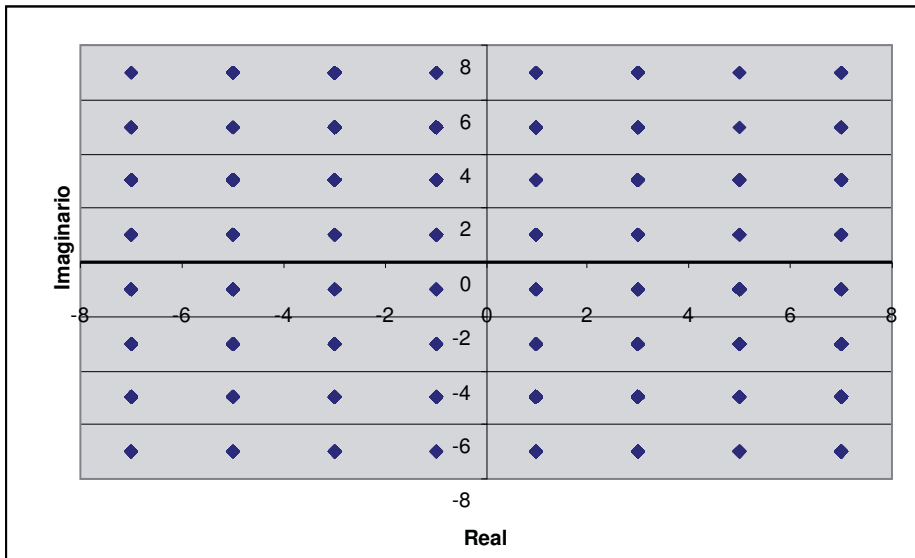


Figura 4.4 Constelación Ideal de 64QAM obtenida de la Formula matemática obtenida del sistema OFDM con SOM implementada.

4.3 Modulador Banda Base

Para poder enviar los bits de la señal bajo OFDM se debe tener en cuenta la Inversa de la Transformada Rápida de Fourier (IFFT), y que es lo que se quiere lograr (ver Figura 5.4).

La salida del modulador 16QAM debe ser aplicada a la Inversa de la Transformada de Fourier para poder obtener la ortogonalidad propia de las características del modulador.

El modulador OFDM que se ocupará en esta implementación es un modulador con 64 subcarriers, es decir, utilizaremos 64 simbolos 16QAM para obtener un solo símbolo OFDM

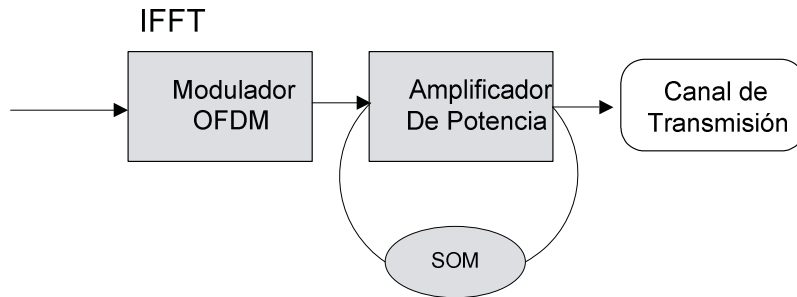


Figura 4.5 Esquema de módulos del modulador OFDM

Esta función se encarga de aplicar la Transformada de Fourier Rápida Inversa a los símbolos obtenidos en el modulador 16QAM, los cuales se les aplican a cada símbolo del Modulador OFDM.

Para obtener la ecuación matemática que describe a IFFT, primero se debe conocer la ecuación que describe a FFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-jk\Omega n} \quad (4.2)$$

por lo tanto, el cálculo de la IFFT se reduce a resolver:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{jk\Omega n} \quad (4.3)$$

Donde, $\Omega = 2\pi / N$.

En la fórmula, k representa el instante de tiempo, entonces, la IFFT de X está determinada por el símbolo de X , N número de subcarriers, k instante de tiempo dentro de X , y n canal.

Luego, se toma la entrada enviada por el modulador 16QAM, y se divide en grupos de 64, es decir, se trabajará con un modulador OFDM de 64 canales, o sea, 64 símbolos al mismo tiempo, los cuales irán ortogonalmente ubicados. Lo cual se puede reflejar en el algoritmo descrito en la tabla XX:

<p>Desde $i=0$ hasta $i<64$</p> <p>{</p> $x_{real}[n] = [real \cdot \cos(\frac{2\pi}{N} i \cdot k) - imag \cdot \sin(\frac{2\pi}{N} i \cdot k)] / N$ $x_{imag}[n] = [imag \cdot \cos(\frac{2\pi}{N} i \cdot k) + real \cdot \sin(\frac{2\pi}{N} i \cdot k)] / N$ <p>} Fin Desde</p>
--

Tabla 4.2 Algoritmo utilizado para implementar la IFFT

Luego, al símbolo obtenido por medio del procedimiento de la tabla 4.2, se le calcula la amplitud, y este valor es enviado al amplificador.

$$A[n] = \sqrt{x_{real}[n]^2 + x_{imag}[n]^2} \tag{4.8}$$

4.4 SOM, Red Neuronal

La Red Neuronal esta dividida en 2 etapas claramente visibles: la etapa de entrenamiento, y la etapa de implementación. Durante la etapa de entrenamiento, la red neuronal obtendrá los valores necesarios para poder llevar a cabo la tarea asignada, y durante la etapa de implementación, la red neuronal aplicará sus resultados en nuestro sistema OFDM.

La tarea fundamental es encontrar la curva inversa de la curva obtenida por el amplificador de potencia con los valores de la señal OFDM, así, se corregirá los errores que por su naturaleza el amplificador genera. Es decir, si r es la amplitud, el amplificador esta dado por la siguiente fórmula:

$$y = e^{Fase(r)_j} \cdot P(r) \tag{4.9}$$

Donde $P(r)$ es el polinomio que genera distorsión por amplitud, el cual es [SIEG02]:

$$P(r) = c_1 r^1 + c_3 r^3 + c_5 r^5 \tag{4.10}$$

Teniendo a $Z(r)$ como el resultado que arroja la red neuronal que se esta implementando, se puede conseguir:

$$y = e^{Fase(r)_j} \cdot Z(r) \tag{4.11}$$

La técnica buscan obtener la curva inversa a la obtenida por el amplificador de potencia tal como se muestra en al figura 4.5.

Donde:

- a): representa a una curva de un polinomio con distorsión no lineal, dada por

$$y = P(r) \quad (4.12)$$

- b) : representa a la curva Compuesta entre la curva del polinomio y su inversa, o sea, la amplitud linealizada, dada por:

$$w = P(z) \quad (4.13)$$

- c): representa la curva de la Inversa del Polinomio, dada por

$$z = P'(P) \quad (4.14)$$

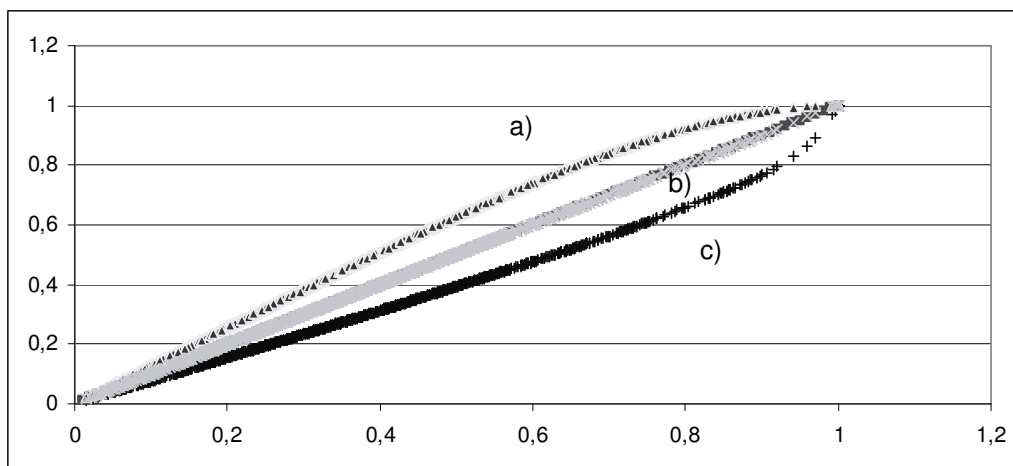


Figura 4.6 Ejemplo de Curva del Polinomio, Inversa y compuesta, la curva es una simulación realizada con el sistema implementado bajo 10000 datos de entrada.

En esta implementación, se trabajara específicamente con la técnica de identificación de patrones VQTAM (Memoria Vector-Cuantificada temporal asociativa) que esta basada en las Redes de Kohonen.

En el Capítulo 3 se menciona que se puede aplicar de diversas formas la técnica de VQTAM:

- Predicción de Series de Tiempo
- Eficiencia de Modelos de Predicción Lineal a través de VQTAM

Mientras la primera trata sobre encontrar una neurona ganadora y en base a esto ir generando el mapa de neuronas que darán forma a los coeficientes. La técnica basada en mínimos cuadrados trabaja la solución como si se tratará de un modelo lineal.

Teniendo en cuenta la teoría explicada en el punto 3.3.5 de este documento, entonces si representamos un modelo lineal como:

$$\hat{y}(t+1) = a_0 + \sum_{j=1}^{n_y} a_j y(t-j+1) \quad (4.15)$$

Donde a_j son los coeficientes del modelo, la técnica básica para computar los coeficientes $a = [a_0 a_1 \dots a_{n_y}]^T$ sería el utilizado por el método de los mínimos cuadrados (Ver capitulo 3.3.7).

$$a = (R^T R)^{-1} R^T p \quad (4.16)$$

Donde p es el vector de predicción y R es la matriz de regresión. Usando VQTAM estas entidades son construidas a través de vectores de pesos:

$$w_i^{in}(t) = [w_{i,1}^{in}(t), \dots, w_{i,n_y}^{in}(t)]^T \quad (4.17)$$

$$w_i^{out}(t) = w_i^{out}(t) \quad (4.18)$$

Con $i = 1, \dots, N$ como sigue:

$$p = [w_1^{out}(t) w_2^{out}(t) \dots w_N^{out}(t)]^T \quad (4.20)$$

$$R = \begin{pmatrix} 1 & w_{1,1}^{in}(t) & w_{1,2}^{in}(t) & \dots & w_{1,n_y}^{in}(t) \\ 1 & w_{2,1}^{in}(t) & w_{2,2}^{in}(t) & \dots & w_{2,n_y}^{in}(t) \\ \dots & \dots & \dots & \dots & \dots \\ 1 & w_{N,1}^{in}(t) & w_{N,2}^{in}(t) & \dots & w_{N,n_y}^{in}(t) \end{pmatrix} \quad (4.21)$$

Donde N es el número de neuronas de la SOM.

Obtenido la inversa del polinomio $Z(r)$, se envían los datos al amplificador, los cuales normalizados, arrojan datos linealizados.

4.5 Amplificador de Potencia

El amplificador de alta potencia permite la recuperación de potencia, sin embargo introduce intermodulación a los símbolos OFDM. La descripción matemática que simula el comportamiento del amplificador se describe mediante el siguiente polinomio:

$$P(r) = c_{10}r^1 + c_{30}r^3 + c_{50}r^5 \quad (4.22)$$

Donde r es la amplitud normalizada de cada símbolo OFDM, el cual está normalizado, para así mantener valores entre 0 y 1.

Los coeficientes de 5º orden c_{10}, c_{30}, c_{50} del modelado de amplificadores son obtenidos de una tabla 5.1 ya publicada [SIEG02] y sus valores se aprecian en la siguiente tabla:

	MGA-86563	MGA-52543
c_1	10	5.6234
c_3	-316.23	-0.08913
c_5	-21097.31	-6.1441
c_{10}	1.3865	1.2895
c_{30}	-0.2731	-0.790
c_{50}	-0.1134	-0.2105

Tabla 4.3 Tabla de coeficientes de 5º orden

Este modelo de amplificador es un amplificador GaAs MMIC de Agilent Technologies, y está ahora ya disponibles en el mercado, y ofrecen un bajo ruido y una excelente ganancia. El polinomio es trabajado con los coeficientes $c_{10}, c_{30},$ y $c_{50},$ porque estos coeficientes trabajan con valores normalizados, y para toda la implementación de esta red neuronal, se utilizarán solo valores normalizados, es decir, valores, que fluctúen entre 0 y 1 (Ver Figura 4.6).

Para lograr esto la amplitud también deberá estar normalizada, esto se logra dividiendo la amplitud de entrada por la amplitud máxima obtenida de las siguientes y anteriores amplitudes obtenidas durante el proceso de modulación OFDM.

Al canal es enviado un valor que también está determinado por la fase del símbolo OFDM, dado por:

$$y = e^{Fase(r)j} \cdot P(r) \quad (4.23)$$

Donde y representa los números complejos que representan al símbolo OFDM luego de pasar por la amplificación.

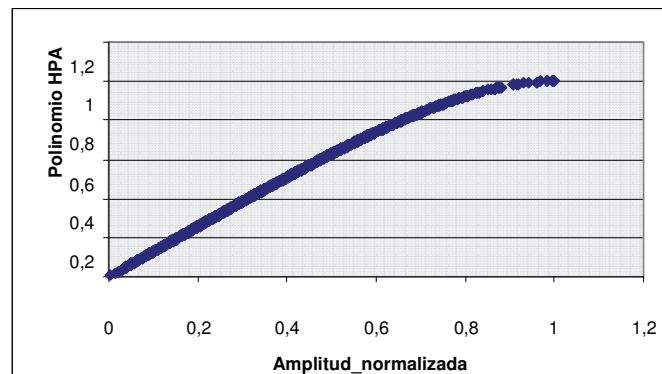


Figura 4.7 Amplitud versus Valores del Polinomio

4.6 Canal Gaussian

Los símbolos obtenidos en el módulo anterior son transmitidos a través de un canal Gaussiano en el cual se añade ruido térmico. Las ecuaciones matemáticas que describen este comportamiento son:

$$n_k = n_q + jn_l \quad (4.24)$$

Donde:

- $n_q = \mu + R \cdot \cos(2\pi \cdot u)$
- $n_l = \mu + R \cdot \text{sen}(2\pi \cdot u)$

- $$R = \sigma \cdot \sqrt{2Ln\left(\frac{1}{1-u}\right)}$$
- $$\sigma^2 = \frac{1}{2 \cdot 10^{10} \frac{snr}{}}$$

Además se define a, b son números aleatorios distribuidos uniformemente, es decir, con distribución Normal con media $\mu = 0$ y σ^2 conocida.

4.7 Demodulador Banda Base

El demodulador OFDM vuelve a 64 símbolos 16QAM el símbolo OFDM recibido del amplificador, o sea, desmaraña todo, y obtiene los 64 símbolos ubicados ortogonalmente dentro del símbolo OFDM.

En la figura 5.8 se aprecia la salida de símbolos 16QAM obtenidos del modulador OFDM, donde el símbolo Z de OFDM no fue pasado por un amplificador.

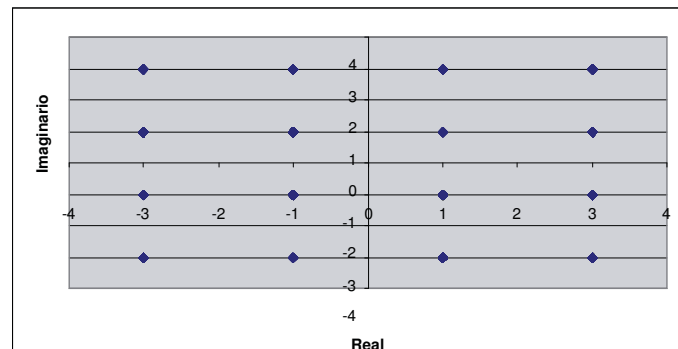


Figura 4.8 Salida del demodulador sin amplificador

Sin embargo, en la figura 5.8 el grafico representado muestra los datos enviados por el modulador OFDM obtenidos del amplificador, o sea, procesados por este, y se nota claramente,

la distorsión y el error que este genera sobre los símbolos.

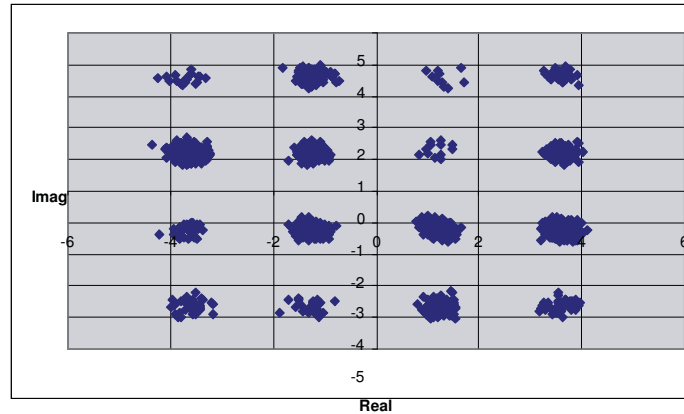


Figura 4.9 Salida del demodulador con amplificador

La idea principal, es hacer que los datos de salida del demodulador sean idénticos a los datos de entrada del modulador OFDM, y que no se genere esta distorsión, es decir, el proceso de la red neuronal tiene como fin hacer que la figura 5.9 tienda a comportarse como la figura 5.8.

Este módulo recibe los símbolos y calcula la distancia existente entre el punto recibido y los cuatro puntos originales (Figura 5.7), tomando como punto recepcionado aquel punto original al cual se encuentra más próximo, para ello se utiliza la siguiente expresión matemática:

$$v_k = \min\{abs(q, r_k)\} \quad \forall k = 1, 2, \dots, R_b / 2 \quad (4.24)$$

Donde,

- R_b es la razón de bits de transmisión.

Además es en esta etapa donde se va acumulando el error producido, para ello se tiene que:

$$e_k = \begin{cases} 1 & \text{si } z_k \neq v_k \\ 0 & \text{si } z_k = v_k \end{cases} \quad (4.25)$$

Con lo cual se obtiene la Razón de Error por Símbolo,

$$SER = \frac{\sum_{k=1}^{R_b/2} e_k}{R_b/2} \quad (4.26)$$

y la Razón de Error de Bits,

$$BER = \frac{SER}{2} \quad (4.27)$$

Con estos valores se genera el gráfico BER v/s SNR.

4.8 Salida (OUTPUT)

Este módulo es el encargado de recibir la cadena de símbolos y obtener los bits que fueron transmitidos.

Este último paso da así por concluida la etapa de recepción y también el proceso completo del Sistema de Transmisión OFDM.

Capítulo 5

Evaluación de Resultados

Para evaluar los resultados, primero se evaluará la forma más óptima para tomar los vectores de pesos para cada neurona, tal como lo señala [BARR01].

Para esto, se utilizará Magnitud del Vector de Error (EVM), y que se define como:

$$EVM(\%) = \sqrt{\frac{\sum_{k=1}^N (I_r - I_t)^2 + (Q_r - Q_t)^2}{\sum_{k=1}^N (I_t^2 + Q_t^2)}} \cdot 100 \quad (5.1)$$

Donde I_r y Q_r son los símbolos recibidos por el demodulador M-QAM (Sea de 64QAM y 16 QAM), y los símbolos I_t y Q_t son los símbolos transmitidos. N es el total de símbolos transmitidos. La EVM se define como la relación, en porcentaje, entre la magnitud de error promedio de los símbolos recibidos y la magnitud del símbolo original. EVM permite conocer la calidad de la modulación, y cuantifica la desviación de los símbolos recibidos a partir de su posición ideal, sin tomar en cuenta la naturaleza de las afecciones. (Figura 5.1).

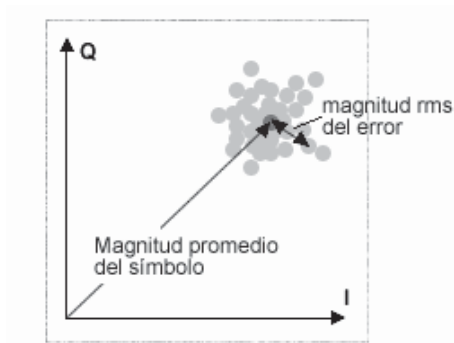


Figura 5.1 Magnitud del Vector de Error (EVM)

Porque en [BARR01] señala, en la página 1247, que:

$$w_i^{out}(t+1) = w_i^{out}(t) + a(t)h(i^*, i; t)[x^{out}(t) - w_i^{out}(t)] \quad (5.2)$$

La ecuación de la actualización del vector de pesos de salida (5.2) que representa a las redes de Kohonen más un algoritmo no supervisado para encontrar los vectores de peso, se puede poner como (3.26), que esta basada en Mínimos cuadrados utilizando la Descomposición QR. Por lo tanto, se comparará el EVM de dos programas iguales, pero que se diferencian en la forma de obtener el vector de peso, ya sea, por (5.2) o por (4.26).

Primero se comparan las dos constelaciones de 16QAM obtenidos por cada sistema. En la Figura 5.2 se puede visualizar la constelación 16QAM obtenida por sistema que basa la obtención de pesos de salida según el uso de mínimos cuadrados. La Figura 5.3 muestra la constelación 16QAM donde los pesos de salida han sido obtenidos por un algoritmo supervisado, como es el modelo Forward de VQTAM.

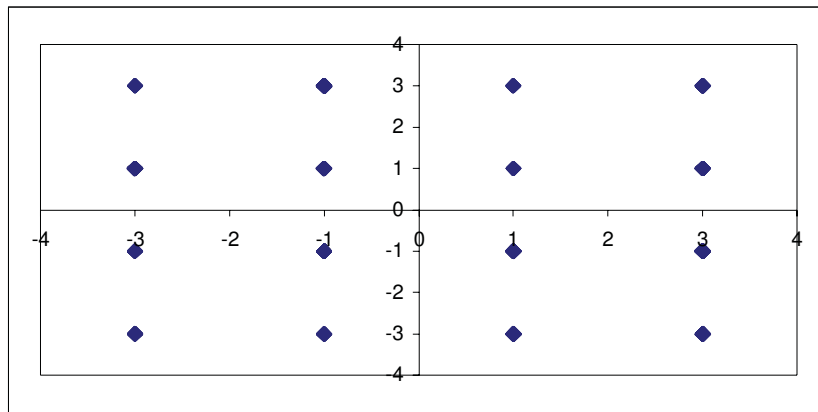


Figura 5.2 Constelación 16QAM de la salida del demodulador con un sistema que obtiene sus pesos de salida por medio de Mínimos Cuadrados.

Como se puede apreciar en la Figura 5.2 la salida es casi idéntica a la constelación teórica 16QAM dibujada en la Figura 5.7, en efecto, el EVM es de 0.6909489938925698% para 10 neuronas. Por otra parte, la constelación obtenida en la Figura 5.3, correspondiente al sistema que obtiene los pesos con el algoritmo Forward de VQTAM, obtiene un EVM igual a 17.259877992629225% con 10 neuronas.

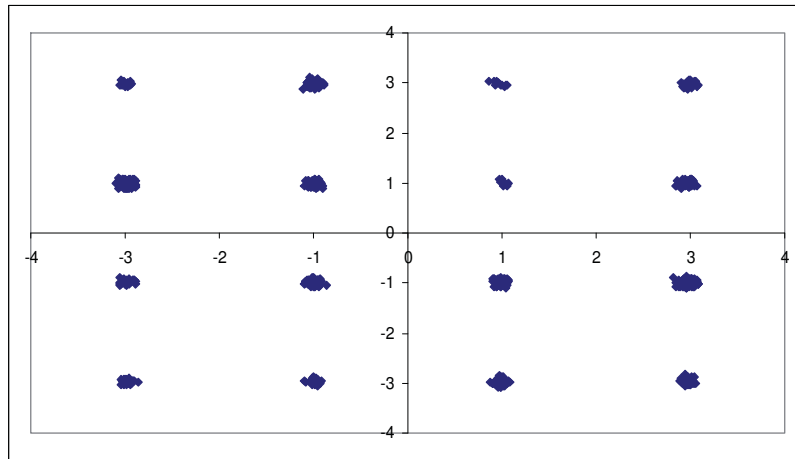


Figura 5.3 Constelación 16QAM de la salida del demodulador con un sistema que obtiene sus pesos de salida por medio del algoritmo no supervisado VQTAM.

Por todo lo anterior se seguirá trabajando con la opción que nos genera la constelación de la Figura 5.2, debido a que las diferencias son significativas, sobre todo, suponiendo que ninguno de los gráficos obtenidos en las Figuras 5.2 y 5.3 contiene ruido externo del canal (canal Gaussiano).

Como se aprecia en la Figura 5.4 se puede ver la constelación para 16QAM de la salida del demodulador OFDM dada una ráfaga de 10.000 aproximadamente para un ruido gaussiano con SNR=8 db, además, de la distorsión propia que genera el amplificador de potencia.

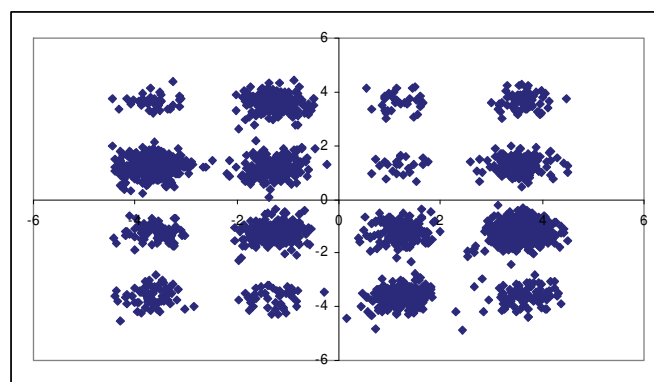


Figura 5.4 Constelación 16QAM para la salida del demodulador dada por OFDM + HPA + W.

El ruido gaussiano en esta figura, agrega un máximo de error del orden de 28% a la constelación, entonces es deducible, que el resto de error es producto del amplificador de

potencia. Como se visualiza en la Figura 5.5, el ruido que genera el amplificador de potencia para esta modulación QAM, es menor, y toda la distorsión que el sistema de 16QAM obtiene, en gran parte es producto del ruido del Canal Gaussiano.

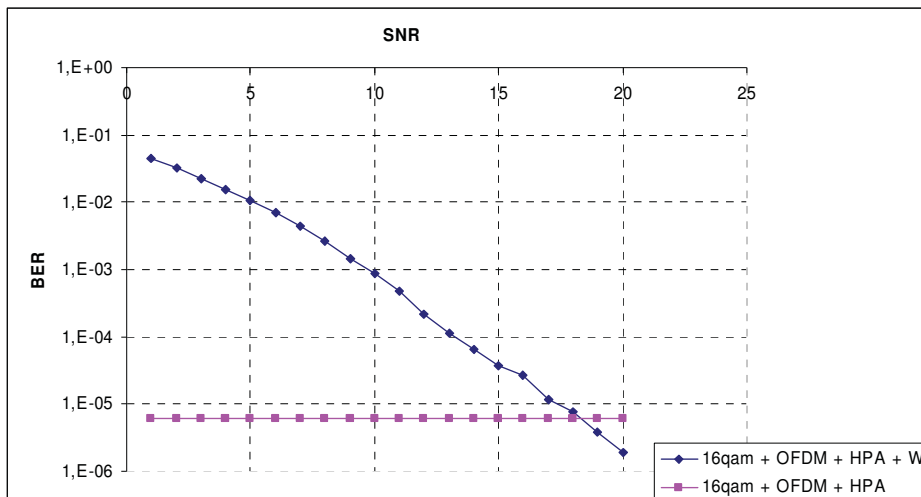


Figura 5.5 Curva BER vs SNR. Sistema sin Corrector Neuronal.

Para el mismo caso de la Figura 5.4, pero con 64QAM, los resultados son mayormente nefastos, esto debido a que son más canales compartiendo el ancho de banda, con lo cual se genera una mayor distorsión por parte del amplificador, y el ruido del canal, que puede ser similar al de 16QAM, se hace más notorio, y afecta más fuerte a los símbolos que recibe el demodulador.

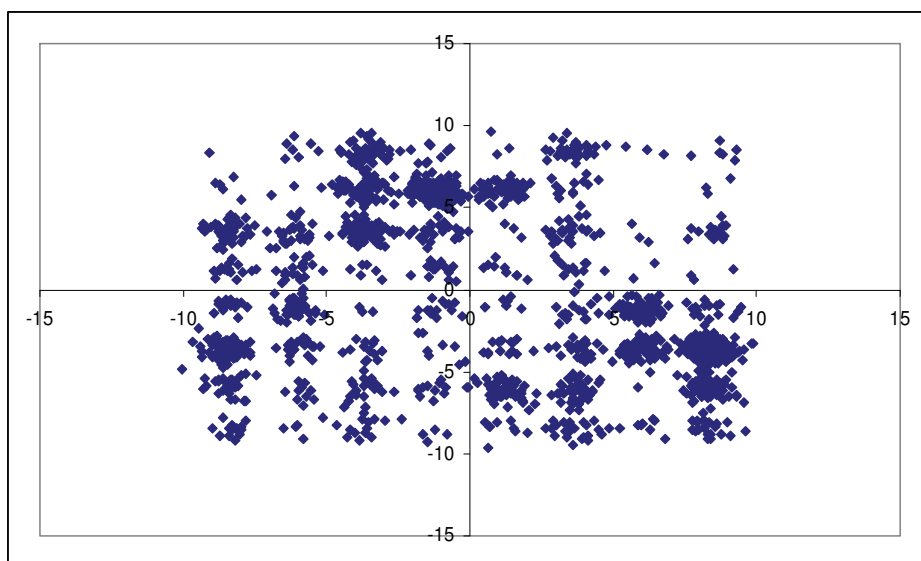


Figura 5.6 Constelación 64QAM para la salida del demodulador dada por OFDM + HPA + W.

El ruido que se logra apreciar en la constelación, está totalmente, vinculado al ruido que genera el amplificador esta vez, como se muestra en la Figura 5.7, la distorsión propia que genera el amplificador de potencia es muy fuerte, y es mas notorio que en el caso de 16QAM.

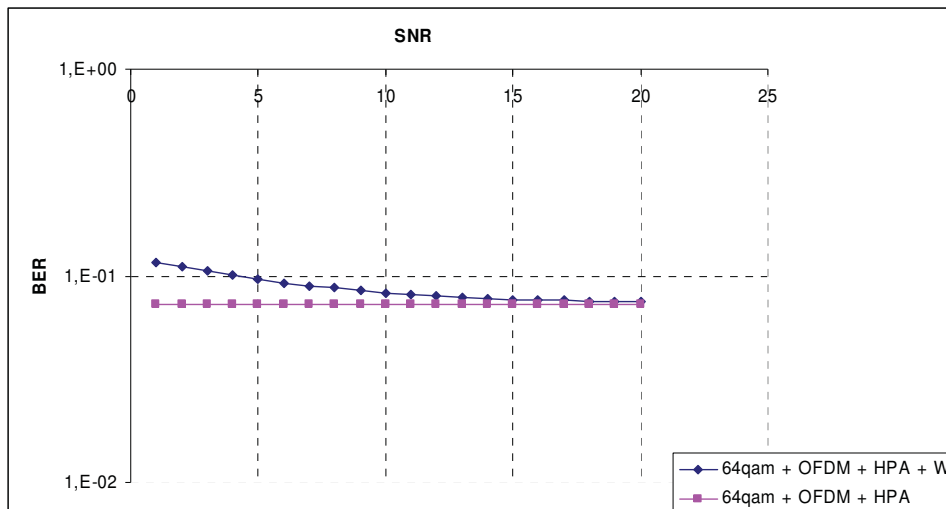


Figura 5.7 Curva BER vs SNR. Sistema sin Corrector Neuronal.

El sistema que se implementó, trata de corregir los símbolos OFDM que salen del amplificador, generando coeficientes inversos para que actúen sobre ellos, y minimicen el error del amplificador de potencia, tal como se mencionó, se probó dos formas de encontrar los pesos de salida, que corresponderían a los coeficientes que generaría la curva inversa.

En este caso, el valor constante de BER que produce sólo el amplificador de potencia es de 0,072916667, por lo tanto, la curva de BER vs SNR del sistema con ruido de canal gaussian, nunca podrá ser menor a este valor.

Dado el análisis anterior, según EVM, quedo demostrado que la forma de encontrar los pesos de salida vía mínimos cuadrados era más óptimo que vía el método Forward de VQTAM.

En el Figura 5.8, se presenta la curva obtenida para 16QAM del BER obtenido con cada incremento del SNR del canal gaussian, en un sistema con un corrector, basado en la obtención de los pesos de salida de cada neurona dado por Mínimos Cuadrados.

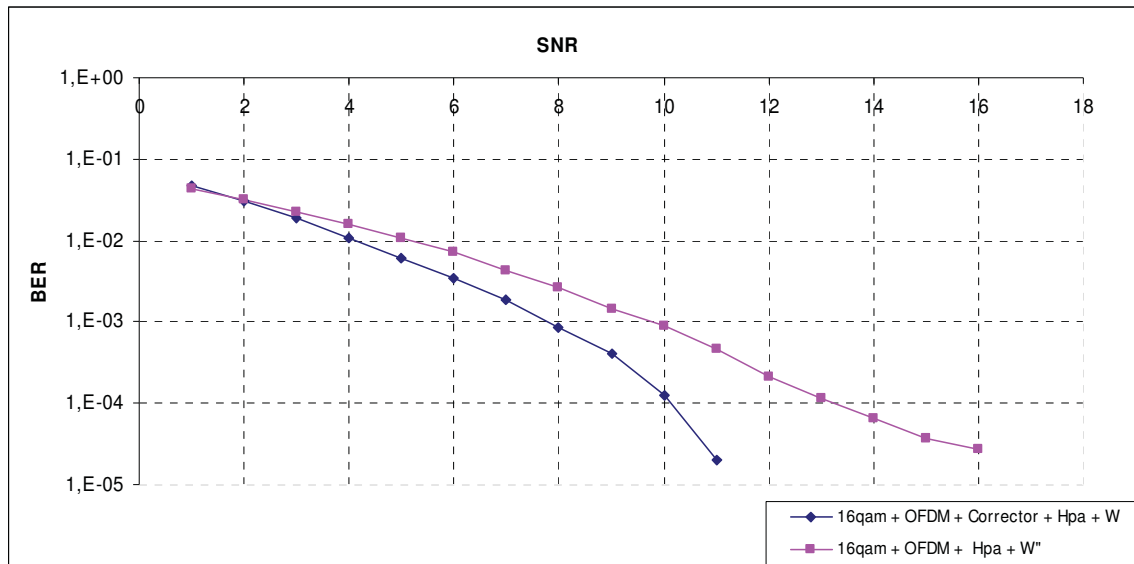


Figura 5.8 Curva BER vs SNR. Sistema con Corrector Neuronal

Es fácil advertir la eficiencia del sistema implementado, y lo que bien corrige la curva. Los valores de BER pueden ser revisados en la tabla 5.1.

SNR	BER
1	4,69E-02
2	3,02E-02
3	1,86E-02
4	1,08E-02
5	6,11E-03
6	3,38E-03
7	1,88E-03
8	8,55E-04
9	4,00E-04
10	1,27E-04
11	1,98E-05
12	0,00E+00

Tabla 5.1 Datos para obtener la Curva BER vs SNR de la Figura 5.8

Con esto, se obtiene una ganancia sobre el sistema sin corrector de 2 db a un $BER = 1.27 \cdot 10^{-3}$, tal como se puede apreciar en la Figura 5.9.

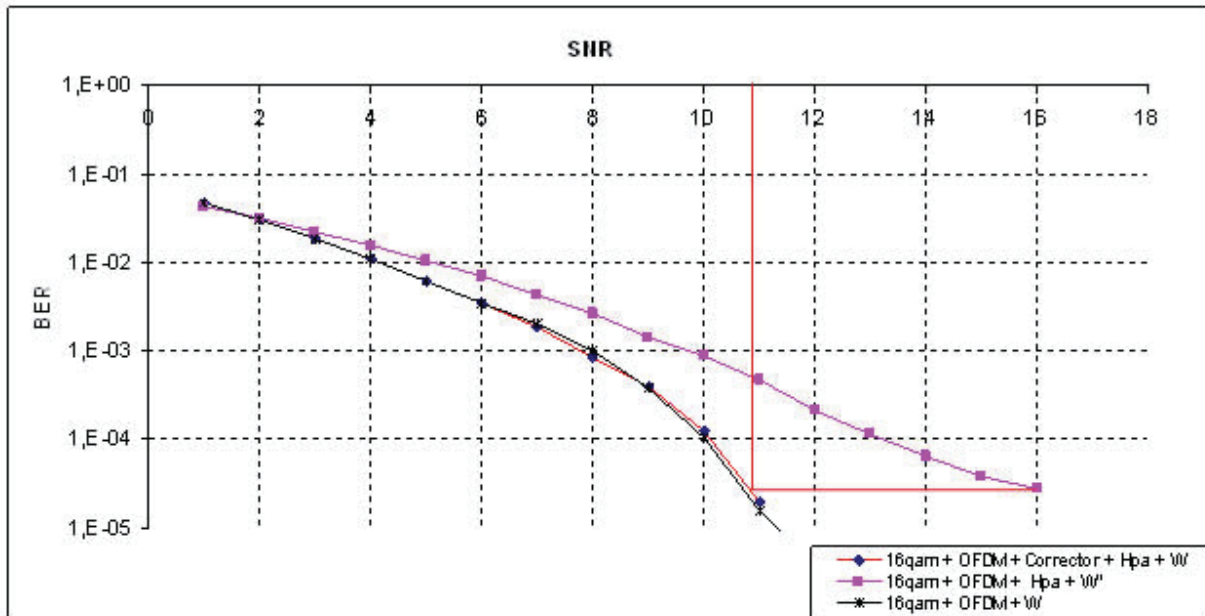


Figura 5.9 Curva BER vs SNR, ganancia obtenida frente a 10240 datos de entrada en 16QAM.

No importa el incremento de SNR, la curva que genera el Corrector, corrige todos los datos que envía el amplificador, generando en la curva del Polinomio vs Amplitud (Figura 4.5) la inversa necesaria, para poder generar la compuesta, que permite mitigar casi absolutamente el ruido no lineal del amplificador de potencia.

Para el caso de 64QAM, los resultados obtenidos son aun más satisfactorios, tal como se aprecia en la Figura 5.10 el BER arrojado por el corrector, influye positivamente en el ruido que genera el amplificador de potencia, y solo se está inmerso en el ruido del canal, por lo que los bits recibidos errados son muy bajos.

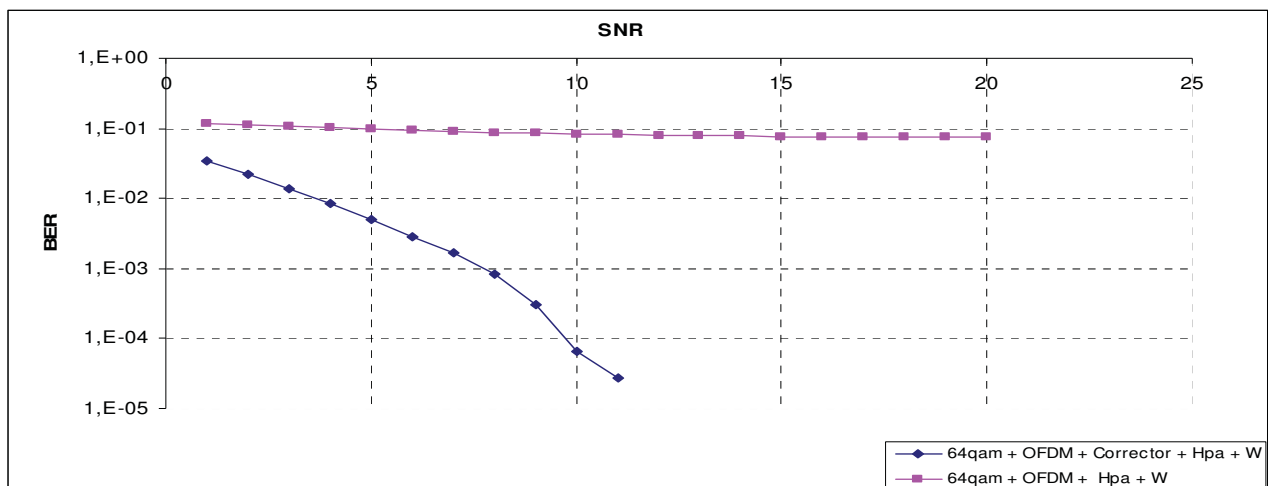


Figura 5.10 Curva BER vs SNR. Sistema con Corrector Neuronal

Por lo tanto, el resultado obtenido para 16QAM y 64QAM es muy satisfactorio, en ambos caso se corrige casi totalmente el error generado por el amplificador de potencia, esto es fácilmente graficado en las curvas de los valores del polinomio, tal como se aprecia en la Figura 5.11.

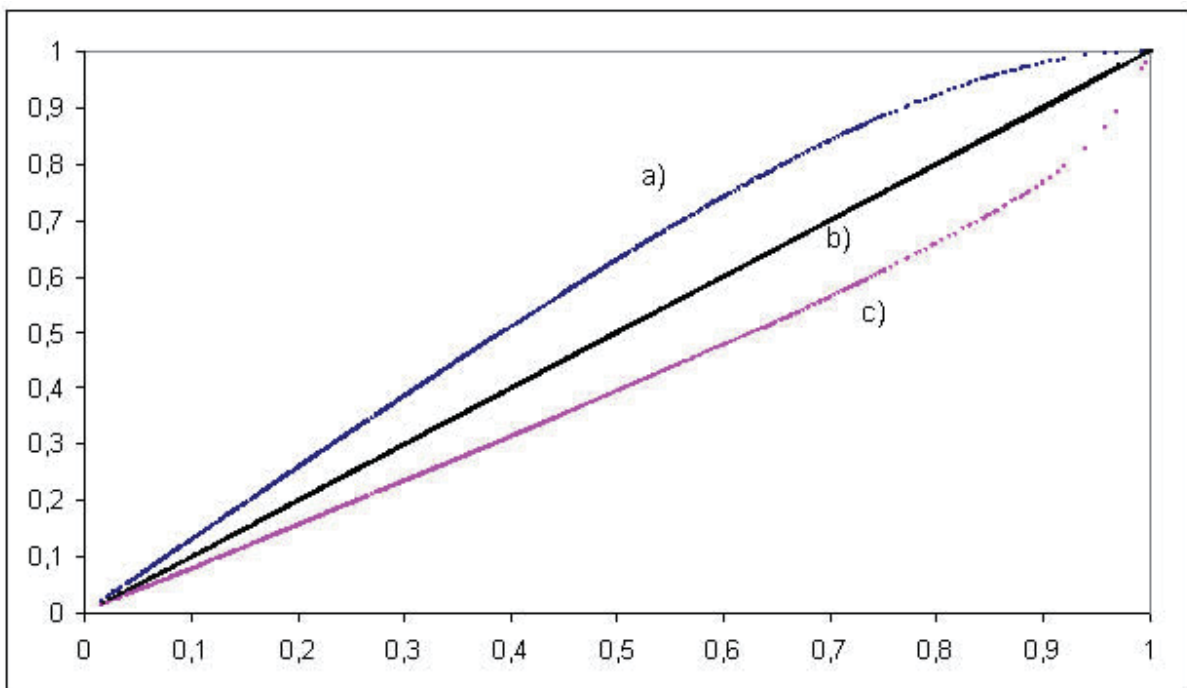


Figura 5.11 Curva de Amplitud vs valor de los Polinomios, a) Polinomio del HPA, b) Compuesta entre a) y c), y c) Inversa de a).

Conclusiones

Esta investigación se ha basado, como primera parte, en la comprensión del problema, por lo mismo se profundizó, estudió y comprendió los tópicos fundamentales para poder solventar una solución adecuada para el problema de distorsión no lineal de los amplificadores de potencia en pos de la utilización de las Redes de Kohonen. Se profundizó en temas tales como la modulación OFDM, y la forma en que esta es ocupada para sistemas de banda ancha, especialmente sistemas de comunicación inalámbrica.

Un algoritmo no supervisado trata de un elemento que por si solo es capaz de buscar las soluciones que se le exige, de este modo, se estudio el algoritmo VQTAM, técnica basada en Mapas autoorganizativos, específicamente las Redes de Kohonen, y esta técnica muestra varias opciones para poder determinar los vectores de los pesos de salida de cada neurona, de este modo, se estudio también el metodo de los Mínimos Cuadrados, el cual es una de las tantas alternativas ofrecidas por VQTAM. Se visualizó la problemática que se presentaba en los amplificadores de potencia en sistemas basados OFDM, y como la distorsión no lineal de estos dispositivos atenta contra la eficiencia de la tecnología inalámbrica. Se acoto más el campo de trabajo solamente a amplificadores terrestres.

Se implementó en JAVA un sistema capaz de modular señal OFDM, amplificar la señal, y también, una red neuronal SOM para corregir los errores producidos por la distorsión no lineal del amplificador de potencia, para lo cual se profundizó en características y elementos necesarios para poder entender la filosofía de esta técnica de Red Neuronal, que consta de un aprendizaje no supervisado por parte de las neuronas artificiales de la red, las cuales son capaces de reconocer el error, y presentarlo, por medio del reconocimiento de patrones.

La idea principal es tomar tal información y corregir el problema, linealizando la distorsión generada por la amplificación de la señal. Para esto sea ha evaluado un polinomio, y se ha comprendido la forma en como trabaja una modulación OFDM usando un amplificador de potencia sin la optimización de una red neuronal.

Para realizar la linealización del polinomio del amplificador, utilizando la técnica de VQTAM, se obtuvo dos tipos de sistemas, diferenciados en la forma de obtener los vectores de los pesos de salida. Así, se comparó ambas técnicas, y se eligió la más optima. Dados los resultados del sistema computacional, la mejor técnica, es el sistema que esta basado en la obtención de los vectores de pesos de salida por medio de Mínimos Cuadrados, técnica alternativa que el autor de VQTAM plantea [BARR01], donde se elimina casi por completo la distorsión generada por el polinomio del amplificador. Con la utilización de Mínimos Cuadrados se logra obtener, como ejemplo, 2 db a un $BER = 1.27 \cdot 10^{-3}$ reduciendo casi por completo el

ruido, y para el caso de VQTAM implementado de la manera natural, es decir, con entradas y salidas dinámicas, se obtiene 1,5 db a un $BER = 1.27 \cdot 10^{-3}$.

Por último, cabe mencionar la satisfacción personal durante este periodo comprendido en la investigación, implementación y comprensión de técnicas y tecnologías de los sistemas de comunicación, y la ampliación de conocimiento logrado, así como también la satisfacción por obtener y alcanzar los objetivos propuestos.

Bibliografía

- [ALON01] Alonso Zarate, Luís Gonzaga, “*Técnicas de acceso y gestión de recursos para garantizar calidad de servicio en sistemas de comunicaciones móviles basados en CDMA*”. Tesis, Universitat Politecnica de Catalunya. 2001.
- [BARR01] Guilherme Barreto, Aluizio Araújo. “Identification and Control of Dynamical Systems Using The Self-Organizing Map”. IEEE: Transactions on Neural Networks, Vol. 15, N° 5. Septiembre 2004.
- [CAST01] A. Castellano, C. Concejero. “*Introducción a las Redes Neuronales y su aplicación a las Búsquedas de documentos*”. Boletín Factores Humanos, ISSN 1135-2450, N°. 19, 1999.
- [CATI01] Enrique Castillo, Antonio Consejo, Pablo Pedregal, Ricardo García, Natalia Alguacil. “*Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia*”. Universidad de Castilla-La Mancha. Versión en Español de "Building and Solving Mathematical Programming Models in Engineering and Science", Pure and Applied Mathematics Series, Wiley, New York. 2002
- [CHAN01] R. Chang, R. Gibby. “A Theoretical Study of Performance of an Orthogonal Multiplexing Data Transmission Scheme”. ISSN: 0096-2244, Communications, IEEE Transactions (1988), Páginas 529 a 540. Publicado originalmente en 1968. IEEE 1988.
- [DURA01] Josué Duran, “*Optimización por Enjambre de Partículas para Reducción de distorsión no lineal en sistemas OFDM*”. Memoria de título Pontificia Universidad Católica de Valparaíso. Diciembre 2007.
- [FREE01] James Freeman, David Skapura. “*Neural Networks Algorithms, Applications and Programming Techniques*”. Addison-Wesley Publishing Company. 1991.
- [FYFE02] Colin Fyfe “*Artificial Neural Networks*”. Department of Computing and Information Systems, The University of Paisley. Room L117. Edition 1.1, 1996

- **[GARC01]** M. García, M. Magret. "Introducción a la teoría de matrices polinomiales". Ediciones UPC, 2º Edición. 1998.
- **[GARC02]** M. García. "Álgebra Lineal: Problemas Resueltos". Segunda Edición. Ediciones UPC. 1º Edición, 1994.
- **[IBOR01]** A. Ibort, M. Rodríguez. "Notas de Álgebra Lineal". Documento para estudiantes del Departamento de Matemáticas, Universidad Carlos III de Madrid. Primera Edición. 2001.
- **[LUE01]** David Luengo García. "*Análisis de un Esquema Novedoso de Comunicaciones Caóticas con OFDM y su Aplicación en Comunicaciones Seguras*". Tesis Doctoral, Universidad de Cantabria, Departamento de Ingeniería de Comunicaciones. 2004.
- **[LUNA01]** Marco Luna, Jaime Sanchez. "CDMA Multiportadora en redes inalámbricas de banda ancha para interiores". Instituto Tecnológico de Chihuahua, Mexico. División de Estudios de Postgrado e Investigación CICESE, Memoria del XXIII Congreso Internacional de Ingeniería Electrónica, ELECTRO 2001, CIC202-01. 2001.
- **[PROA01]** John Proakis. "Digital Communication". Editorial MCGRAW-HILL, Primera Edición, Estados Unidos. 2000
- **[RAMO01]** Francisco Ramos Pascual. "Medidas de distorsión no lineal en dispositivos de radiofrecuencia (Parte I)". Artículo de Estudio y Tesis de la Universidad de Granada, Adaptación para Revista Gigatronic nº15, Edición GM2 Publicaciones Técnicas, S.L. España. 2003.
- **[ROJA01]** Alejandra rojas, "Análisis de Distorsión No Lineal para WLAN". Memoria de título Pontificia Universidad Católica de Valparaíso. Agosto 2005.
- **[SERR01]** Serrano Cinca, Carlos. Artículo "*Self-organizing Maps for Initial Data Analysis: Let Financial Data Speak for Themselves*". Visual Intelligence in Finance using Self-organizing Maps, Editorial Springer Verlag. 1998.
- **[SIEG01]** Siegfried Grob, University of Ulm, "*Modeling Nonlinearities in Communication System and Mitigating their effects*". Proyecto de adquisición del grado de Doctor de la University of Ulm. 2005.

- [SIEG02] Siegfried Grob, Jürgen Lindner “Polynomial Model Derivation of Nonlinear amplifiers”. Advances in Wireless Communications: Enabling Technologies for 4G de Mobile eConference 2004. 2005.
- [TRAN01] Rodolfo Lorenzo Bautista, Universidad Nacional Autónoma de Mexico, “Sistemas eléctricos de potencia II. Teoría y problemas”. Publicado por la Facultad de Ingeniería. UNAM. 2007, 450 p. 2006.

Código Fuente

A continuación se escriben las funciones más importantes del código fuente para realizar el experimento mencionado en el punto 5. Código realizado en Java.

Algunas funciones como: lectura de bits, asignación de bits, paso de bits a QAM, paso de señal OFDM a QAM, paso de QAM a bits, han sido reducidas de este código fuente, y sólo se agrega una parte por cada una, esto, por que son muy largas y repetitivas.

Inicializador de Variables

```
FileReader f = new FileReader("bits_pocos2.txt");
BufferedReader br = new BufferedReader(f);
int [] BITS_m = new int [44830];
int [] BITS_final = new int [44830];
int p_bits=0;
double var_real=0, var_imag=0;
double BER = 0;
double BERqam = 0;
double EVM = 0, EVMd=0;
```

Inicialización de variables VQTAM (Algoritmo de autoaprendizaje)

```
int Neuronas =10;
int T = 80;
double a, a0, aT;
    a0 = 1;
    aT = Math.pow (10, -5);
double o, o0, oT;
    o0 = 250;
    oT = Math.pow (10, -5);
double [][] Wi = new double [2][Neuronas];
```

Inicialización de Variables para Mínimos Cuadrados – QR

```
int varX=49, i=0, n=0, j=0, k=0, BITS=0;
int numeroW = 10;
int n_filas=64, m_columnas=numeroW;
double [][] matrizA = new double [n_filas][m_columnas];
double [][] B = new double [n_filas][1];
double [][] X= new double [m_columnas][1];
double SNR=10;
```

Inicialización de variables para la IFFT

```
double [] Xreal = new double [64];
double [] Ximag = new double [64];
double [] F = new double [64];
double [] A = new double [64];
double ereal=0, eimag=0, Zreal=0, Zimag=0, Amax=0, temp=0, itemp, qtemp;
int N=64,w,p=0;
int v=0; //hasta 117
```

Inicialización de variables para el HPA

```
double [] hpaREAL = new double [N];
double [] hpaIMAG = new double [N];
double [] hpaTEMP = new double [N];
double P=0;
double c1,c3,c5,c10,c30,c50;
double A_norm = 0;
```

ALGORITMO: Modulador Banda Base

```
double var_teta = 2*Math.PI/N;
Amax=0;
for (k=0;k<64;k++)
{
    Xreal[k] = 0;
    Ximag[k] = 0;
```

```

    itemp = 0;
    qtemp = 0;

    A[k]=0;
    for (w=(0 ); w<(64);w++)
    {
        ereal = Math.cos ( var_teta* k * w );
        eimag = Math.sin ( var_teta* k * w );
        Xreal[k] += (matrizncomplejos [0][w + (v*64)]*ereal - matrizncomplejos [1][w +(v*64)]*eimag)/N;
        Ximag[k] += (matrizncomplejos [1][w + (v*64)]*ereal + matrizncomplejos [0][w +(v*64)]*eimag)/N;
    }
    Xreal[k] = Xreal[k] * Math.sqrt(N);
    Ximag[k] = Ximag[k] * Math.sqrt(N);

//Calculo de FASE de X[k]
    itemp = Xreal[k];
    qtemp = Ximag[k];
    temp=0;
    if ( itemp == 0 ) itemp = 1;
    if ( itemp < 0 ){
        temp = -Math.PI;
    }
    F [k] = Math.atan(qtemp/itemp) + (-temp) ;

// Calculo Amplitud X[k]
    A[k] = Math.sqrt (Math.pow( Xreal[k], 2) + Math.pow(Ximag[k], 2));
    Amax = Math.max( A[k],Amax );

```

ALGORITMO: VQTAM – Kohonen – Algoritmo no supervizado

```

//Inicializando pesos de neuronas
System.out.println ("Neuronas: "+Neuronas+" Iteraciones: "+T+"");
for (i=0;i<Neuronas;i++)
{
    Wi[0][i]= 1;
    Wi[1][i]=Math.random(); }

```

```

//Inicializando Número de iteraciones
double h;
int t;

//Inicializando entrada X
double [] Xout = new double [64];
double [] Xin = new double [64];
for(k=0;k<64;k++)
{
    Xin [k] = hpaTEMP[k];

    Xout [k] = A[k]/Amax;
}

//Ejecución de Red Neuronal
double temp_i_ganadora=1;
int i_ganadora = 0, k_neurona=0;
double temp_xin=0, temp_xout=0;
for (t=0;t<T;t++)
{

    for (k=0;k<64;k++)
    {

//Busqueda de Neurona ganadora
        temp_i_ganadora=0;
        for (i=0;i<Neuronas;i++)
        {
            if (i==0)
            {
                i_ganadora = 0;
                temp_i_ganadora = Math.abs (Xin [0] - Wi[0][i]);
            }else{
                if (Math.abs (Xin [k] - Wi[0][i])< temp_i_ganadora)
                {
                    i_ganadora = i;
                    temp_i_ganadora = Math.abs (Xin [k] - Wi[0][i]);
                }
            }
        }
    }
}

```

```

    }
}

//Actualización de pesos
for (i=0;i<Neuronas;i++)
{
a=a0*Math.pow(aT/a0, t/T) ;
o=o0*Math.pow(oT/o0, t/T);

//Función del Vecino Gausiano
h = 1 / Math.exp( Math.pow (Math.abs(i-i_ganadora ),2)/(2*Math.pow (o,2)));
//W in
temp_xin=(Xin [k] - Wi[0][i]);
Wi[0][i] = a*h*(temp_xin);
//W out
temp_xout=(Xout [k] - Wi[1][i]);
Wi[1][i] = a*h*(temp_xout);
}
}
}

```

ALGORITMO: Mínimos Cuadrados

```

//Llenado de Matriz A y B
for(i=0;i<n__filas;i++)
{
for(j=0;j<m__columnas;j++)
{
matrizA [i][j]=Math.pow((hpaTEMP[i] ,(j+1)));
}
B[i][0]= A[i]/Amax;
}

// Método de Gram–Schmidt
//U = u1,u2...
double [][] U = new double [n__filas][m__columnas];

```

```

double norma, ppunto, utemp;
int columna;
//Calculo de U
    double [] Un = new double [n__filas];
//Llenando U
    for (j=0;j<m__columnas;j++)
    {
        for (i=0;i<n__filas;i++)
        {
            if (j>0)
            {
                U[i][j]= matrizA [i][j];

                for (columna=0;columna<j;columna++)
                {
                    norma=0;
                    for (w=0;w<n__filas;w++)
                    {
                        norma=Math.pow(U[w][columna],2)+norma;
                    }
                    ppunto=0;
                    for (w=0;w<n__filas;w++)
                    {
                        ppunto=matrizA [w][j]*U[w][columna]+ppunto;
                    }
                    utemp=ppunto/norma;
                    U[i][j]= U[i][j]-utemp*U[i][columna];
                }

                Un [j] += Math.pow (U[i][j], 2);
            }else{
                U[i][j]= matrizA [i][j];
                Un [j] += Math.pow (U[i][j], 2);
            }
        }
    }
    Un [j] = Math.sqrt (Un [j]);

```

```
}

// Calculo de Q
//Q = u1/||u1||, u2/||u2||
double [][] Q = new double [n__filas][m__columnas];

//Llenando Q
for (j=0;j<m__columnas;j++)
{
  for (i=0;i<n__filas;i++)
  {
    Q[i][j]= U[i][j]/ Un [j];
  }
}

//Calculo de R
int j_temp=0;
double temp_matriz=0;

//Transpuesta de Q

//Deifinir QT
double [][] QT = new double [m__columnas][n__filas];
for (j=0;j<n__filas;j++)
{
  for (i=0;i<m__columnas;i++)
  {
    QT[i][j]= Q[j][i];
  }
}

//Multiplicacion de QT y A
//R=QT *A
double [][] R = new double [m__columnas][m__columnas];
for (i=0;i<m__columnas;i++)
{
```

```

for (j=0;j<m__columnas;j++)
{
    R[i][j]=0;
}
}
for(i=0;i<m__columnas;i++)
{
for(j=0;j<m__columnas;j++)
{

temp_matriz=0;
for(j_temp=0;j_temp<n__filas;j_temp++)
{
temp_matriz=QT[i][j_temp]*matrizA [j_temp][j]+temp_matriz;

}
if(i<j+1)
R[i][j]=temp_matriz;
}
}

//Calculo de QT*B
double [][] qB = new double [m__columnas][1];
for(i=0;i<m__columnas;i++)
{
for(j=0;j<1;j++)
{

temp_matriz=0;
for(j_temp=0;j_temp<n__filas;j_temp++)
{
temp_matriz=QT[i][j_temp]*B[j_temp][j]+temp_matriz;

}
qB[i][j]=temp_matriz;
}
}

```



```

}

// Obtención de X
for (i=0;i<m__columnas;i++)
{
if(i<1)
{
X[m__columnas-i-1][0]= qB[m__columnas-i-1][0]/R[m__columnas-i-1][m__columnas-1];
}else{
temp_matriz=0;
for(j=0;j<i;j++)
{
temp_matriz=R[m__columnas-i-1][m__columnas-j-1]*X[m__columnas-j-1][0]+temp_matriz;
}
X[m__columnas-i-1][0]=(qB[m__columnas-i-1][0]-temp_matriz)/R[m__columnas-i-1][m__columnas-i-1];
}
}
}

```

ALGORITMO: Corrector Neuronal

```

double [] A_nn = new double [64];
double Amax_nn=0;
for (k=0;k<64;k++)
{
A_norm=A[k]/Amax;
A_nn[k]=0;
for (i=0;i<m__columnas;i++)
{

A_nn[k]= A_nn[k] + X[i][0]*Math.pow((A_norm) ,(i+1));
}
if (A_nn[k]>1)
{

A_nn[k]=1;
}
Amax_nn = Math.max( A_nn[k],Amax_nn );
}

```

```
}
```

ALGORITMO: Amplificador Sin Corrector

```
//Amplificador - MGA52543
```

```
  N=64;
  c1 = 5.6234;
  c3 = -0.8913;
  c5 = -6.1441;
  c10 = 1.2895;
  c30 = 0.0790;
  c50 = 0.2105;

  for (k=0;k<64;k++)
  {
    hpaREAL [k] = 0;
    hpaIMAG [k] = 0;
    A_norm = A[k]/Amax;
    P = c10*A_norm - c30*Math.pow(A_norm,3) - c50*Math.pow(A_norm,5);
    hpaREAL [k] = Math.cos(F[k])*P*Amax;
    hpaIMAG [k]= Math.sin(F[k])*P*Amax;
  }
```

ALGORITMO: Amplificador con Corrector

```
//Amplificador - MGA52543
```

```
  N=64;
  c1 = 5.6234;
  c3 = -0.8913;
  c5 = -6.1441;
  c10 = 1.2895;
  c30 = 0.0790;
  c50 = 0.2105;
  for (k=0;k<64;k++)
  {
```

```
    hpaREAL [k] = 0;
    hpaIMAG [k] = 0;
    A_norm = A_nn [k]/Amax_nn;
    P = c10*A_norm - c30*Math.pow(A_norm,3) - c50*Math.pow(A_norm,5);
    hpaREAL [k] = Math.cos(F[k])*P*Amax;
    hpaIMAG [k]= Math.sin(F[k])*P*Amax;
}
```

ALGORITMO: Canal Gaussian

```
double cg_R=0,cg_a=0,cg_RR=0,cg_b=0,cg_temp=0;
SNR=21;
for (k=0;k<64;k++)
{

    cg_a=Math.random();
    cg_R=Math.sqrt(1/(2*Math.pow(10,SNR/10)));
    cg_RR=Math.sqrt(2*Math.log(1/(1 - cg_a)));
    cg_temp=Math.sin(2*Math.PI*cg_a);
    hpaREAL [k] = hpaREAL [k]+ cg_R*cg_RR*cg_temp;

    cg_b=Math.random();
    cg_R=Math.sqrt(1/(2*Math.pow(10,SNR/10)));
    cg_RR=Math.sqrt(2*Math.log(1/(1 - cg_b)));
    cg_temp=Math.sin(2*Math.PI*cg_b);
    hpaIMAG [k] = hpaIMAG [k]+ cg_R*cg_RR*cg_temp;

}
```

ALGORITMO: Demodulador Banda Base

```
dREAL[k]=0;
dIMAG[k]=0;
```

```
for ( i=0; i<N; i++) {
    double q = var_t*k*i;
    dREAL[k] += (hpaREAL[i]*Math.cos(q) - hpaIMAG[i]*Math.sin(q));
    dIMAG[k] += (hpaIMAG[i]*Math.cos(q) + hpaREAL[i]*Math.sin(q));
}
dREAL[k] = dREAL[k]/Math.sqrt(N);
dIMAG[k] = dIMAG[k]/Math.sqrt(N);
```

ALGORITMO: Calculo EVM

```
EVM= Math.pow(dREAL [k] - matrizncomplejos [0][k + (v*64)],2)+Math.pow(dIMAG [k] - matrizncomplejos
[1][k + (v*64)],2) + EVM;
EVMd = Math.pow(matrizncomplejos [0][k + (v*64)] + matrizncomplejos [1][k + (v*64)],2) + EVMd;
EVM = Math.sqrt(EVM/EVMd)*100;
```

ALGORITMO: Calculo VER

```
for(i=0;i<p_bits;i++)
{
    if(BITS_m[i]!=BITS_final[i])
    {BER++;
    }
}
BER=BER/(p_bits);
```

Asignación de símbolos QAM (para 16QAM)

```
//Caso 1: Real
if (dREAL[k]>=2)
    var_real=3;
if(dREAL[k]>=0 && dREAL[k]<2)
    var_real=1;
if(dREAL[k]>=-2 && dREAL[k]<0)
```

```

                                var_real=-1;
    if(dREAL[k]<-2)
                                var_real=-3;
//Caso 2 Imaginario
    if (dIMAG[k]>=2)
                                var_imag=3;
    if(dIMAG[k]>=0 && dIMAG[k]<2)
                                var_imag=1;
    if(dIMAG[k]>=-2 && dIMAG[k]<0)
                                var_imag=-1;
    if(dIMAG[k]<-2)
                                var_imag=-3;

```

ALGORITMO: Asignador de Bits (solo un fragmento, lo que sigue es casi igual a la muestra)

```

    if(var_real===-3 && var_imag===-3) // -3 -3
    {
        BITS_final[p_bits]=1;
        BITS_final[p_bits+1]=1;
        BITS_final[p_bits+2]=1;
        BITS_final[p_bits+3]=0;
    }
    if(var_real===-1 && var_imag===-3) // -1 -3
    {
        BITS_final[p_bits]=1;
        BITS_final[p_bits+1]=1;
        BITS_final[p_bits+2]=1;
        BITS_final[p_bits+3]=1;
    }

```

ALGORITMO: Obtención de Bits (16QAM)

```

for (j=0;j<n;j++)
{
    if (matriz[0][j]==1) //1xxx

```

```
{
  if (matriz[1][j]==1) //11xx
  {
    if (matriz[2][j]==1) //111x
    {
      if (matriz[3][j]==1) //1111-15
      {
        matriz015 [0][j] = 15;
        matrizncomplejos [0][j] = (-1);
        matrizncomplejos [1][j] = (-3);
      }
      if (matriz[3][j]==0) //1110-14
      {
        matriz015 [0][j] = 14;
        matrizncomplejos [0][j] = (-3);
        matrizncomplejos [1][j] = (-3);
      }
    }
  }
}
```