

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL.

Facultad de Ingeniería ELECTRICA y COMPUTACION

" MODELAMIENTO Y SIMULACION DE REDUCCION DE PEAK-TO-AVERAGE POWER RATIO UTILIZANDO TONOS RESERVADOS PARA SISTEMAS 802.11a "

TESINA DE GRADUACION

"ALGORITMO DE OPTIMIZACIÓN DE SISTEMAS MULTIPORTADORA"

Previa a la obtención del Título de:

INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES

Presentado por

Miguel Alejandro Cevallos Sáenz

Juan Carlos Pinto Bonilla

Guayaquil - Ecuador

2010

AGRADECIMIENTO

Primeramente a Dios, por darme fuerza para seguir adelante, a mi Familia por su apoyo incondicional.

Juan Carlos

A Dios por los caminos que me permite avanzar y aquellas personas cercanas que puso en mi vida especialmente para acompañarme, guiarme y apoyarme.

Miguel

DEDICATORIA

A Dios, a toda mi familia, amigos, compañeros, profesores y a todos aquellos que me ayudaron para poder realizar este proyecto.

Juan Carlos

A Dios, y especialmente a mis padres que siempre me brindaron su apoyo.

Miguel

TRIBUNAL DE SUSTENTACION

.....
Dr. Hernán Córdova

Profesor del Seminario

.....
Ing. Juan Carlos Aviles

Delegado del Decano

DECLARACION EXPRESA

“La responsabilidad por los hechos, ideas y doctrinas expuestas en este trabajo me corresponden exclusivamente; y, el patrimonio intelectual de la misma, a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL”

(Reglamento de exámenes y títulos profesionales de la ESPOL)

.....
Miguel Cevallos Sáenz

.....
Juan Carlos Pinto

RESUMEN

En el presente proyecto se selecciono y coloco los conceptos básicos necesarios para definir el PAPR y las variables y forma mas característicos del mismo, en la primeras secciones (y apéndices A al D), basados en los distintos conceptos e información disponible en gran cantidad en variados estudios publicados y que debió buscarse, revisarse y seleccionarse para adquirir la información de utilidad que ayudara en las explicaciones y quedaran como base para el presente estudio, es así que posteriormente (sección 1.4) se da la explicación de los efectos perjudiciales que puede traer, y la necesidad de técnicas de corrección.

En la sección 2 dado que se usara la técnica de tonos reservados, se escogió un Standard (el 802.11a) en el cual halla mayor posibilidad de visualizar el uso y resultado de la técnica, se condiciona/limita en cierta forma el proyecto para no caer en análisis de demasiados parámetros que resulten en una extensión grande del mismo o estudio de muchos detalles al mismo tiempo, se describe el funcionamiento básico de tonos reservados

Pero necesitaba implementarse la técnica y al haber varias formas de hacerlo se busco 2 posibilidades, estos fueron los algoritmos gradiente y de pulso gaussiano, es así que en la sección 3 se hace lo posible en explicar en detalle las bases y el funcionamiento de las mismas.

Sección 4 es la simulación de tonos reservados, mediante los 2 algoritmos mencionados, para esto se tomo como punto de partida un código de creación de señales para 802.11a (Krishna) y se realizo las secciones de corrección del PAPR en el código (en base al funcionamiento detallado en la seccion3) y se busco la generación de todas las graficas y tablas que se observan en esa sección de simulación tratando de obtenerse datos de los resultados de las correcciones de PAPR hecha por dichos algoritmos (algoritmo gradiente y pulso gaussiano) .

Sección 5 muestra análisis de resultados aprovechando la corrida del algoritmo implementado en la sección de simulación y variaciones del mismo, para observar detalles, ver que tan efectiva resulta la corrección y comparar técnicas.

ÍNDICE GENERAL

	Pág.
RESUMEN.....	VI
ÍNDICE GENERAL.....	VIII
ABREVIATURAS.....	XI
ÍNDICE DE FIGURAS.....	XIII
INTRODUCCIÓN.....	1

CAPITULO 1

1.- ANALISIS GENERAL DEL PROBLEMA	3
1.1.- Relación de Potencia Pico a Promedio (PAPR)	3
1.2.- Valor crítico de PAPR	5
1.3.- Aproximación empírica de la distribución del PAPR	8
1.4.- Efectos relacionados al PAPR.....	9
1.4.1.- Problemas del PAPR con respecto al hardware.....	9
1.4.2.- Que sucede sino se hace nada al respecto?.....	22

CAPITULO 2

2.- DEFINICION DEL ESCENARIO DEL PROYECTO	29
2.1.- Estándar y limitaciones	29

2.2.- Tonos Reservados	30
------------------------------	----

CAPITULO 3

3.- ALGORITMOS PARA REDUCIR EL PAPR UTILIZANDO TONOS

RESERVADOS	36
-------------------------	-----------

3.1.- Algoritmo de gradiente simple con rápida convergencia	36
---	----

3.1.1- Descripción de iteraciones con algoritmo gradiente	44
---	----

3.2.- Reducción de PAPR mediante Pulso Gaussiano.....	45
---	----

3.2.1.- Pasos iterativos con Pulso Gaussiano	56
--	----

CAPITULO 4

4.- SIMULACION	60
-----------------------------	-----------

4.1.- Descripciones varias	60
----------------------------------	----

4.2.- Observaciones valor UMBRAL.....	63
---------------------------------------	----

4.3.- Reducción con algoritmo gradiente	69
---	----

4.4.- Reducción con pulso Gaussiano	75
---	----

4.5.- Valores comparativos de ambos algoritmos.....	77
---	----

CAPITULO 5

5.- ANALISIS DE RESULTADOS	86
5.1.- Resumen	86
5.2.- Respecto al número de tonos	88
5.3.- Existe relación entre la cantidad de PAPR excesivo a corregir y el número de iteraciones?	94
5.4.- Comparación de resultados con ambos algoritmos.....	96

CONCLUSIONES Y RECOMENDACIONES

APENDICES

BIBLIOGRAFÍA

ABREVIATURAS

ADC	Analog-to-Digital Converter
OFDM	Orthogonal Frequency Division Multiplexing
WI-FI	Wireless Fidelity
IEEE	Institute of Electrical and Electronics Engineers
WiMAX	Worldwide Interoperability for Microwave Access (interoperabilidad mundial para acceso por microondas)
FFT	Fast Fourier Transform
PAPR	Peak-to-Average Power Ratio
HPA	High Power Amplifier
CDF	Cumulative Distribution Function
CCDF	Complementary Cumulative Distribution Function
DAC	Digital Analogue Converter
IBO	Input Back-Off

OBO	Output Back-Off
DC	Direct Current
PAE	Power Added Efficiency
SNR	Signal to Noise Ratio
SDR	Signal to Distortion power Ratio
IMD	Intermodulación
BPSK	Binary Phase Shift Keying
DTF	Discrete Fourier Transformation
DMT	Discrete Multi-Tone

ÍNDICE DE FIGURAS

Pag.

CAPITULO 1

Figura 1. 1. <i>Relación de potencia pico a promedio</i>	4
Figura 1. 2. <i>Función de Distribución Acumulativa Complementaria CCDF para un sistema OFDM de diferentes N</i>	8
Figura 1. 3. <i>Linealidad característica de un HPA</i>	13
Figura 1. 4. <i>Gráfica de PAE vs potencia de entrada</i>	16
Figura 1. 5. <i>Relación entre Eficiencia y PAPR en un HPA</i>	19
Figura 1.6. <i>Repuesta espectral a la inyección de 2 tonos en un amplificador con distorsión no lineal</i>	26

CAPITULO 2

Figura 2.1. <i>Generación de una señal $c(t)$ para reducir picos altos de $x(t)$</i>	31
Figura 2.2. <i>Reservación de tonos para formar c</i>	33

CAPITULO 3

Figura 3.1. Esquema del algoritmo gradiente.....	43
Figura 3.2. Representación del algoritmo gradiente de SDR para un primer paso.....	45
Figura 3.3. Pulso Gaussiano	47
Figura 3.4. Pulso Gaussiano en el dominio de la frecuencia y en el dominio del tiempo	51
Figura 3.5. Esquema de Pulso Gaussiano para reducir PAPR.....	59

CAPITULO 4

Figura 4.1. Pasos de transmisión de OFDM con señal correctora c	61
Figura 4.2. Pasos para corrección.....	62
Figura 4.3. Número aproximado de símbolos(de los 209) que necesitan corrección para distintos valores del UMBRAL.....	67
Figura 4.4. Núm de iteraciones promedio que debe hacerse para corregir un símbolo con pico excesivo , para distintos valores de UMBRAL	67
Figura 4.5. Aproximación e iteraciones de la reducción de pico para varios valores de N_{iu}	71
Figura 4.6. Reducción de picos utilizando algoritmo gradiente.....	73
Figura 4.7. Graficas de subportadoras utilizadas (frecuencia).....	74
Figura 4.8. Reducción de picos que sobrepasan el Umbral usando pulso Gaussiano	76

Figura 4.9. Subportadoras utilizadas (frecuencia) para pulso gaussiano .77

Figura 4.10. CCDF del papr para algoritmo gradiente y pulso gaussiano78

CAPITULO 5

Figura 5.1. Número de iteraciones promedio para realizar corrección según ntonos89

Figura 5.2. Pulso gaussiano de la señal correctora c en frecuencia y tiempo92

Figura 5.3. Gráfica de numero de tonos con respecto al exceso de PAPR que se corrigió para los algoritmo gradiente y pulso gaussiano95

Figura 5.4. Cuantas veces cada técnica se acercó mas a la corrección deseada.....100

Figura 5.5. Veces en que cada tecnica converge mas rapido101

ÍNDICE DE TABLAS

Pag.

Tabla I. <i>Carateristicas de tonos reservados.</i>	35
Tabla II. <i>Resultados de simulacion reorganizados.</i>	82
Tabla III. <i>Reorganizacion de resultados de simulacion.</i>	97

INTRODUCCION

Los sistemas multiportadoras están actualmente muy extendidos tanto en sistemas de transmisión cableados como inalámbricos (por ejemplo, Orthogonal Frequency Division Multiplexing (OFDM), utilizado en aplicaciones WiFi, IEEE 802.11, y WiMAX, IEEE 802.16). En este reporte nos enfocaremos en sistemas WI-FI 802.11a.

OFDM (para mas detalles ver APENDICE A) ofrece muchas ventajas en sistemas multiportadoras, alta transmisión de datos, alta eficiencia espectral, implementación simple de Transformada de Fourier Rápida (FFT), inmune a desvanecimiento multitrayectoria. Sin embargo presenta inconvenientes cuando las subportadoras independientes se alinean en fase generando así altos picos de potencia lo que ocasiona serios problemas al sistema.

Desde el punto de vista de la eficiencia energética en el transmisor, operar con señales que presentan una alta relación entre potencia de pico y potencia media (peak-to-average power ratio (PAPR), definida mas adelante) resulta sumamente ineficiente.

Para que el amplificador de alta potencia (HPA por sus siglas en ingles) del transmisor trabaje en zona lineal, es necesario operar con niveles potencia de reserva similares o por encima del nivel de PAPR, originando así una ineficiencia energética en el HPA, aspecto crítico especialmente en equipos móviles operados con baterías. Para mitigar la ineficiencia asociada a trabajar con señales con gran PAPR, existen técnicas de reducción y en nuestro proyecto tomaremos la llamada “tonos reservados” que se analizará mas adelante.

CAPÍTULO 1

1. ANALISIS GENERAL DEL PROBLEMA

1.1. Relación de Potencia Pico a Promedio (PAPR)

Un nivel de PAPR alto ocasiona serios problemas y por tal motivo es un parámetro muy importante en el diseño de sistemas multiportadoras ortogonales, a continuación se presenta la definición del PAPR y su efecto cuando se torna crítico.

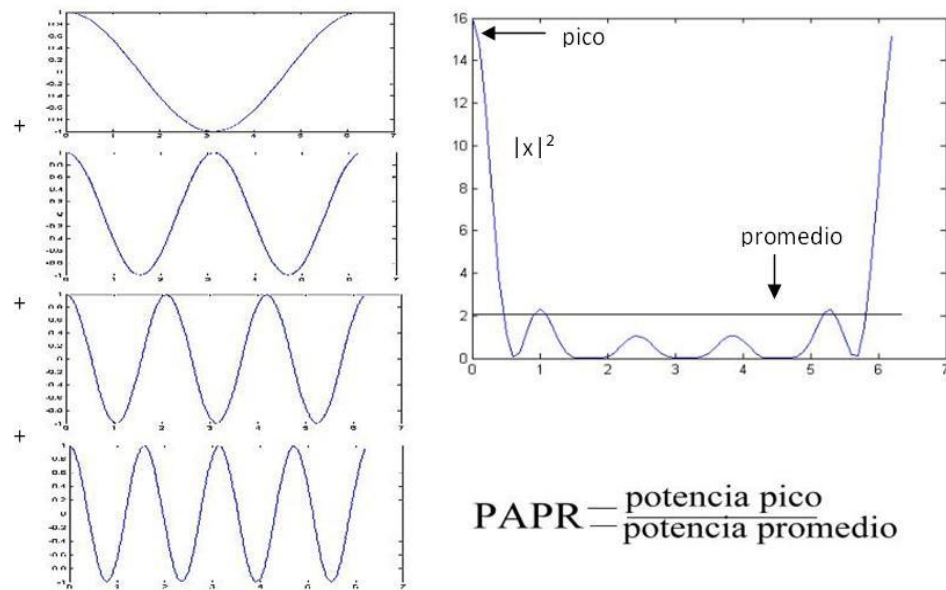


Fig. 1.1.- Relación de potencia pico a promedio

Tomando como referencia básica la relación entre el valor pico y el promedio de una señal (medida de variación de amplitud de una señal), se puede definir el *PAPR* con la siguiente expresión en un intervalo de tiempo T (ref [1] pag8):

$$PAPR = \frac{\max_{t \in T} |x(t)|^2}{E\{|x(t)|^2\}} \quad (1.1)$$

Donde el numerador es la máxima potencia instantánea de la señal OFDM que se tiene en el rango sobre el cual se esta haciendo la medición y $E\{|x(t)|^2\}$ es el valor esperado de la magnitud de la envolvente $x(t)$ al cuadrado y se la puede hallar para una señal en banda base de la siguiente manera:

$$E\{ |x(t)|^2 \} = \frac{1}{T} \int_{t=0}^T |x(t)|^2 dt \quad (1.2)$$

Lo cual esta expresado para un periodo de tiempo T continuo, pero puede expresárselo de manera similar para el caso discreto de una señal con N muestras así:

$$E\{ |x_n|^2 \} = \frac{1}{N} \sum_{n=0}^{N-1} |x_n|^2 \quad (1.3)$$

1.2. Valor Crítico de PAPR

Con la definición de *PAPR* podemos analizar el peor de los casos, o sea cuando este parámetro tiene su valor crítico (alto PAPR) y cual es su consecuencia.

La señal OFDM es la suma de múltiples sinusoidales con una frecuencia de separación $\frac{1}{T}$ donde cada senoide es modulada por una información independiente X_k [3]. Matemáticamente tenemos la señal a transmitir:

$$x(t) = \sum_0^{K-1} X_k e^{\frac{j2\pi kt}{T}} \quad (1.4)$$

Por simplicidad asumamos que $X_k = 1$ para todas las subportadoras.

Tenemos entonces el valor pico de la señal:

$$\max[x(t)x(t)^*] = \max \left[\sum_0^{K-1} X_k e^{\frac{j2\pi kt}{T}} \sum_0^{K-1} X_k e^{\frac{-j2\pi kt}{T}} \right] \quad (1.5)$$

$$= \max \left[X_k X_k^* \sum_0^{K-1} \sum_0^{K-1} e^{\frac{j2\pi kt}{T}} e^{\frac{-j2\pi kt}{T}} \right] \quad (1.6)$$

$$= K^2 \quad (1.7)$$

El valor cuadrático medio de la señal es:

$$E[x(t)x(t)^*] = E \left[\sum_0^{K-1} X_k e^{\frac{j2\pi kt}{T}} \sum_0^{K-1} X_k e^{\frac{-j2\pi kt}{T}} \right] \quad (1.8)$$

$$= E \left[X_k X_k^* \sum_0^{K-1} \sum_0^{K-1} e^{\frac{j2\pi kt}{T}} e^{\frac{-j2\pi kt}{T}} \right] \quad (1.9)$$

$= K^2$, por el valor esperado dividimos para

K (número total de portadoras),

$$= K \quad \text{obtenemos lo siguiente:} \quad (1.10)$$

De manera que el PAPR de un sistema OFDM con K subportadoras y que todas las subportadoras están dadas en la misma modulación,

es:

$$PARP = \frac{K^2}{K} = K \quad (1.11)$$

K sería el pico máximo de PAPR cuando todas las subportadoras son igualmente moduladas, y todas ellas alineadas en fase.

Por ejemplo si $K= 52$ subportadoras, entonces el PAPR esperado es 52, expresado en decibeles:

$$PARP_{dB} = 10\log_{10}(PAPR) \quad (1.12)$$

Reemplazando valores en (1.12)

$$PARP_{dB} = 10\log_{10}(52)$$

$$PARP_{dB} = 17.16dB$$

Por lo que este valor puede saturar la linealidad de los amplificadores de potencia o de requerir potencia extra para la transmisión cayendo en un sistema ineficiente (como se vera mas adelante).

De ahí, el principal motivo de reducir el PARP es el simple hecho de que no disponemos de potencia infinita, ya que nos vemos limitados a las capacidades de los amplificadores de potencia.

Una desventaja de OFDM es que el pico de la señal puede ser alta hasta N veces la potencia promedio (N es el número de portadoras).

Esos largos picos incrementan la cantidad de distorsión de intermodulación resultando en un incremento en la tasa de error. La potencia de la señal promedio debe de mantenerse baja de hecho

para prevenir la limitante del amplificador transmisor. Minimizando el PAPR permite que la más alta potencia promedio sea transmitida para una potencia pico fija, mejorando la totalidad de la relación señal ruido en el receptor. Por eso es importante reducir el PAPR.

1.3 Aproximación Empírica de la Distribución del PAPR

La probabilidad de que el PAPR sea menor o igual a un valor λ puede aproximarse por una función de distribución acumulativa (CDF) (ref [6] pag 20) que se escribe como:

$$\Pr\{PAPR[x(n)] \leq \lambda\} = [1 - \exp(-\lambda)]^N \quad (1.13)$$

Es interesante ver la curva de la CCDF

$$\Pr\{PAPR[x(n)] > \lambda\} = 1 - [1 - \exp(-\lambda)]^N \quad (1.14)$$

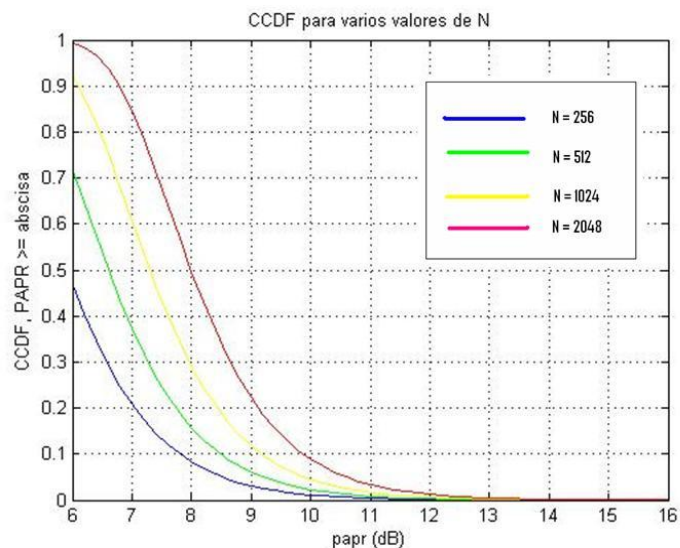


Fig. 1.2.-Función de Distribución Acumulativa Complementaria CCDF para un sistema OFDM de diferentes N

En la figura 1.2 se puede observar como la probabilidad de que el PAPR alcance valores cada vez mas grandes va disminuyendo hasta valores de probabilidad muy pequeños de que eso ocurra.

1.4 Efectos relacionados al PAPR

Como sabemos, en sistemas OFDM cuando el valor N es grande el símbolo OFDM puede aproximarse a una distribución de Gauss, lo cual implica que algunas muestras tengan probabilidad de presentar picos.

En la práctica la aparición de los altos valores de PAPR afecta en el sistema de transmisión resultando en hardware ineficiente y efectos en la tasa de error de bit al transmitir la señal.

1.4.1 Problemas del PAPR con respecto al hardware:

Existen dos dispositivos que son seriamente afectados: convertidores DAC/ADC y amplificadores de potencia HPA. Cada uno de estos dispositivos tienen diferentes efectos, a continuación se explica.

Convertidores D/A y A/D

El PAPR es una medida de variación del rango dinámico de las señales OFDM , es así que un alto valor de PAPR indica altas variaciones en el rango de la señal.

Los altos valores de PAPR incrementan la complejidad al implementar los convertidores DAC/ADC dado que los largos picos requieren representarse con mejor precisión

La principal fuente de distorsión de los DAC/ADC ´s es en el cuantizador (ref [4] pag 42). Para evaluar la cuantización es conveniente representar la señal de entrada en coordenadas cartesianas:

$$x = \Re\{x\} + j\Im\{x\} \quad (1.15)$$

Con dicha representación, la característica ideal de input-output de un cuantizador uniforme es:

$$Q(u) = \begin{cases} \Delta \text{round}(u / \Delta), & |u| \leq A \\ A \text{sign}(u), & |u| > A \end{cases} \quad (1.16)$$

Donde u representa $\Re\{x\}$ o $\Im\{x\}$, Δ es el tamaño de paso del cuantizador y A es el nivel de saturación. Este es un ejemplo de un cuantizador de redondeo con saturación.

Basándonos en este modelo vemos que los DAC/ADC's tienen un umbral definido, o sea un nivel saturación A. Si el nivel de PAPR supera este nivel, el cuantizador se distorsiona.

Para que los DAC/ADC trabajen normalmente con altos niveles de PAPR, ellos deben tener un alto rango dinámico ocasionando complejidad.

Para evitar la pérdida de información esos grandes rangos de variación deben ser compensados al diseñar el hardware de los DAC/ADC y lógicamente al tener que implementar esas mejoras , ellas serán aprovechadas mayormente en las porciones que exhiben esa mayor concentración de energía (picos) la cual solo se da en pequeñas porciones de la señal , al diseñar el hardware de compensación solo para pequeñas porciones de la señal se cae en ineficiencia en el diseño y costos de dichas implementaciones .

Amplificadores de potencia HPA:

Para transmitir una señal que presente picos altos se requiere de un amplificador de potencia en el transmisor, dichos amplificadores consumen una alta cantidad de potencia y para su correcto funcionamiento se necesita su comportamiento lineal habiendo un compromiso existente entre la linealidad y la eficiencia del HPA.

Para observar esto se introducirán ciertos conceptos acerca del HPA. Para esto observemos inicialmente la grafica característica de un amplificador de potencia de la figura 1.3.

En la curva podemos notar la marca que dice P1dB, Se denomina punto de compresión, normalmente a 1 o 3dB, a aquel punto en el que la ganancia ha bajado dichos dB respecto del valor de ganancia ideal del amplificador. Hasta este punto se considera que el HPA trabaja en zona lineal.

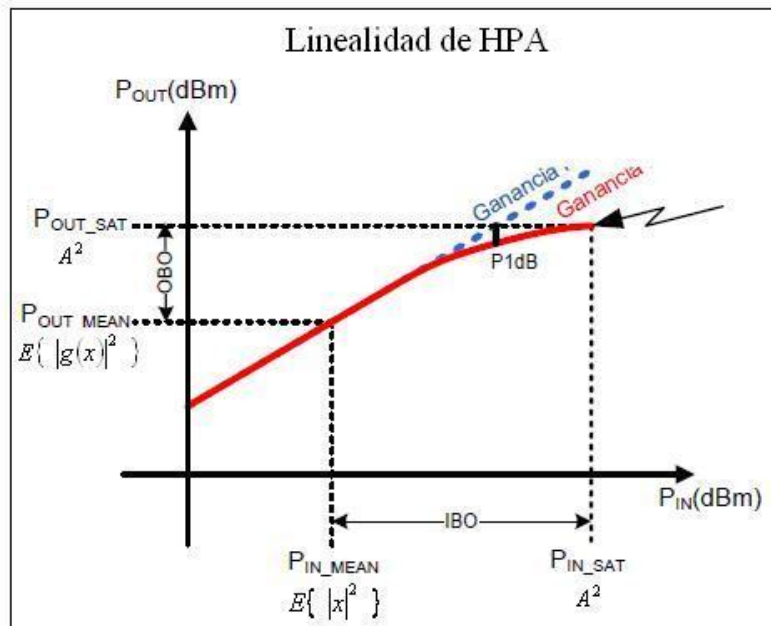


Fig. 1.3.- Linealidad característica de un HPA

Observamos también dos parámetros llamados IBO y OBO :

El Back-Off de Entrada (IBO) se define como la diferencia entre potencia máxima a la entrada del amplificador (Potencia de saturación) A^2 y la potencia media de la señal de entrada

$$E\{|x|^2\}.$$

$$IBO = 10 \log_{10} \left(\frac{A^2}{E\{|x|^2\}} \right) [dB]$$

(1.17)

Este valor también se utiliza en muchos artículos con el nombre de clip level.

El Back-Off de Salida (OBO) se define como la diferencia entre potencia máxima a la salida del amplificador (Potencia de saturación) y la potencia media de la señal de salida.

$$OBO = 10 \log_{10} \left(\frac{A^2}{E\{|g(x)|^2\}} \right) [dB] \quad (1.18)$$

En esta formula se observa el valor $g(x)$ el cual corresponde a la señal de salida del amplificador (ver en mas detalle en el APENDICE B).

Para obtener una amplificación lineal en una situación ideal el IBO se establece igual al PAPR de la señal. De esta manera la señal no resulta interferida por los efectos no lineales introducidos por el amplificador de potencia (HPA) en compresión.

En general, la potencia de salida es mas pequeña que la potencia de entrada en el caso de no – linealidad. Sin embargo, en si la señal multiportadora opera en la región lineal del HPA estas dos potencias son muy similares así que e puede decir que: $IBO \approx OBO$.

En cuanto a la eficiencia del HPA, podemos definirla como una medida que nos dice cuan efectivo un amplificador convierte la potencia generada de una fuente DC a potencia útil en la salida.

Este parámetro de eficiencia se denomina Power Added Efficiency (PAE) y se calcula según la ecuación

$$PAE = \frac{P_o}{P_{dc}} \quad (1.19)$$

P_o es la potencia de salida y P_{dc} es la potencia de la fuente dc.

El consumo de energía en un equipo de comunicaciones por parte del HPA puede llegar a ser del 70% respecto del total.(dependiendo del tipo de amplificador como por ejemplo clase A , B , etc) . Aquí radica la importancia de maximizar la eficiencia de este dispositivo.

En la Figura 1.4 se observa dos graficas, la primera de la curva de trabajo del HPA (donde se ve el punto de trabajo) y la segunda de su eficiencia PAE.

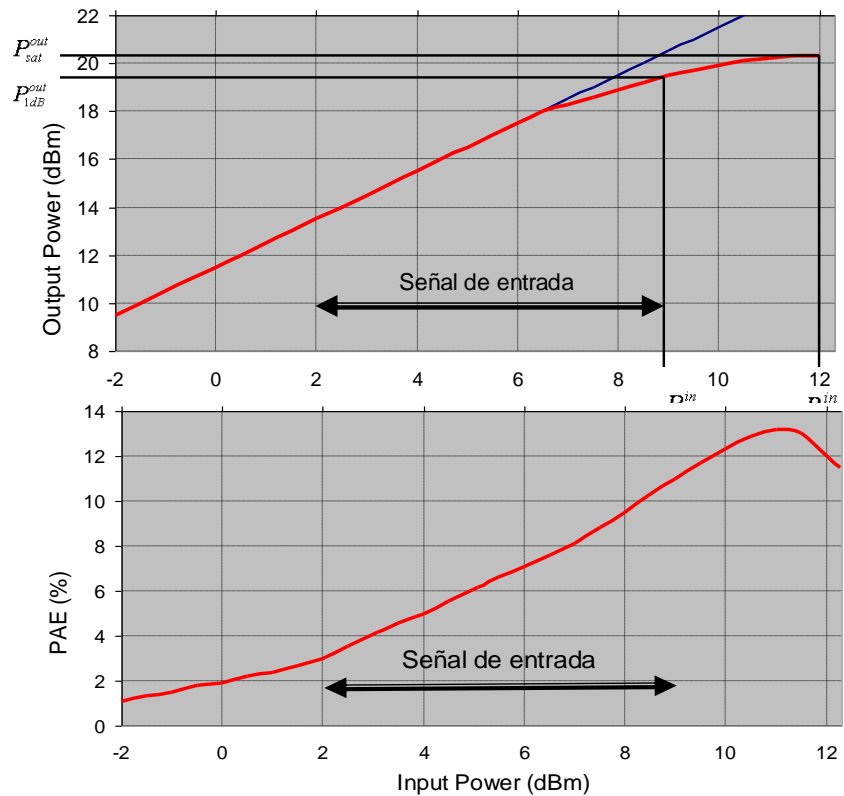


Fig. 1.4.- Gráfica de PAE vs potencia de entrada

Si el punto de trabajo en el HPA es más alto la eficiencia que se puede conseguir es mayor. Por el contrario mayor será la degradación de la señal provocada por los efectos no lineales del amplificador, generando productos de intermodulación y distorsión armónica (mas adelante se observara porque un comportamiento no lineal causa estos efectos). Este es el denominado compromiso entre linealidad y eficiencia.

Al darse la presencia de altos niveles de PAPR se requeriría de amplificadores con un alto rango lineal de operación lo que conduce a un alto costo de implementación.

En OFDM se alcanza la potencia instantánea máxima con una probabilidad muy reducida lo que implica que el sistema sea poco eficiente ya que para garantizar linealidad de los picos - que ocurren ocasionalmente- es necesario trabajar con altos niveles de backoff (IBO) disminuyendo la eficiencia energética. Este punto es muy importante ya que la baja eficiencia energética en la etapa de amplificación debido a utilizar OFDM es realmente un problema.

Si logramos mantener el funcionamiento del HPA en zona lineal, a expensas de costo de implementación para que soporte los altos niveles de PAPR de todas formas se ve que al incrementar dichos valores se obtendrá un descenso de la eficiencia del amplificador, esto lo podemos notar manejando las siguientes expresiones:

Anteriormente se definió el PAE pero a continuación lo podemos expresar en términos relacionado al PAPR si lo

tomamos en un determinado periodo de tiempo T, tal que el PAE para ese lapso se aproxima a:

$$PAE_T = \frac{\overline{Po}}{Pdc} \quad , 0 < t < T \quad (1.20)$$

Y tomamos el caso en que el HPA trabaja en zona lineal (para que no se den distorsiones), como se expresó anteriormente el comportamiento del HPA será tal que $IBO \approx OBO$ y por ende $P_{out} \approx P_{in}$, lo que para el intervalo T escogido significa que $\overline{Po} \approx \overline{Pin}$

Entonces:

$$PAE_T \approx \frac{\overline{Pin}}{Pdc} \approx \frac{E\{|x(t)|^2\}}{Pdc} \quad , 0 < t < T \quad (1.21)$$

cuando el HPA funciona en zona lineal.

Si multiplicamos por la Pin_{max} de ese rango de tiempo T, de tal forma que no altere la expresión podemos obtener:

$$PAE_T \approx \frac{E\{|x(t)|^2\}}{Pdc} \bullet \frac{Pin_{max}}{Pin_{max}} \quad (1.22)$$

$$PAE_T \approx \frac{Pin_{max} / Pdc}{Pin_{max} / E\{|x(t)|^2\}} \approx \frac{Po_{max} / Pdc}{Pin_{max} / E\{|x(t)|^2\}} \quad (1.23)$$

$$PAE_T \approx \frac{PAE_{MAX}}{PAPR} \quad 0 < t < T \quad (1.24)$$

En donde se aprecia que cuando el HPA trabaja en zona lineal, si tenemos cantidades mas grandes de PAPR la eficiencia PAE se vera disminuida (ref [2] pag 18,19) , esto también lo podemos notar en la figura 1.5

que muestra la disminución de la eficiencia al incrementarse el PAPR en el caso de amplificadores clase A y clase B :

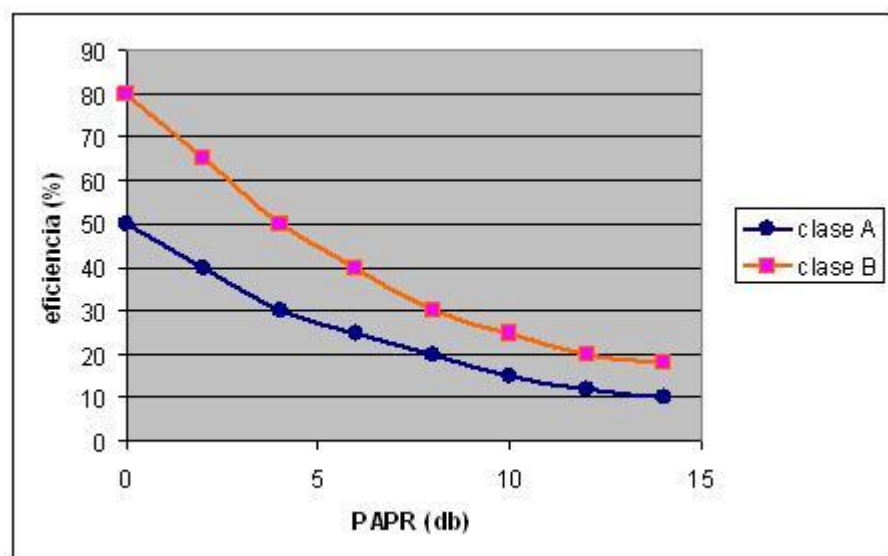


Fig. 1.5.- Relación entre Eficiencia y PAPR en un HPA

Se puede resaltar que como en un sistema de comunicaciones el elemento que consume mayor energía es el Amplificador de Potencia y como vimos anteriormente acerca de los valores críticos del PAPR que se incrementa hasta un valor N de las subportadoras las cuales en la

práctica son valores altos hacen que para el eficiente funcionamiento del HPA se demande altos consumos de potencia DC por lo que se consume grandes cantidades de energía y además se causa deterioro del tiempo de vida de las baterías en aplicaciones móviles.

Por eso en la práctica dichos valores no son tolerables buscándose alternativas para estos problemas en el HPA. Se pensaría en trabajar con factores ligados a disminuir o poder manejar ese incremento de potencia, por decir como ejemplo:

- El número de subportadoras no debe ser grandes para evitar la posibilidad de generar picos altos
- El amplificador de potencia debe tener alta reserva de potencia para garantizar linealidad
- Los convertidores digitales analógicos y analógicos digitales deben tener un amplio rango dinámico

La primera opción toma en consideración la variable número de subportadoras, pensando en el valor del punto crítico del PARP que como se vio anteriormente se incrementa a medida que aumenta N , sin embargo de inmediato se nota la

limitación grave en el sistema dado que si se disminuye el número de subportadoras se disminuye la trama y la tasa de datos, quitando la ventaja de OFDM ya que lo que se requiere es justamente tener un mayor número de subportadoras para transmitir la información y no lo contrario convirtiéndola en una opción inaceptable.

Las 2 opciones siguientes tratan justamente de ajustar el hardware de los convertidores y del HPA para que puedan soportar los rangos/variaciones que maneja la señal, sin embargo como se comentó anteriormente de ese ajuste solo se sacaría provecho en los pocos segmentos de la señal donde se produce esos incrementos grandes, los cuales al ser poco frecuentes hacen que dichos ajustes sean ineficientes además de ser costosos.

Como se ve en la búsqueda de soluciones no cualquier opción es acertada, generalmente el planteamiento de ciertas soluciones inmediatas está ligada a otros factores a su vez., como a continuación puede seguirse observando en otro tipo de soluciones, como por ejemplo:

Si disminuimos la potencia media de la señal también se disminuirá los picos que el amplificador necesita manejar pero existe el problema que al disminuir la potencia media se disminuirá también la SNR causando disminución de la calidad de la señal.

1.4.2 Que sucede si no hacemos nada al respecto?

Si no disminuimos la potencia media de la señal que entra al amplificador y no hacemos nada con los picos altos que pasan a través del amplificador la señal podría llegar a ser clippeada en esos picos altos.

Clipping podría considerarse la solución mas usada ya que generalmente se hace por default, sin embargo no significa que no traiga sus consecuencias.

Cuando los amplificadores limitan la transferencia de esos altos picos de potencia inducidos en la señal son forzados a trabajar en zona no-lineal por lo que ese clippeo introduce no-linealidad en la transmisión del símbolo OFDM causando

distorsión y un incremento en la tasa de error de bit e incremento en el espectro fuera de banda.

Para observar mejor este efecto nótese lo siguiente:

En un sistema lineal la señal de salida es igual a la señal de entrada multiplicada por una ganancia a .

$$v_{OUT}(t) = av_{IN}(t) \quad (1.25)$$

Supongamos que tenemos una señal de entrada (estaría ingresando al HPA) y dicha señal esta compuesta de 2 señales (por decir 2 subportadoras)

$$v_{IN} = v_{IN1}(t) + v_{IN2}(t) \quad (1.26)$$

Cuando el HPA posee un comportamiento lineal esperaríamos una respuesta como la siguiente:

$$v_{out}(t) = a(v_{IN1}(t) + v_{IN2}(t)) \quad (1.27)$$

Pero dado que tenemos el caso de un comportamiento no lineal pueden obtenerse respuestas como:

$$v_{out}(t) \approx \sum_{n=0}^{\infty} (a_n \cdot v_{IN}(t)^n) \quad (1.28)$$

Entonces como estamos trabajando con subportadoras se puede notar que la forma de la señal de entrada y su

respectiva salida podrían darse como se muestra a continuación:

$$v_{IN} = V_1 \cos(\omega_1 t) + V_2 \cos(\omega_2 t) \quad (1.29)$$

$$\omega_1 = \left(2\pi + \frac{\Delta f}{2}\right), \quad \omega_2 = \left(2\pi - \frac{\Delta f}{2}\right)$$

$$\begin{aligned} v_{out}(t) = & \frac{a_2 V_1^2}{2} + \frac{a_2 V_2^2}{2} + \\ & + V_1 \left[a_1 + \frac{3a_3 V_1^2}{4} + \frac{3a_3 V_2^2}{2} \right] \cos(\omega_1 t) + \\ & + V_2 \left[a_1 + \frac{3a_3 V_2^2}{4} + \frac{3a_3 V_1^2}{2} \right] \cos(\omega_2 t) + \\ & + \frac{a_2 V_1^2}{2} \cos(2\omega_1 t) + \frac{a_2 V_2^2}{2} \cos(2\omega_2 t) + \\ & + a_2 V_1 V_2 \left[\cos((\omega_2 - \omega_1)t) + \cos((\omega_2 + \omega_1)t) \right] + \\ & + \frac{a_3 V_1^3}{4} \cos(3\omega_1 t) + \frac{a_3 V_2^3}{4} \cos(3\omega_2 t) + \\ & + \frac{3a_3 V_1^2 V_2}{4} \left[\cos((2\omega_1 + \omega_2)t) + \cos((2\omega_1 - \omega_2)t) \right] + \\ & + \frac{3a_3 V_2^2 V_1}{4} \left[\cos((2\omega_2 + \omega_1)t) + \cos((2\omega_2 - \omega_1)t) \right] + \end{aligned} \quad (1.30)$$

Por simplicidad para el desarrollo sólo ha sido considerada una respuesta de hasta tercer orden.

Se puede observar tanto la creación de distorsión armónica como los productos de intermodulación de 2º y 3er orden introducidos por el HPA debido a su comportamiento no lineal.

La distorsión armónica de segundo orden se encuentra en $2\omega_1$ y $2\omega_2$ y la de tercer orden en $3\omega_1$ y $3\omega_2$. La distorsión producida por los productos de intermodulación de segundo orden se encuentran en $\omega_1 + \omega_2$ y $\omega_1 - \omega_2$ y la de tercer orden en: $2\omega_1 + \omega_2$, $2\omega_1 - \omega_2$ y en $2\omega_2 + \omega_1$, $2\omega_2 - \omega_1$. Los productos de intermodulación de 3er orden son los más perjudiciales debido a que son los más cercanos a los tonos fundamentales (ref [5] pg 4,5,6).

Para visualizar gráficamente la distorsión introducida por un amplificador se inyectan dos tonos tal que

$$\omega_1 = \left(2\pi + \frac{\Delta f}{2} \right), \quad \omega_2 = \left(2\pi - \frac{\Delta f}{2} \right)$$

Donde para una frecuencia central $f_c = 2.4$ Ghz y un espaciado entre los tonos de $\Delta f = 20$ MHz se obtiene el resultado de la Figura 1.6.

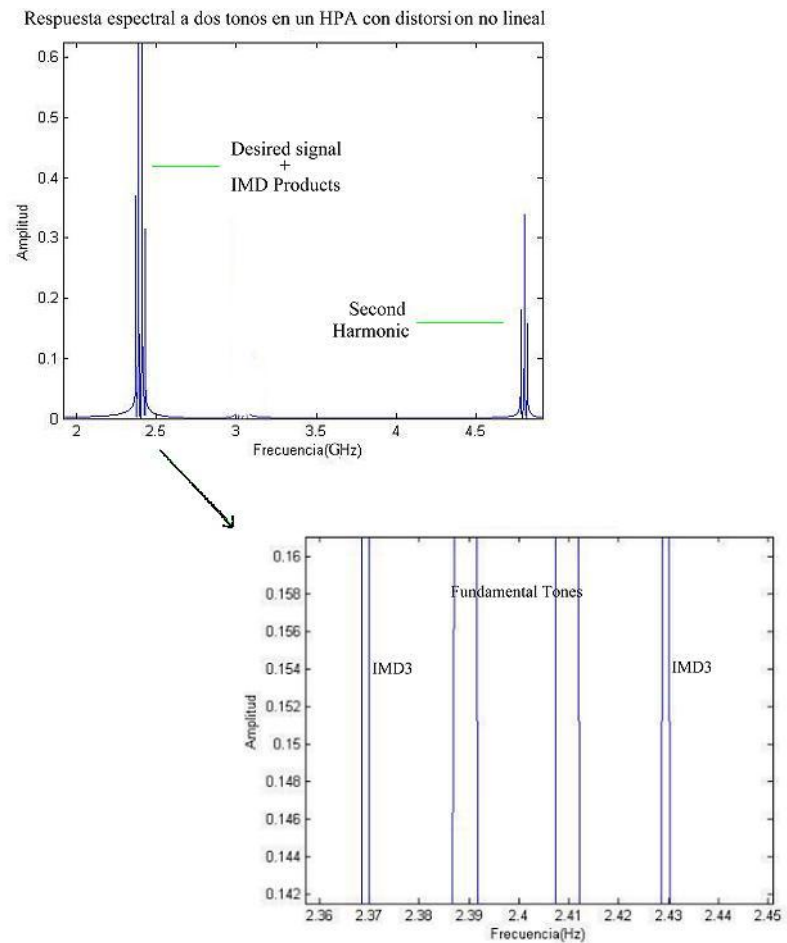


Fig. 1.6.- Respuesta espectral a la inyección de 2 tonos en un amplificador con distorsión no lineal

En ella se puede apreciar la distorsión armónica y los productos de intermodulación (IMD) debidos al comportamiento no lineal del amplificador.

Estos resultados muestran que el clipping de la señal a pesar de ser una alternativa bastante usada, no es la mejor opción,

ubicándola de preferencia como la alternativa al final de haber utilizado otro tipo de técnica.

Es en esta búsqueda de procedimientos mas óptimos para la solución del problema de los altos niveles de PAPR es donde han hecho su aparición una gran cantidad de distintos y variados métodos con el paso del tiempo gracias a los estudios de distintos autores.

Cabe notar que en sistemas de telecomunicaciones hay tres factores que se deben tomar en cuenta para obtener alguna mejora en el sistema, que son: potencia, ancho de banda y tasa de transmisión. Por ejemplo si queremos mejorar o aumentar la tasa de transmisión debemos de aumentar el ancho de banda lo cual no es algo realmente factible. En nuestro caso específico, debemos de buscar alguna técnica para reducir el PAPR con sacrificio de ancho de banda o bien de tasa de transmisión de datos.

Para elegir una técnica adecuada para reducir el PAPR se deben tomar en cuenta los siguientes factores (ref [7] pag 43):

- Capacidad para reducir el PAPR
- Incremento de potencia en la señal transmitida
- Pérdida en tasa de datos
- Complejidad de cálculo

En la siguiente sección presentamos una técnica para reducir el PAPR, iniciamos con la definición y al final mencionamos sus características.

CAPÍTULO 2

2.-DEFINICIÓN DEL ESCENARIO DEL PROYECTO

2.1 Standard y limitaciones

En este informe se ha realizado un análisis basado en diferentes fuentes, los cuales nos han permitido facilitar un enfoque específico. Aplicaremos esta técnica a los sistemas de comunicación inalámbricos porque es donde los avances tecnológicos de comunicación tienden a desarrollarse aceleradamente y necesitan optimizar en gran medida el consumo de energía, por ejemplo los equipos móviles ahorrarían consumo de baterías por no tratar de incrementar su potencia ante una señal distorsionada.

El sistema WI-FI 802.11a será nuestro estándar de análisis, cuyos parámetros a considerar en la simulación son:

- Numero de frecuencias que llevan información 52.
- Número de frecuencias pilotos para estimación del canal 4.
- Índice de frecuencia 0 se usará con cero también.
- Se agregan 16 símbolos que se usarán como Prefijo Cíclico.
- Emplearemos la modulación BPSK

Nuestro escenario de estudio se limita principalmente en el transmisor, que es donde hay el problema.

Para nuestro objetivo de reducir el PAPR nos vamos a enfocar en una técnica llamada Reservación de tonos, lo cual detallamos a continuación.

2.2 Tonos Reservados

En este esquema, algunas subportadoras OFDM son reservadas. Esas subportadoras no llevan información de datos, son solamente usadas para reducir $PAPR$. Las subportadoras elegidas formaran una señal c que será adicionada a la señal de información x con el

objetivo de reducir sólo los picos de x . Este método es llamado “Reservación de Tonos”.

Se utilizara como notación letras minúsculas para señales en el dominio del tiempo y sus respectivas mayúsculas en el dominio de la frecuencia.

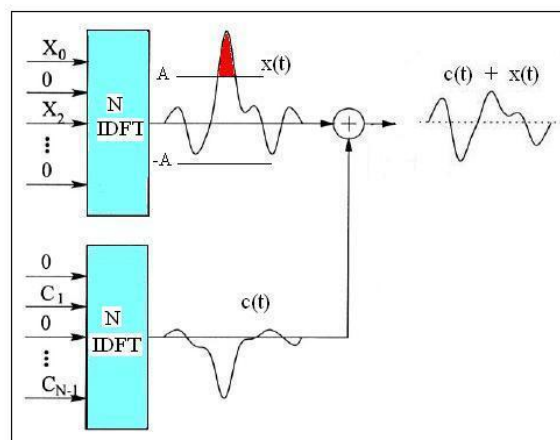


Fig. 2.1.- Generación de una señal $c(t)$ para reducir picos altos de $x(t)$

Este método restringe el vector de datos X , y el vector de reducción de pico C a estar en subespacios de frecuencias disjuntas (ref [4] pag 48), o sea:

$$X_k = 0 \quad k \in \{i_1, \dots, i_R\} \quad (2.1)$$

$$C_k = 0 \quad k \notin \{i_1, \dots, i_R\} \quad (2.2)$$

Esta formulación es sin distorsión y lleva a una decodificación muy simple de los símbolos de datos que son extraídos desde la secuencia recibida por elección del set de valores $k \notin \{i_1, \dots, i_R\}$ en la salida de FFT del receptor. Además, esto permite una técnica de optimización simple para el cálculo del vector de reducción de picos c . Los R valores no-cero en la matriz C serán llamados tonos de reducción de picos.

Asumamos que los R tonos $\{i_1, \dots, i_R\}$ han sido fijos en el comienzo de la transmisión y que ellos no se cambiarán hasta que la transmisión es realizada o alguna nueva información sobre el canal es realimentado al transmisor.

Llamemos C' los valores no-cero de C , o sea:

$$C' = [C_{i_1} \dots C_{i_R}]^T \quad (2.3)$$

Y $Q = [q_{i_1} | \dots | q_{i_R}]$ la submatriz de Q construida por elección de sus columnas $\{i_1, \dots, i_R\}$, entonces [7]:

$$c = QC = Q'C' \quad (2.4)$$

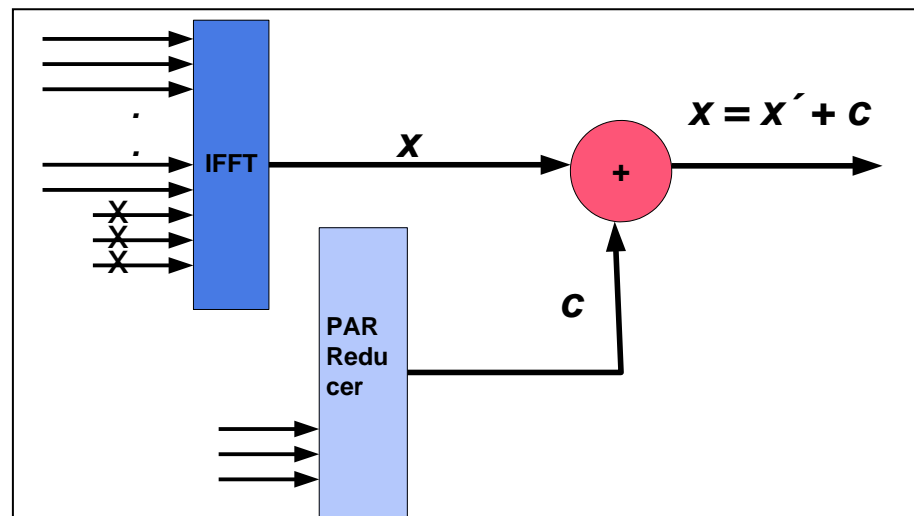


Fig. 2.2.- Reservación de tonos para formar c

La señal reducida con la señal reductora de picos queda expresado como:

$$x' = x + c \quad (2.5)$$

Donde:

x' es la matriz $N \times 1$ que representa la señal reducida

x es la matriz $N \times 1$ que representa la señal original

c es la matriz $N \times 1$ que representa la señal reductora

$$c = QC = Q'C'$$

Donde:

Q es la matriz IFFT, de orden $N \times N$

C es la matriz reductora de PAR, de orden $N \times 1$

Q' es la matriz IFFT reducida con solamente las columnas que corresponden a las frecuencias reservadas, de orden $N * R$

C' es la matriz C reducida con los tonos no reservados removidos, de orden $R * 1$

$$x' = x + Q'C' \quad (2.6)$$

$$PAPR\{x + c\} = \frac{\max \|x + c\|^2}{E|x[n]|^2} \quad (2.7)$$

Una vez que el numerador no depende de c , el problema de minimizar el PAR de las señales combinadas es equivalente a c^* o C^* que minimice los máximos valores picos de $(x + c)$

$$\min_c \|x + c\|_\infty^2 = \min_{C'} \|x + Q'C'\|_\infty^2 \quad (2.8)$$

Existen software's que resuelven el problema convexo de arriba. Pero un algoritmo de búsqueda sub-optima de gradiente iterativo también da buenos resultados lo cual presentamos en el capitulo siguiente
Podemos resumir las características de esta técnica en la siguiente tabla:

Características de tonos reservados	
Perdida de tasa de datos	SI
Distorsión	NO
Aumenta potencia media de la señal	SI
Complejidad en transmisor	ALTA
Complejidad en Receptor	BAJA

Tabla I.- Características de Tonos Reservados

Como vemos en la tabla la técnica de tono reservado en una técnica bastante buena para ser análisis de estudio

CAPÍTULO 3

3.- ALGORITMOS PARA REDUCIR EL PAPR UTILIZANDO TONOS RESERVADOS

Para este proyecto se realizó el análisis utilizando un algoritmo iterativo para reducir el *PAPR*. También presentamos otro método llamado “Pulso Gaussiano”.

3.1 Algoritmo de gradiente simple con rápida convergencia en tonos reservados

En esta parte se describe un algoritmo iterativo simple para calcular la señal reductora del PAR de los tonos reservados basado en el gradiente de “error cuadrático medio” de la función de distorsión del transmisor (ref [4] pag 71).

Hay una nomenclatura que nos será útil en el transcurso de esta sección. Es la fórmula general de la norma euclidiana, la cual se expresa como:

$$\|x\|_2 = \sqrt{\sum_j |x_j|^2} \quad (3.1)$$

Comenzaremos representemos $g(x)$ como la función no lineal de HPA, que en varias literaturas se maneja también como la función de clippeo ya que muestra el comportamiento que tendrá la señal al pasar de un valor A en el cual deja su comportamiento lineal y comienza a recortarse (como al saturarse el HPA, mas acerca del modelado no lineal del HPA en el APENDICE B)

$$\text{Donde: } g(x) = \begin{cases} |x|, & |x| \leq A \\ Ae^{j\arg(x)}, & |x| > A \end{cases} \quad (3.3)$$

$\arg(x)$ es la fase de la señal x

A es el nivel clipping

La señal de entrada al pasar por un HPA y llegar a funcionamiento en zona no lineal comienza a recortarse tal que a su entrada x se obtiene una salida g que es menor (recortada) esa energía extra recortada causa distorsión $x-g$ que en términos de energía es $(x-g)^2$

Aplicando esto a nuestro caso para cada una de las muestras x_n y sacando la totalidad de distorsión en una señal multiportadora tenemos:

$$\sum_{n=0}^{N-1} (x_n - g(x_n))^2 = \left(\sqrt{\sum_{n=0}^{N-1} (x_n - g(x_n))^2} \right)^2 = \|x - g(x)\|_2^2 \quad (3.4)$$

El radio señal a distorsión SDR de una señal nos puede indicar que tan pura nos sale la señal y la podemos obtener por:

$$SDR = P_{señal}/P_{distorsion} \quad (3.5)$$

Y para nuestra señal en sus N muestras podemos encontrar la potencia total como:

$$\sum_{n=0}^{N-1} (x_n)^2 = \left(\sqrt{\sum_{n=0}^{N-1} (x_n)^2} \right)^2 = \|x\|_2^2 \quad (3.6)$$

Reemplazando la potencia de la señal con la de la distorsión generada se tiene:

$$SDR = \frac{\|x\|_2^2}{\|x - g(x)\|_2^2} \quad (3.7)$$

Adicionando la señal de reducción de $PARP$ para formar una nueva señal x tal que sea $x + c = x + Q'C'$ (como se muestra anteriormente para nuestro caso de tone reservación) a la señal transmitida, la SDR es modificada como:

$$SDR = \frac{\|x\|_2^2}{\|x + Q'C' - g(x + Q'C')\|_2^2} \quad (3.8)$$

Donde el numerador se mantiene debido a que es de mayor interés encontrar la relación con respecto a la señal pura y que la inclusión de la señal c no debe crear un incremento significativo en la potencia promedio de la señal.

En la formula del *PARP* el objetivo fue disminuir el numerador $x + Q'C'$ para minimizar el *PARP*.

Para una no linealidad general, un enfoque mas natural es maximizar el *SDR*, lo cual es una medida de calidad mas aceptada de la señal x .

Puesto que el numerador es una constante, la optimización de este problema es equivalente a minimizar el denominador para obtener una mejoría en *SDR*.

$$\min_c \|x + Q'C' - g(x + Q'C')\|_2^2 \quad (3.9)$$

Para estos sacamos el gradiente del denominador, matemáticamente nos queda

$$\nabla C' \|x + Q'C' - g(x + Q'C')\|_2^2 \quad (3.10)$$

Para resolver este gradiente vamos a descomponer la función objetivo o sea resolviendo primeramente la norma euclidiana para después derivar con respecto a C' .

Por lo tanto la norma euclidiana de $\|x + Q'C' - g(x + Q'C')\|_2$ es:

$$\sqrt{\sum_{|x_n+c_n|>A} |(x_n + c_n - g(x_n + c_n))|^2} \quad (3.11)$$

Reemplazando en $\nabla C' \|x + Q'C' - g(x + Q'C')\|_2^2$ nos queda:

$$\nabla C' \left(\sqrt{\sum_{|x_n+c_n|>A} |(x_n + c_n - g(x_n + c_n))|^2} \right)^2 \quad (3.12)$$

Simplificando la potencia con la raíz nos queda:

$$\nabla C' \sum_{|x_n+c_n|>A} |(x_n + c_n - g(x_n + c_n))|^2, \quad \text{para resolver el gradiente}$$

derivamos con respecto a C' , y como las componentes de la matriz

c se obtiene mediante $q_n^{row} C'$, o sea, $c_n = q_n^{row} C'$, donde q_n^{row} es la

enésima fila de la matriz IFFT reducida Q' , por lo tanto obtenemos

la siguiente expresión:

$$\sum_{|x_n+c_n|>A} \frac{\partial}{\partial C'} |(x_n + q_n^{row} C' - g(x_n + q_n^{row} C'))|^2 \quad (3.13)$$

$$\sum_{|x_n+c_n|>A} 2(x_n + q_n^{row} C' - g(x_n + q_n^{row} C')) \frac{\partial}{\partial C'} ((x_n + q_n^{row} C' - g(x_n + q_n^{row} C'))) \quad (3.14)$$

$$\sum_{|x_n+c_n|>A} 2(x_n + q_n^{row}C' - g(x_n + q_n^{row}C')) \left(\frac{\partial}{\partial C'} x_n + \frac{\partial}{\partial C'} q_n^{row}C' - \frac{\partial}{\partial C'} g(x_n + q_n^{row}C') \right) \quad (3.15)$$

reemplazando $g(x_n + q_n^{row}C') = Ae^{j\arg\{x_n+c_n\}}$, nos queda:

$$\sum_{|x_n+c_n|>A} 2(x_n + q_n^{row}C' - Ae^{j\arg\{x_n+c_n\}}) \left(\frac{\partial}{\partial C'} x_n + \frac{\partial}{\partial C'} q_n^{row}C' - \frac{\partial}{\partial C'} Ae^{j\arg\{x_n+c_n\}} \right) \quad (3.16)$$

$$2 \sum_{|x_n+c_n|>A} (x_n + q_n^{row}C' - Ae^{j\arg\{x_n+c_n\}})(0 + q_n^{row} - 0) \quad (3.17)$$

y como $q_n^{row}C' = c_n$, finalmente tenemos:

$$2 \sum_{|x_n+c_n|>A} (x_n + c_n - Ae^{j\arg\{x_n+c_n\}}) q_n^{row} \quad (3.18)$$

Donde:

q_n^{row} , es la n -ésima fila de la matriz IFFT reducida Q'

Note que el término distorsión es diferente de cero solamente para esos valores de n donde la señal es clipeada (recortada)

Por el algoritmo de gradiente iterativo, tenemos la siguiente formula aplicado a C' :

$$C'^{(i+1)} = C^i - \frac{\mu}{2} \nabla C' \|x + Q'C' - g(x + Q'C')\|_2^2 \quad (3.19)$$

Reemplazando el gradiente, tenemos:

$$C^{(i+1)} = C^{(i)} - \mu \sum_{|x_n + c_n^{(i)}| > A} (x_n + c_n^{(i)} - Ae^{j \arg\{x_n + c_n^{(i)}\}}) q_n^{row} \quad (3.20)$$

Para facilidad podemos operar en el dominio del tiempo, o sea multiplicamos por la matriz Q' :

$$Q' C^{(i+1)} = Q' C^{(i)} - \mu \sum_{|x_n + c_n^{(i)}| > A} (x_n + c_n^{(i)} - Ae^{j \arg\{x_n + c_n^{(i)}\}}) Q' q_n^{row} \quad (3.21)$$

$$c^{(i+1)} = c^{(i)} - \mu \sum_{|x_n + c_n^{(i)}| > A} (x_n + c_n^{(i)} - Ae^{j \arg\{x_n + c_n^{(i)}\}}) Q' q_n^{row} \quad (3.22)$$

Para este método las DTF's no son necesarias para calcular $c^{(i)}$, con la actualización del gradiente de la fórmula de arriba, el algoritmo para reducir el *PAPR* es

$$x^{(i+1)} = x^{(i)} - \mu \sum_{|x_n^{(i)}| > A} (x_n^{(i)} - Ae^{j \arg\{x_n^{(i)}\}}) Q' q_n^{row} \quad (3.23)$$

Donde:

$$x^{(i)} = x + c^{(i)} \quad (3.24)$$

Podemos simplificar la ecuación de arriba, asumiendo que:

$\alpha_n^{(i)} = x_n^{(i)} - Ae^{j \arg\{x_n^{(i)}\}}$ es un escalar complejo, y $\hat{P}_n = Q' q_n^{row}$ es un vector constante que solo depende del set de tonos reservados R y así puede ser precalculado y almacenado en memoria durante la inicialización del MODEM.

El algoritmo iterativo completo de tono reservado basado en el gradiente de SDR es simplificado:

$$x^{(i+1)} = x^{(i)} - \mu \sum_{|x_n^{(i)}| > A} \alpha_n^{(i)} \hat{p}_n \quad (3.25)$$

Donde, $x^{(0)} = x$. Podemos aplicar esta ecuación al caso de multiportadoras real con las modificaciones sencillas siguientes (ref [4] pag 79,80) :

$$\alpha_n^{(i)} = x_n^{(i)} - A \cdot \text{sign}\{x_n^{(i)}\} \quad (3.26)$$

$$\hat{p}_n = Q q_n^{row} \quad (3.27)$$

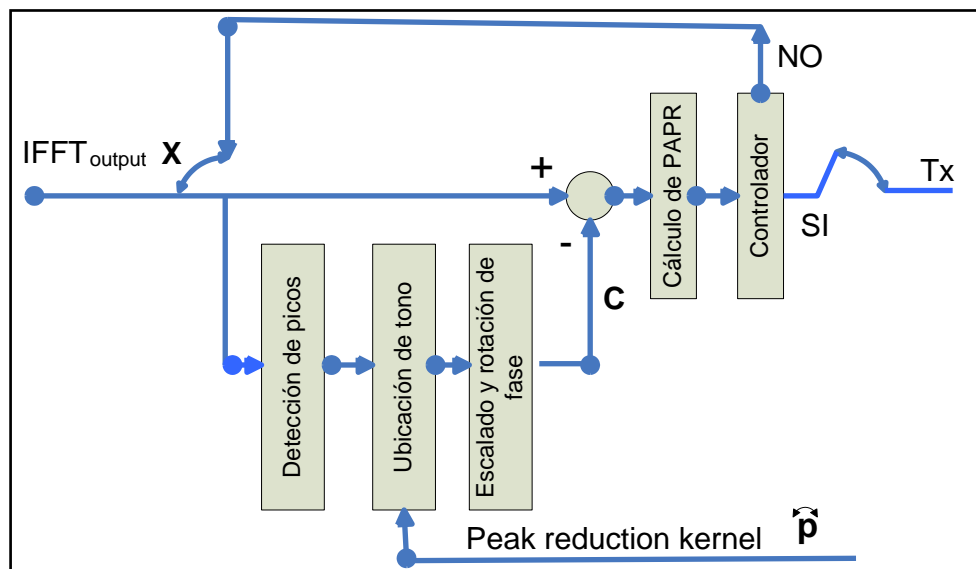


Fig.3.1.- Esquema del algoritmo gradiente

3.1.1 Descripción de iteraciones con algoritmo gradiente:

Estos pasos solo se requieren hacer una sola vez al inicio, lo cual consiste en:

1. Seleccionar el valor objetivo de PARP, o sea el nivel de A . Esto determinará las propiedades del set de tonos reservados R .
2. Escoger el set de tonos reservados R
3. Calcular y almacenar \hat{p} asociado con set de tonos reservados R .

Ejecución:

1. Condición inicial : $x^{(0)} = x$
2. Encontrar el pico máximo tal que $|x_n^{(i)}| > A$. Si todas las muestras están abajo del objetivo A , saltar al paso 5
3. Actualizar $x^{(i)}$ de acuerdo a la formula

$$x^{(i+1)} = x^{(i)} - \mu \sum_{|x_n^{(i)}| > A} \alpha_n^{(i)} \hat{p}_n$$

4. Incrementar el contador, $i = i + 1$. Si $i < \text{MaxIteracion}$ saltar a 2
5. Transmitir $x^{(i)}$.

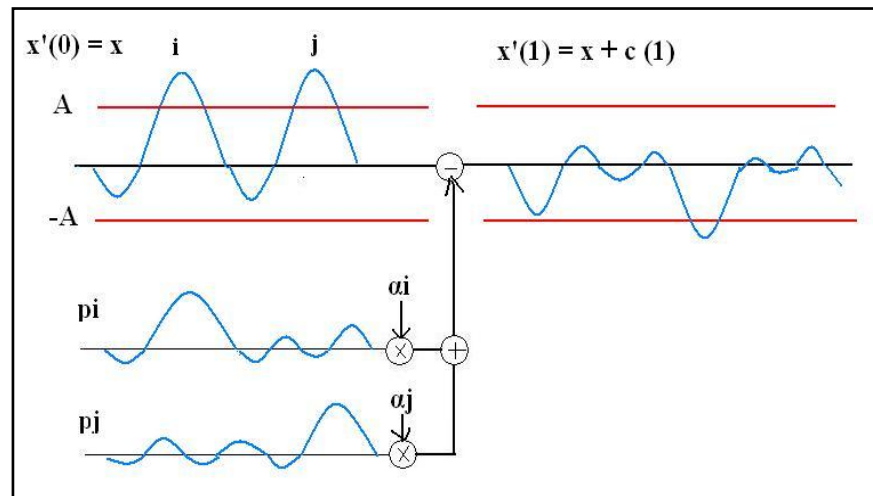


Fig. 3.2.- Representación del algoritmo gradiente de *SDR* para un primer paso

3.2 Reducción de PAPR mediante tonos reservados usando Pulso Gaussiano

En el enfoque de reservación de tonos, un pequeño número de subportadoras (tonos) son reservados para crear una señal C como se muestra en e ilustra en el gráfico de la figura 2.1, Dicha señal C se crea con el objetivo de cancelar los picos altos de señal portadora de la información en el transmisor. Esta señal de cancelación debe ser generada en el dominio de la frecuencia usando el mínimo número de tonos para maximizar el rendimiento de datos, sin embargo es también preferible tener señal estrecha en el dominio

del tiempo para evitar la generación de picos secundarios. En otras palabras, el enfoque de reducción del PAPR en tonos reservados es un problema diseñar una señal limitada: una señal debe de ser diseñada en el dominio de la frecuencia, pero sus efectos es evaluado en el dominio del tiempo. En el corriente enfoque de tonos reservados, la señal de cancelación es principalmente generada desde procesos de pruebas y error o implica computacionalmente complejos procedimientos de optimización.

Aquí se analiza un algoritmo simple, en el cual una señal Gaussian Windows es empleada en el dominio de la frecuencia para formar le pulso de cancelación en el dominio del tiempo.

Para apreciar mejor el funcionamiento de dicho método comencemos por observar la forma de un pulso Gaussiano, el cual se muestra a continuación:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}} \quad (3.28)$$

Donde: a , b y c son constantes reales ($a > 0$).

La gráfica de la función es simétrica con forma de campana, conocida como **campana de Gauss**. El parámetro a es la altura de

la campana centrada en el punto b , determinando c el ancho de la misma.

Por ejemplo, este es una gráfica cuando se dan los valores $a=1$, $b=5$,
 $c=1$

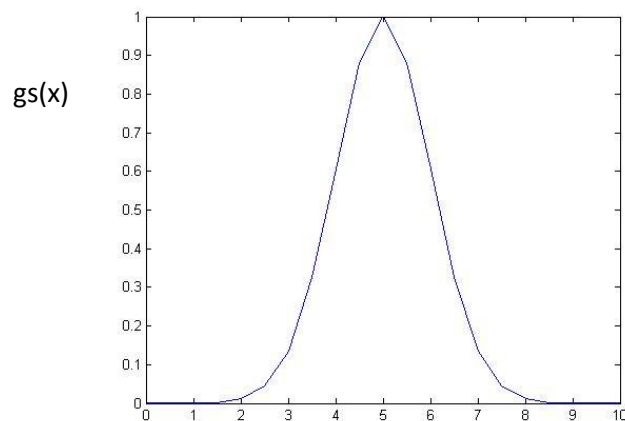


Fig. 3.3.- Pulso Gaussiano

El valor de c^2 corresponde al valor de la varianza o sea el cuadrado de la desviación estándar de la señal $c^2 = \sigma^2$

Para nuestra aplicación como señal reductora nosotros deseamos crear este pulso en un set de tonos reservados L , para centrar el pulso en el centro del set de tonos usaremos el parámetro b que es el que nos coloca el centro de la campana, tal que $b=L/2$, mientras que el ancho de la ventana no debe pasar de ese set de tonos reservados , por lo que la desviación estándar máxima de la señal debería ser de $L/2$ (para que la campana no exceda hacia los lados

del set de tonos reservados que se tiene $\sigma_{\max} = L/2$, este valor lo podemos manejar disminuyéndolo según el ancho de la ventana que deseemos, dividiendo para un valor α mayor a cero, tal que la desviación estándar deseada sea:

$$\sigma = (L/2) * (1/\alpha) \quad (3.29)$$

Tómese en cuenta que el parámetro que maneja la varianza en el pulso gaussiano es c , tal que $c^2 = \sigma^2$ por lo que en nuestro caso necesitaremos $c^2 = \sigma^2 = ((L/2) * (1/\alpha))^2$

Si para la magnitud del pulso solo manejamos el valor de 1, finalmente los valores que necesitamos usar en la expresión del pulso Gaussiano para aplicarlo a nuestro set de tonos reservados serán:

$$a = 1 \quad b = L/2 \quad c^2 = \sigma^2 = ((L/2) * (1/\alpha))^2$$

reemplazando en $f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$, tenemos:

$$f(x) = e^{-\frac{1}{2} \left[\alpha \frac{x-L/2}{L/2} \right]^2} \quad (3.30)$$

Donde al pasarla a forma discreta para nuestras L muestras (de los L tonos reservados) podemos expresar que para una muestra m tal que $0 < m < L - 1$

$$f(m) = e^{-\frac{1}{2} \left[\alpha \frac{m - \frac{L}{2}}{L/2} \right]^2} \quad (3.31)$$

Y ya tenemos la expresión para el pulso que ubicaremos en el rango de la frecuencia [8]. Ahora, este pulso tiene sus correspondientes efectos en el dominio del tiempo, los cuales para visualizarlos debemos hacer su transferencia al dominio del tiempo a través de la transformada inversa de Fourier, pero para esto notaremos una propiedad importante que se tiene al realizar la transformada de Fourier de un pulso Gaussiano.

En el APENDICE D se muestra un análisis de la obtención de la transformada de Fourier de un pulso Gaussiano llegando a esta expresión:

$$f(x) = \exp(-Mx^2) \xrightarrow{F} F(k) = \sqrt{\frac{\pi}{M}} e^{-\frac{k^2}{4M}} \quad (3.32) \quad (3.33)$$

Donde se puede apreciar como la transformada de Fourier de un pulso Gaussiano resulta en otra forma similar a la primera (otro pulso Gaussiano), esto lleva a la consideración lógica de que el caso contrario es similar, es decir la transformada inversa de Fourier de un pulso Gaussiano es también otro pulso Gaussiano. Para apreciar mejor este efecto vamos a graficar con la herramienta MATLAB la transformada inversa de Fourier de la expresión (3.33) que obtuvimos, para lo cual implementamos un pequeño código que se coloca a continuación en el cual se aprovecha la función IFFT existente de MATLAB para evitar complicadas integrales y obtener en el caso discreto rápidamente la transformada inversa.

El código es el siguiente:

```
alpha=4 ; %
L=4 ; % para 4 tonos reservados
m=(0:0.05:L);
Xf = exp (-0.5*( (alpha * ( m-(L/2) ) / (L/2) ).^2 )) ; % el pulso
gaussiano en frecuencia
plot(m,Xf)
title('Pulso en el dominio de la frecuencia')
xlabel('muestra')
ylabel('magnitud')
```

```

xt = ifft(Xf) ; % la transformada inversa de Fourier para pasar al
dominio del tiempo

xt=fftshift(xt); % para graficar con el centro en el centro del set de
tonos

figure(2)

plot(m,abs(ct))

title('Pulso en el dominio del tiempo')

xlabel('muestra')

ylabel('amplitud')

```

Que nos ayudara a mostrar el caso cuando se trata de 4 tonos reservados ($L=4$) con lo cual se obtuvo las siguientes graficas:

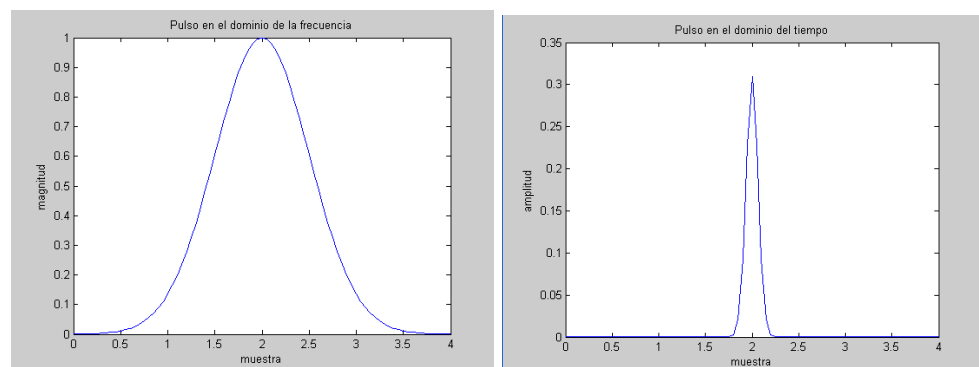


Fig. 3.4.- Pulso Gaussiano en el dominio de la frecuencia y en el dominio del tiempo

Donde se muestra la señal que resultaría en el tiempo, y que serviría para realizar la corrección c de nuestro sistema (por el método de

tonos reservados pues dicha señal se implemente en el set de tonos reservados) tal que $x' = x + c$

Para realizar la corrección de nuestro sistema el valor de la señal c debe ser tal que elimine el exceso de la señal sobre el umbral que nos proponemos (dicho umbral puede ser por ejemplo el valor de saturación del amplificador o un valor del cual no deseamos que exceda la señal), si tal valor límite por ejemplo es A_s , para eliminar el pico excedente necesitaremos de un valor $x_{\text{pico}} - A_s$ el cual restar a la señal original en ese pico, y dado que la señal que va a eliminar el excedente es con el pulso Gaussiano podemos fijar la amplitud del pulso en dicho valor.

Por ejemplo sea el pulso en frecuencia que generamos $G_s(f)$ con el cual la transformada inversa de Fourier nos da una señal en tiempo $F^{-1}(G_s(f)) = g_s(t)$, ese pulso $g_s(t)$ tendrá un valor máximo (pico del pulso) $g_{s\text{max}}$. Al ser ese $g_s(t)$ el pulso que se usará para formar nuestra señal de corrección c la señal tendrá que tener la amplitud tal para realizar la corrección de ese exceso de pico, tal que:

$$c = -((x_{\text{pico}} - A_s) / g_{s\text{max}}) * g_s(t)$$

Este sería el caso para x y $g_s(t)$ con valores reales, pero considerando los valores de x y $g_s(t)$ complejos (como pasa en OFDM/DMT), sabemos que se posee un valor de magnitud y una fase respectiva, y que para eliminar un pico complejo no solo se necesitaría saber la magnitud sino colocar el pico corrector en fase contraria al del pico para que pueda restarse (se puede hacer poniéndolo con la misma fase del pico pero sumado π para que quede en sentido contrario y se pueda restar), es así que si nuestro valor de pico x es $x_{pico} = |x_{pico}| e^{j \angle(x_{pico})}$ la señal correctora debe ser:

$$c = \underbrace{(|x_{pico}| - A_s) / |g_{smax}|}_{\text{ajuste en magnitud}} * \underbrace{e^{j(\angle(x_{pico}) + \pi)} / e^{j \angle(g_s(t))}}_{\text{ajuste en fase}} * |g_s(t)| e^{j \angle(g_s(t))}$$

o sea que nuestra señal c , basada en el pulso gaussiano, para realizar la corrección se debe ajustar a:

$$c = (|x_{pico}| - A_s) / |g_{smax}| * e^{j(\angle(x_{pico}) - \angle(g_s(t)) + \pi)} * g_s(t) \quad (3.34)$$

Algo más a considerar es que el pico corrector debe coincidir en la posición con el pico excesivo, para esto debemos hacerlos coincidir.

Para esto último podemos resaltar una propiedad de las transformadas de Fourier que es la propiedad de traslación:

$$\mathfrak{F}\{f(t - a)\}(w) = e^{-iwa} \cdot \mathfrak{F}\{f\}(w) \quad (3.35)$$

Viéndolo de forma inversa y aplicándoselo a nuestro pulso en frecuencia para trasladarlo a un tiempo t_p donde esta el pico se puede expresar

$$(e^{-i*np*2\pi f}) * G_s(f) \xrightarrow{F^{-1}} g_s(t-t_p) \quad (3.36)$$

Pero para poner lo equivalente en tiempo discreto (no continuo) necesitamos tomar en cuenta el número de muestras $N_{muestras}$ tal que $n = 1, 2, 3 \dots N_{muestras}$. si nuestro pico aparece en una muestra $n = n_p$ podemos trasladar el pico de la función correctora hasta dicha muestra ya que:

$$(e^{-i*(n_p/N_{muestras})*2\pi n}) * G_s(n) \xrightarrow{F^{-1}} g_s(n-(n_p/N_{muestras})) \quad (3.37)$$

Con lo cual podemos colocar la señal en la muestra n_p que tiene el pico, nótese que en nuestro caso también se podría usar una variable n que corresponde a numero de muestras en ambos

extremos dado que se esta usando un numero de muestras N en la función en dominio del tiempo y N subportadoras para la frecuencia.

Con todo esto se puede apreciar que el pulso Gaussiano puede ser optimizado en el dominio del tiempo y en el dominio de la frecuencia. Al seleccionar un pequeño número de tonos en el dominio de la frecuencia se pudo observar que podemos generar un pulso estrecho en el dominio del tiempo

Al generar una corrección en un pico siempre puede darse posibilidad de generación de nuevos picos o de la ya existencia de otros por lo que se acostumbra llevar la corrección a un método iterativo de corrección y comprobación nuevamente. Es así que se crea el algoritmo de corrección en el cual se ira variando un coeficiente que en este caso tiene que ver con el pulso Gaussiano.

El coeficiente de la ventana Gaussiana, G_s , puede ser calculado de la siguiente fórmula:

$$G_{s[m+1]} = e^{-\frac{1}{2} \left[a \frac{m-L/2}{L/2} \right]^2} \quad 0 < m < L-1 \quad (3.38)$$

Donde α representa el recíproco de la desviación estándar (como se vio anteriormente) , y el ancho de la ventana es inversamente relacionado a α . El valor de L representa el número de tonos reservados para generar la señal de cancelación.

A diferencia de otros métodos, la generación de esta señal de cancelación no requiere complicados procedimientos de optimización

Esta señal pico simple es fácilmente optimizada tanto en el dominio del tiempo como en el dominio de la frecuencia. Solo una vez la operación IFFT es requerida (para cada pico).

3.2.1 Pasos iterativos con pulso Gaussiano

Una vez que una eficiente señal de cancelación es obtenida, un algoritmo de conversión rápida puede ser aplicada al sistema OFDM/DMT para cancelar los altos picos de modo que la señal transmitida no excede el umbral requerido A . El algoritmo puede ser implementado como sigue:

- 1.- Una señal de cancelación predefinida es generada usando el pulso Gaussiano $G_s(m)$ (ecuación 3.31) y colocada dentro del

set de tonos reservados solamente tal que la señal de cancelación solamente tiene valores no cero en la locación de tonos reservados (en el resto de los tonos será cero), esto en el dominio de la frecuencia y pasara a ser un pico agudo en el dominio del tiempo

2.- Primero chequeamos si hay picos que exceden el umbral requerido A en la señal portadora de información. Si lo hay, las magnitudes del picos y sus correspondientes locaciones (muestra donde esta) son detectados.

3.- Para cada pico detectado en el dato de información, el pico de la señal de cancelación predefinida es trasladada a la locación del pico gracias a la propiedad de traslación de la transformada de Fourier (formula 3.37) y escalada al valor tal que la potencia del tono pico puede ser reducida al nivel objetivo deseado (usando la formula 3.34) . El proceso se hace para todas las muestras con pico tal que todas esas señales de cancelación apropiadamente escalada y trasladada en fase son sustraídas de la señal de información original.

4.-Después de la cancelación de picos, la señal compuesta será detectado de nuevo, ya que algunos picos secundarios pueden aparecer durante la operación de cancelación de picos anterior. El proceso es continuado hasta que todos los picos están abajo del umbral requerido o hasta que el número máximo de iteraciones son excedidos.

Una vez que la señal cancelación de picos basado en pulso Gaussiano tiene una forma aguda en el dominio del tiempo la ocurrencia de picos secundarios es minimizada. Típicamente solamente una iteración es requerida para remover cada pico detectado en la señal original en el dominio del tiempo. En este enfoque, cuando la señal el pulso de la señal de cancelación es trasladada, escalada y rotada en fase en el dominio del tiempo, los valores de la señal en el dominio de la frecuencia solamente cambia en la locación de los tonos reservados pero permanecen sin cambio en los otros tonos. El vector de datos X no es afectado por las operaciones de cancelación de picos, de modo que no necesita información lateral o alguna operación en el receptor .

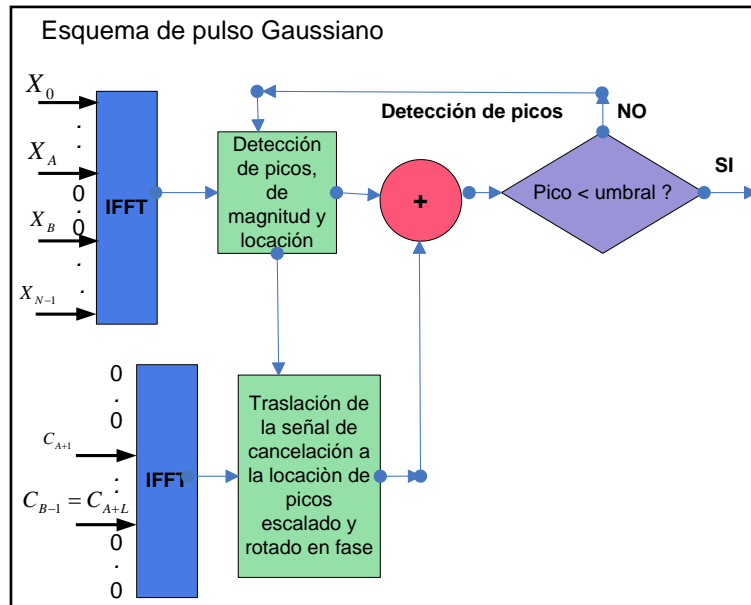


Fig. 3.5.- Esquema de Pulso Gaussiano para reducir P

CAPÍTULO 4

4.- Simulación

4.1 descripciones varias

Para las simulaciones de este estudio se ha basado en el transmisor OFDM para el standard 802.11a de tal forma el símbolo OFDM se distribuye como se explica en el Apéndice C , con 52 portadoras que se utilizan para la transmisión pero de la cual tomaremos un set para usarlos como tonos reservados. El algoritmo creado toma base de el algoritmo de Krishna [3], el cual hace una transmisión OFDM/DMT , en la figura 4.1 podemos observar una serie de pasos por medio del cual se puede ir elaborando la transmisión.

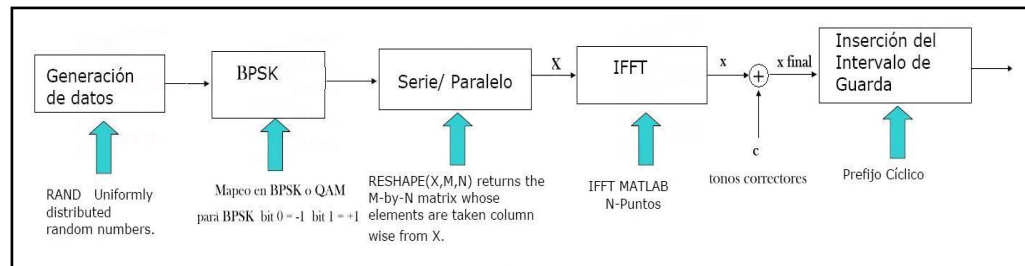


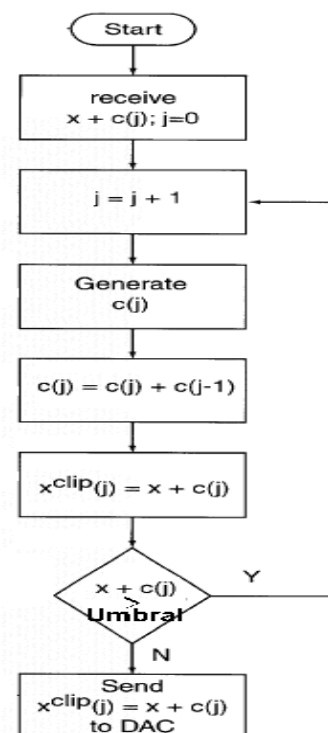
Fig.4.1.- Pasos de transmisión de OFDM con señal correctora c

Algunas de las funciones que observamos son la función random con la cual se crea los datos aleatoriamente para iniciar el proceso de la transmisión , la función Reshape que puede dividir esos datos y es útil para crear los distintos símbolos que se enviaran , también se nota la IFFT que sería la base del proceso de modulación OFDM, y se ha añadido al gráfico la función correctora c que se suma a la x (tratando de manejar la nomenclatura sobre tonos reservados que se vio anteriormente) para formar la x final que sería la señal con la corrección realizada.

El código creado se llama "PAPR reduction with tone reservation (algoritmo gradiente y pulso gaussiano) " que se encuentra en el APENDICE E y lo utilizaremos en todo este proceso de simulación , en el se observa que luego de usar las líneas que inicializan las variables que usara el código (líneas 12 al 31) y de realizar la matriz Q (líneas 34 al 46) se procede a repartir 10000 datos aleatorios de

tal forma que se producen 209 símbolos 802.11a , siguiéndose el estándar (Los primeros 6 y los últimos 5 subsímbolos en el dominio de la frecuencia no son usados y se “rellenan” con cero, las frecuencias de la -26 a la -1 y de 1 a 26, con un total de 52 componentes espectrales se podrán usar para transmisión.) , cada símbolo OFDM vendría a estar “numerado” por el valor “ i ” es decir representara el número del símbolo que estamos usando y las líneas 49 a 86 del código hacen ese proceso de reacomodar los datos al Standard , mapear para bpsk (bpsk usa +1 y -1) y crear la señal “ X_f ” que es la señal OFDM a la cual por medio de la ifft obtenemos el “ x_t ” nuestra señal en el dominio del tiempo a la que haremos el análisis del PAPR.

Fig. 4.2.- Pasos para corrección añadiendo una señal c



Las líneas 89 a 93 encuentran el PAPR de ese símbolo i , y entonces se procederá a la corrección de PAPR en caso de necesitarlo.

4.2.- Observaciones valor UMBRAL y numero de correcciones

En general como se describió en la sección 2.2 tone reservation se basa en tomar un set de tonos para en dichas frecuencias crear una señal correctora C que se suma a la señal $x(t)$ y la modifique para disminuir los picos, esta señal se crea en el código siguiendo el principio de crear , probar y volver a modificar iterativamente tal como se muestra en la figura 4.2 donde se va variando la señal c creada hasta que cumpla el objetivo de que los valores del nuevo $x(t)$ ya no superen los límites.

Cabe anotar que para no tener un retardo excesivo o caer en correcciones infinitas (al corregir se pueden crear picos nuevos) se coloca un límite de iteraciones. (en nuestro código se controla con el valor "maxit").

En el código de matlab utilizado la reducción del PAPR se realiza por dos métodos (algoritmo gradiente y pulso gaussiano) , pero ambos se basan en el mismo principio (tone reservation) por lo que siguen la

misma secuencia descrita de crear una señal correctora c , sumarla a la señal x_t original para crear una nueva señal x_{final} , comprobar nuevamente si esta señal x_{final} cumple el objetivo de no tener picos excesivos, de lo contrario regresa al ciclo de crear la señal correctora c y volverla a sumar para seguir corrigiendo la x_{final} hasta que cumpla el objetivo (o se supere el número de iteraciones permitido).

La verificación de que el x_{final} cumpla las características se hace muestra por muestra, el hecho de haber usado una $ifft$ para pasar de la señal X_f a la x_t hace que por ser un proceso no continuo sino discreto, y dado que la $ifft$ que se usa es de longitud $n_{FFTSize} = 64$, entonces la x_t está repartida en 64 muestras, a cada una de las cuales se les verifica si existe exceso en su magnitud sobre el nivel UMBRAL máximo permitido.

Ese UMBRAL máximo que se utiliza debe ser establecido dependiendo del amplificador que se utiliza y eso dependiendo de los valores de PAPR influenciara a que la señal se clipee o no, tal como se vio en la sección 14.1 donde se habla de los HPA y en el apéndice 2 donde se habla de la función no lineal de la salida que tiene el HPA o función de distorsión, de tal forma que como se muestra en la 1.3 la curva del HPA tiene un límite hasta donde funciona lineal (un límite luego del cual ya se

clippea la señal), en dicha figura se marca ese valor limite con el valor IBO (ecuación 1.17 pag 13) que al relacionarse con el PAPR de manera que el PAPR no debe exceder ese valor IBO para que el HPA funcione correctamente nos lleva a pensar al comparar la formula 1.17 (IBO) con la 1.1 (PAPR) que el $x(t)_{\max}$ no debe exceder el valor de A.

Entonces los picos de la señal $x(t)$ no debe exceder esa señal A tanto para que no exista el excesivo PAPR ni la respectiva saturación del HPA. Dicho valor A en el algoritmo es el valor UMBRAL.

Cabe notar que si comenzamos a variar el valor del umbral que establecemos, podemos por ejemplo tener un umbral bastante bajo lo cual provocara que el algoritmo encuentre una mayor cantidad de picos que superarán el umbral y la corrección se realizará mas veces, dando lugar a un mayor retardo originado por las distintas iteraciones que han aumentado para poder corregirlo. También hay que notar que generalmente se limita el número de iteraciones y en caso de haber un mayor numero de ellas se podría llegar a ese limite con lo cual la corrección deseada no se lograría totalmente, pudiéndose obtener picos de todas formas, caso similar sucede si el número de iteraciones es demasiado pequeño por lo que se llega a el fácilmente deteniendo el proceso de reducción sin haber terminado y quedando aun la existencia

de picos. Para nuestro caso establecimos un límite de iteraciones de 30 dado que no se desea caer en un excesivo retardo del proceso.

En nuestro código también necesitamos establecer un valor UMBRAL en base al cual hacer la corrección, se escogió el valor de 14, este valor es en base al hecho que necesitamos observaciones de cómo se hace la corrección del PAPR en un número significativo de muestras, es decir que por ejemplo si de nuestros 209 símbolos enviados solo hubieran 2 símbolos con picos excedentes no nos servirían tan pocas correcciones para comparar los valores o crear gráficas, el caso excesivo tampoco es favorable, si el umbral es tan grande que aparecen muchos picos, aparecerán incluso varios dentro de un mismo símbolo OFDM por lo que se necesitarían correcciones largas de muchas iteraciones haciendo que la ejecución del código demore mucho y que el número de iteraciones de las correcciones en su mayoría o totalidad sea el número máximo de iteraciones sin llegarse a completar correcciones satisfactorias.

Ahora bien, es dicho valor UMBRAL=14 favorable para ver los resultados de las técnicas del PAPR? Se puede observar las gráficas siguientes generadas con una variación del código hecho (en dicha variación solo se usa algoritmo gradiente y se toma varios valores de

UMBRAL para observar, dicho código también se encuentra en el APENDICE E , el segundo código de esa sección).

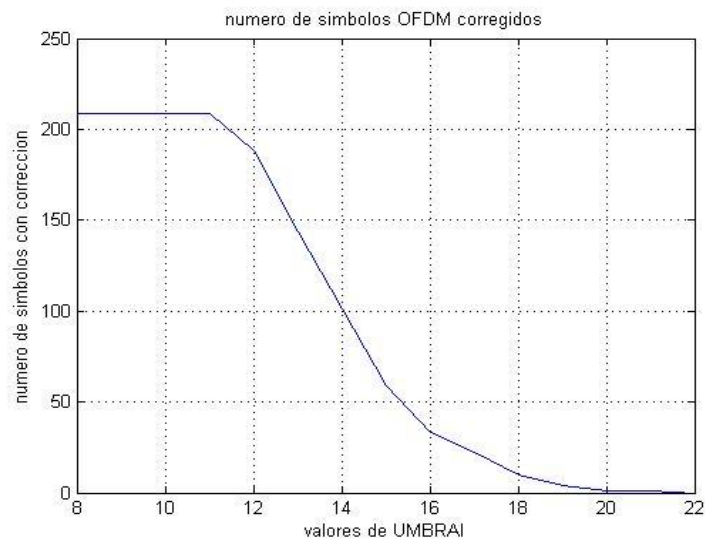


Fig. 4.3.- Numero aproximado de símbolos (de los 209) que necesitan corrección para distintos valores del UMBRAL

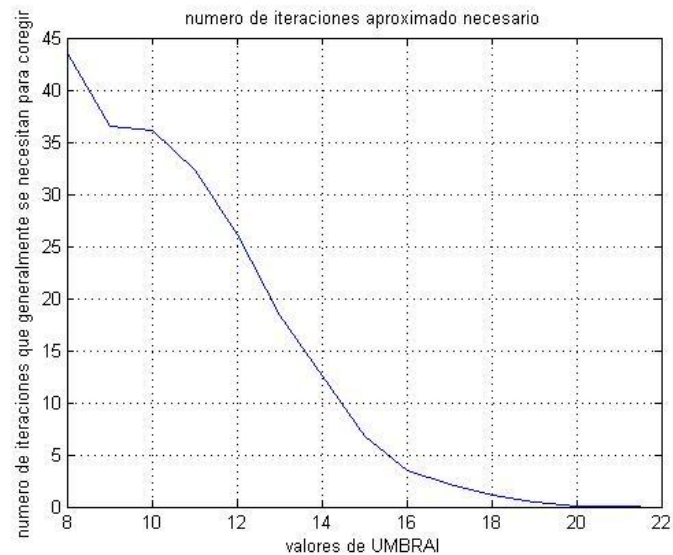


Fig. 4.4.- Número de iteraciones promedio que debe hacerse para corregir un símbolo con pico excesivo , para distintos valores de UMBRAL

En la figura 4.3 podemos observar que un valor de UMBRAL por ejemplo menor que 10 es muy bajo ya que los valores de x_t en su mayoría lo excederán y necesitara correcciones en prácticamente todos los símbolos (en la grafica 4.4 dicho valor UMBRAL menor de 10 muestra que cada corrección necesitara mas de 30 iteraciones, superando el numero limite de iteraciones propuesto), un valor UMBRAL de 20 en cambio es muy alto y hace que prácticamente los valores de x_t no lleguen a el por lo que no aparecen picos excedentes para corregir. El valor UMBRAL=14 da un valor aproximado de 100 símbolos que necesitan corregirse de los 209, dando un numero apropiado de muestras para hacer graficas y comparaciones y correcciones con un numero de iteraciones promedio adecuado.

Como se dijo anteriormente realizando la verificación muestra a muestra del x_t es que se verifica la existencia de los picos excesivos (sobre el UMBRAL) y se procede, en el caso de detectarlos, a la corrección por medio de la creación y añadidura de la señal c produciendo la nueva x_{final} con el proceso iterativo antes descrito.

Para la creación de dicha señal c en el set de tonos reservados se utiliza los 2 métodos revisados anteriormente, algoritmo gradiente y pulso gaussiano.

4.3.- Reducción por tonos reservados usando algoritmo gradiente:

Regresando a la revisión de nuestro código de matlab (apéndice E) observamos que las líneas 105 a 143 se encargan de la revisión de la señal x_t y la creación de la señal correctora c .

El proceso de creación de la señal c esta basado en el algoritmo gradiente visto anteriormente y utiliza la expresión (3.22) de la pag 42.

$$c^{(i+1)} = c^{(i)} - \mu \sum_{|x_n + c_n^{(i)}| > A} (x_n + c_n^{(i)} - Ae^{j \arg\{x_n + c_n^{(i)}\}}) Q' q_n^{row}$$

Para formar la señal correctora c , la misma que va cambiando iteración por iteración según se observen mas picos en el símbolo (que se revisa muestra por muestra) o se creen nuevos picos al efectuarse una corrección.

Sin embargo en la expresión observamos el valor μ , que es el tamaño de paso, es decir que nos sirve para ajustar que tan rápido debe converger el algoritmo, hay distintas formas de hacer variar dicho valor u optimizarlo pero dichos casos variados no se tomaran en

consideración para este estudio, en el código de matlab realizado dicho valor se expresa como "Niu" y esta fijado en un valor de 14, se puede mostrar los efectos tomando varios valores de "Niu" , a continuación se muestran los gráficos de 3 corridas de un código de matlab basado en el principal (tal como se hizo para el valor UMBRAL) pero centrándose en ver los efectos al variar Niu (el código también esta en el APENDICE E 3er codigo).

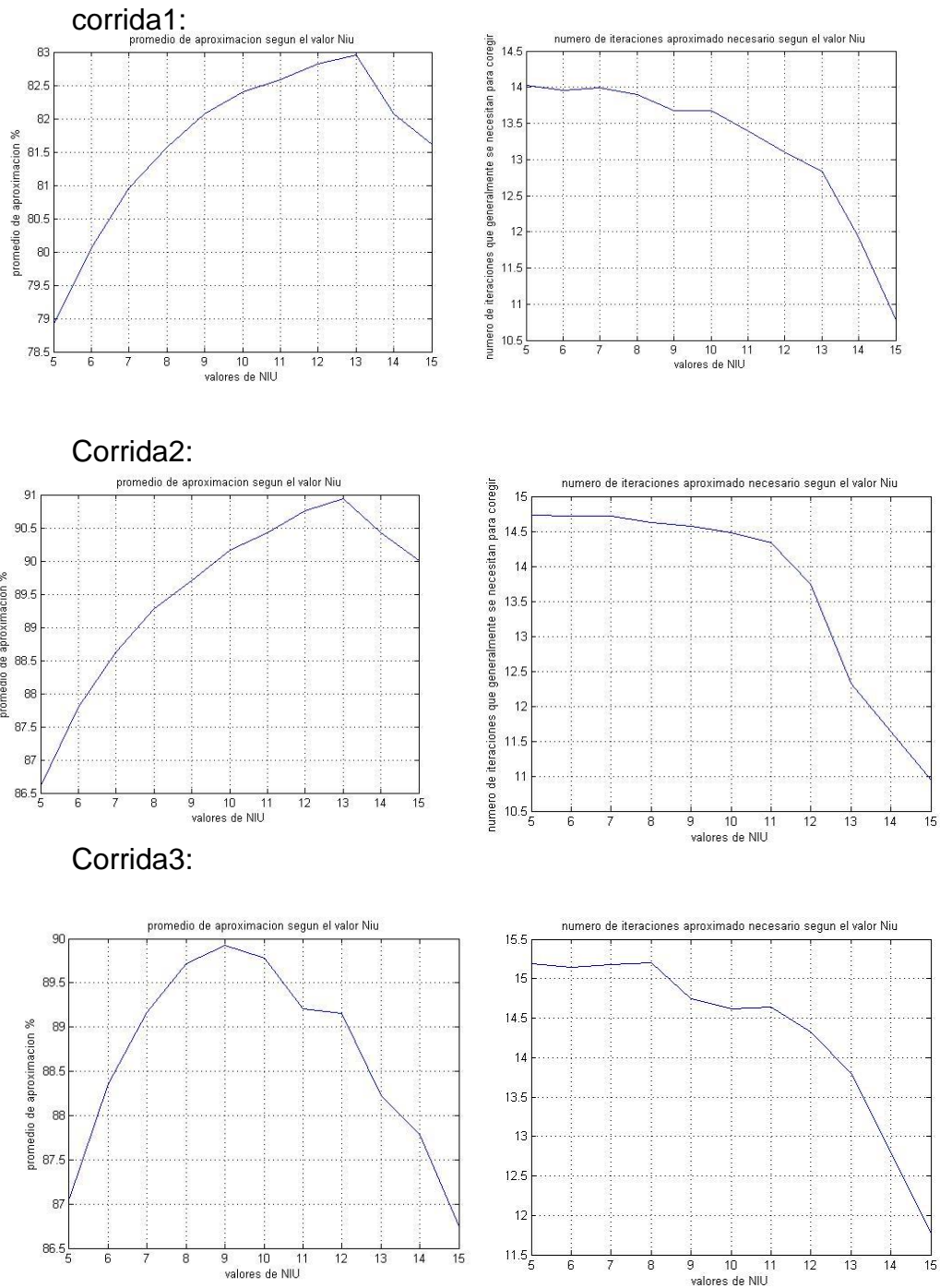


Fig. 4.5.- Aproximación e iteraciones de la reducción de pico para varios valores de Niu

La figura 4.5 muestra que tan cerca deja la corrección del pico al valor del UMBRAL (en %) , un 100% será que la reducción fue 100% exacta, dichos valores de la figura son en promedio por las reducciones que se hallan hecho (en todos los símbolos donde se efectúa corrección) para cada valor “Niu” escogidos, se muestra las graficas de 3 corridas y las que acabamos de nombrar son las de la izquierda, mientras que las de la derecha muestran en promedio que tantas iteraciones se necesitan para realizar esas correcciones. Con dichas graficas de la parte derecha se puede observar que a mayor Niu la corrección se hace mas rápidamente, esto es lo deseable, que el algoritmo tenga un menor numero de iteraciones hasta alcanzar su objetivo (converge mas rápido) , sin embargo por la grafica de la izquierda podemos notar que si incrementamos ese Niu demasiado, se encontrara la tendencia a no tener una corrección muy exacta. El valor establecido de Niu=14 da un número de iteraciones y aproximación aceptable. (en 13 se notaba que en muchas muestras el numero de iteraciones estaba cercano o era el numero máximo y en valores mayores ya se notaba tendencia a una corrección no muy exacta)., por eso es el que se utilizo.

Siguiendo con el código principal: Las líneas 146 a 179 se encargan de mostrar que el algoritmo consiguió la reducción del pico de x_t por medio

de la señal c , es así que corriendo el código podemos observar la corrección de un símbolo en la primera grafica obtenida:

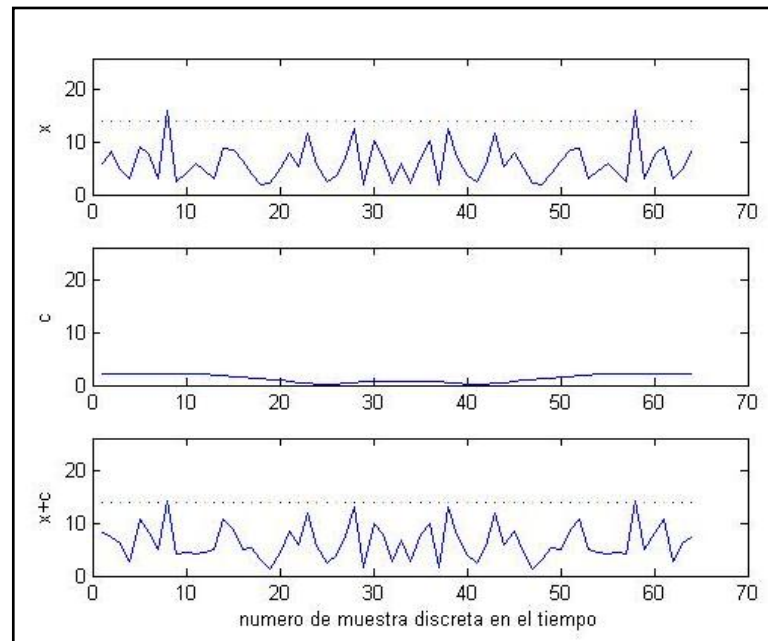


Fig. 4.6.- Reducción de picos utilizando algoritmo gradiente

En la figura 4.6 vemos 3 gráficas, la superior representa la señal de información original x , la gráfica intermedia la señal reductora c , y la gráfica inferior representa la señal reducida, o sea la señal $x' = x + c$ es decir la x final se nota claramente la corrección de 2 picos con el algoritmo, cabe notar que el código muestra la grafica del primer símbolo en que se encuentra necesaria una corrección (siempre lo hará así) , el resto de datos de corrección quedaran guardados en variables que se especificaran mas adelante.

La siguiente grafica que el código muestra es la ubicación de las frecuencias:

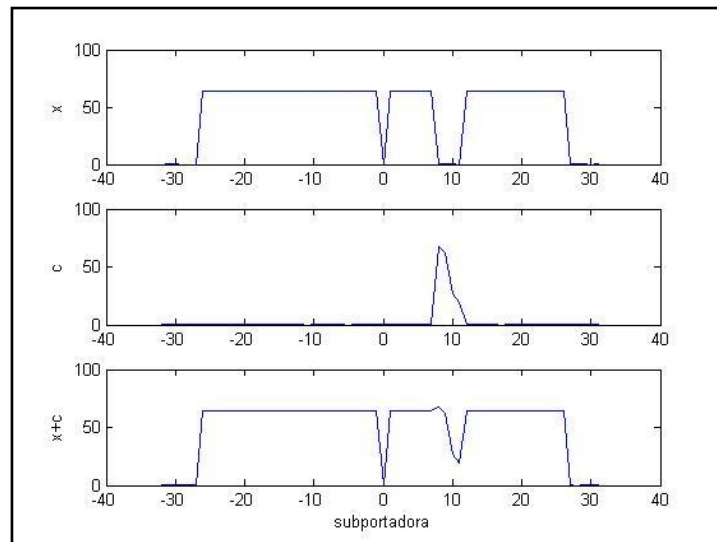


Fig. 4.7.- Gráficas de subportadoras utilizadas (frecuencia)

La señal c (la de en medio) ocupara justamente las subportadoras que se asignaron para los tonos reservados.

4.4.- Reducción por pulso gaussiano

La otra técnica que se usara para la obtención de la señal correctora c es la de pulso gaussiano, como se vio en la sección 3.2, esta técnica crea el pulso en la frecuencia (usando los tonos reservados) aprovechando la característica de que la transformada inversa de Fourier también será un pulso gaussiano. Los pasos del algoritmo también están descritos en la sección 3.2.1, en el código Matlab la sección que se dedica a la corrección por pulso gaussiano esta desde la línea 194 a 226 , para la corrección se sigue los pasos vistos pero con las siguientes observaciones:

- al pulso gaussiano en frecuencia no se lo dejo con amplitud 1, dado que la señal c si no logra la corrección al primer intento sigue creando pulsos que se suman pero que sus pulsos en el tiempo son trasladados a las muestras con picos, pero en frecuencia se mantienen en el mismo rango, por lo que se tiene un crecimiento de la magnitud en frecuencia, tal que si la amplitud del pulso es 1 se va sumando y formando una magnitud mas grande, por eso se ha puesto amplitud 0.1 al pulso en el dominio de la frecuencia.

- se debe notar también que matlab utiliza para sus matrices y vectores subíndices que comienzan con 1 y no con 0, pero las formulas vistas en la sección de pulso gaussiano parten desde el pulso con centro en 0 , por eso al tener que trasladara en frecuencia se usa en lugar de $L/2$ el valor $(L/2)-1$ y para centrarla $(L-1)/2$ (considerando que las señales tanto $x(t)$ como c y el pulso gaussiano comience justo en el valor de 1 y antes de eso no exista nada)

La corrida del código (líneas 242 a 274) muestra el tercer y cuarto gráficos, el tercero (figura 4.8) con los resultados de la corrección con pulso gaussiano del primer símbolo que se encuentra con problema de pico excesivo:

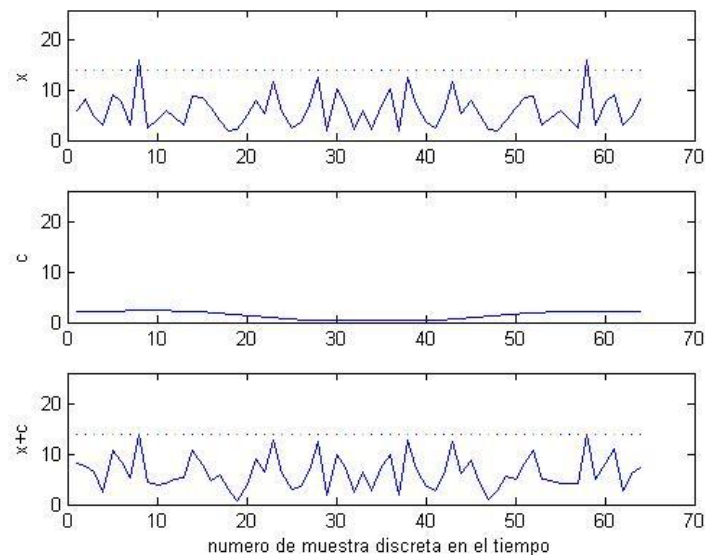


Fig. 4.8.- Reducción de picos que sobrepasan el Umbral usando pulso Gaussiano

La gráfica obtenida en la figura 4.8 es similar a la 4.6, obteniéndose los resultados parecidos mediante métodos diferentes

En lo que respecta a la cuarta grafica (figura 4.9) es también la forma del puso en la frecuencia y mostrándose como se utilizan los índices reservados para los tonos de la señal correctora.

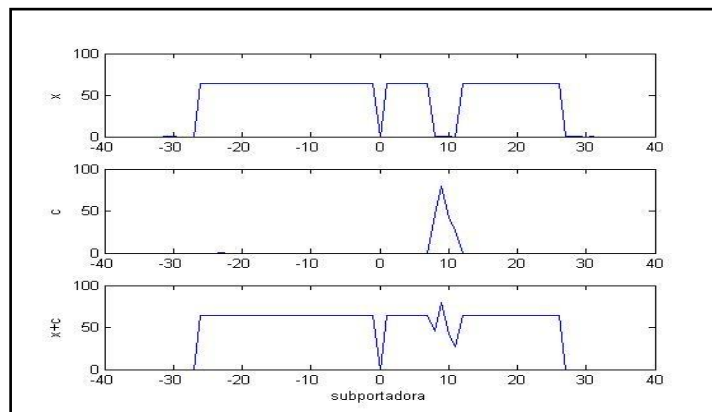


Fig. 4.9.- Subportadoras utilizadas (frecuencia) para pulso gaussiano

4.5.- Valores comparativos de ambos algoritmos

Las líneas 277 a 281 se encargan de calcular y guardar los valores del papr cuando el símbolo ya fue modificado con la corrección respectiva, lo mismo sucedía en las líneas 182 a 186, es así que los valores originales del PAPR de la señal x_t quedan guardados en la variable `paprSymboldB(ii)` (recordar que `ii` es el símbolo que se esta tratando) , el

PAPR de la señal x_{final} ya corregida con algoritmo gradiente esta en $paprMODSymboldB(ii)$ y la de el PAPR de la señal x_{final2} modificada con pulso gaussiano esta en $paprMOD2SymboldB(ii)$. Todos esos datos del $papr$ se usan al final para poder graficar las funciones de distribución acumulativa complementaria del $papr$ antes y después de las correcciones, dicha sección son las líneas 291 a 323 con las gráficas de la figura que se muestra :

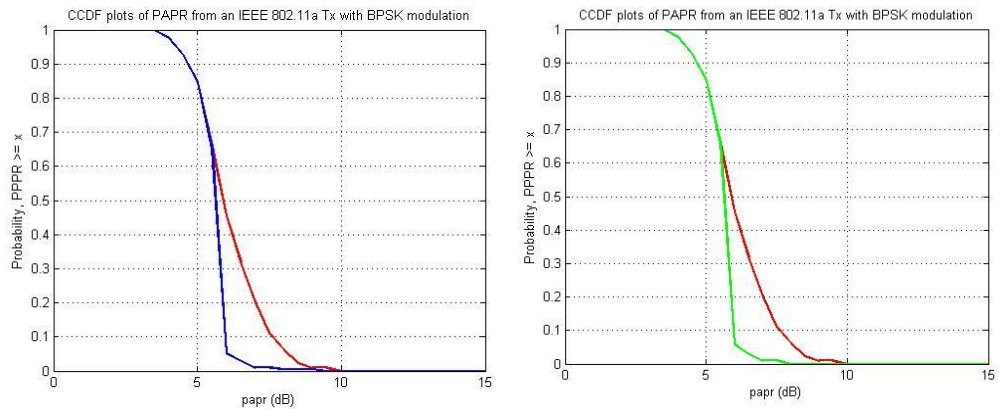


Fig. 4.10.- CCDF del $papr$ para algoritmo gradiente y pulso gaussiano

La línea roja es la distribución de la señal original , la azul es la grafica de la distribución de $papr$ cuando se uso la técnica de tonos reservados con algoritmo gradiente, y la curva verde es la distribución de $papr$ tonos reservados creados con pulso gaussiano.

Se nota en ambos casos la corrección tal que valores de papr mayores a un valor aproximado de 6db son de muy baja probabilidad, es decir la técnica logro la limitación de papr hasta cierto valor disminuyendo grandemente los casos no deseados.

Cual técnica resulto mejor? A simple vista (figura 4.5, 4.8 y 4.10) ambos algoritmos producen los mismos efectos, sin embargo para notar mas los resultados de cada una de las técnicas se a aprovechado de guardar los datos no solo de las observaciones de una corrección sino de todas las correcciones realizadas , y como vimos anteriormente se ajusto la variable UMBRAL para contar con un buen numero de muestras con un numero de iteraciones aceptables (con UMBRAL=14 se obtenían aproximadamente 100 correcciones) , es así que para el caso de esta corrida se observo que de los 209 símbolos que se enviaron , se realizo la corrección de PAPR a 104 símbolos (para este caso).

En el transcurso del código de Matlab se ha aprovechado de guardar datos justamente para notar más los efectos de cada técnica.

Es así que se pueden ver en el contenido del código los siguientes valores:

EXCESO:

Lo ponemos como el valor que se paso el PAPR de su valor limite marcado por el IBO (este ultimo también llamado clip level) . Estaría dado por el $ABS(PAPR - IBO)$ (nótese que el valor A dentro de la formula 1.17 del IBO es el UMBRAL) . El valor $E\{|x|^2\}$ varia en cada símbolo.

CORRECCION1:

Guarda los valores de cuanto variación se logro en el papr con la corrección con respecto al papr sin la corrección, usando el algoritmo gradiente.

CORRECCION2:

Guarda los valores de cuanto variación se logro en el papr con la corrección con respecto al papr sin la corrección, usando la creación del pulso gaussiano.

Numiteraciones1:

Guarda cuantas iteraciones se tuvieron que hacer para lograr la corrección de papr de cada símbolo con el algoritmo gradiente.

Numiteraciones2:

Guarda cuantas iteraciones se tuvieron que hacer para lograr la corrección de papr de cada símbolo con el pulso gaussiano.

Al final de la corrida del código se pueden rescatar todos estos datos, tal que para la corrida que estamos mostrando se procedió a organizarlos en una tabla de Excel y finalmente todos estos datos se pueden observar en el APENDICE F , donde esta al tabla con los valores para cada símbolo .

Es interesante reorganizar la información , quitando los datos donde no se efectúo corrección y dejando solo los de los 104 símbolos donde si hubo, se obtiene la siguiente tabla donde se ha ordenado en base a la magnitud de PAPR que se tenia que corregir (EXCESO) y se puso el porcentaje de aproximación que cada técnica tuvo con el valor deseado (tecnica1 es algoritmo gradiente y tecnica2 pulso gaussiano) :

simbolo enviado	PAPR	exceso sobre el nivel del PAPR maximo deseado	correccion con algoitmo gradiente	correccion con pulso gaussiano	porcentaje de acercamiento de la correccion1 a lo deseado	porcentaje de acercamiento de la correccion2 a lo deseado
ii	paprSymboldB'	EXCESO'	CORRECCION1'	CORRECCION 2'	porcentaje aproximacion con tecnica1	porcentaje aproximacion con tecnica2
98	6,1159	0,0058	0,0058	0,0058	100,00%	100,00%
175	6,1360	0,0259	0,0259	0,0259	100,00%	100,00%
131	6,1666	0,0565	0,0565	0,0565	100,00%	100,00%
190	6,2833	0,0818	0,0818	0,0818	100,00%	100,00%
32	6,1946	0,0845	0,0686	0,0707	81,18%	83,67%
110	6,1965	0,0863	0,0863	0,0863	100,00%	100,00%
9	6,2060	0,0959	0,0959	0,0959	100,00%	100,00%
171	6,2238	0,1136	0,1136	0,1136	100,00%	100,00%
86	6,2384	0,1282	0,1282	0,1282	100,00%	100,00%
53	6,2478	0,1377	0,1377	0,1377	100,00%	100,00%
135	6,2637	0,1535	0,1535	0,1535	100,00%	100,00%
16	6,2750	0,1648	0,1801	0,1648	90,72%	100,00%
178	6,3717	0,1701	0,1701	0,1701	100,00%	100,00%
145	6,2891	0,1790	0,1790	0,1790	100,00%	100,00%
143	6,2947	0,1846	0,1846	0,1846	100,00%	100,00%
105	6,3111	0,2009	0,2009	0,2009	100,00%	100,00%
204	6,4088	0,2073	0,0687	0,0355	33,14%	17,12%
70	6,3437	0,2335	0,3752	0,2335	39,31%	100,00%
64	6,3547	0,2445	0,2445	0,2445	100,00%	100,00%
22	6,3664	0,2563	0,2563	0,2563	100,00%	100,00%
189	6,4726	0,2711	0,2711	0,2711	100,00%	100,00%
52	6,4191	0,3090	0,3559	0,3090	84,82%	100,00%
31	6,4274	0,3172	0,3172	0,3172	100,00%	100,00%
159	6,4296	0,3194	0,3194	0,3194	100,00%	100,00%
166	6,4321	0,3219	0,3219	0,3219	100,00%	100,00%
146	6,4484	0,3382	0,3367	0,3382	99,56%	100,00%
38	6,4547	0,3446	0,3446	0,3446	100,00%	100,00%
185	6,5541	0,3525	0,3521	0,3525	99,89%	100,00%
183	6,5613	0,3597	0,3597	0,3597	100,00%	100,00%
56	6,4991	0,3889	0,3612	0,3729	92,88%	95,89%
149	6,5170	0,4068	0,4068	0,4068	100,00%	100,00%
174	6,5424	0,4322	0,4322	0,4322	100,00%	100,00%
20	6,5664	0,4563	0,3953	0,4431	86,63%	97,11%
176	6,5904	0,4803	0,4803	0,4803	100,00%	100,00%
120	6,6446	0,5345	0,5922	0,5345	89,20%	100,00%
103	6,6605	0,5504	0,5778	0,5504	95,02%	100,00%
201	6,7929	0,5913	0,6076	0,5913	97,24%	100,00%
117	6,7037	0,5936	0,4834	0,4476	81,44%	75,40%

Tabla II : resultados de simulación reorganizados

197	6,7959	0,5943	0,5923	0,5943	99,66%	100,00%
181	6,8008	0,5993	0,5993	0,5993	100,00%	100,00%
107	6,7594	0,6493	0,6493	0,6493	100,00%	100,00%
203	6,8531	0,6516	0,6513	0,6516	99,95%	100,00%
18	6,7659	0,6558	0,6558	0,6558	100,00%	100,00%
206	6,8697	0,6682	0,6682	0,6682	100,00%	100,00%
126	6,7973	0,6872	0,6868	0,6872	99,94%	100,00%
77	6,8354	0,7252	0,6188	0,6887	85,33%	94,97%
42	6,9039	0,7937	0,7680	0,7913	96,76%	99,70%
82	6,9070	0,7968	0,7968	0,7968	100,00%	100,00%
75	6,9095	0,7993	0,7993	0,7993	100,00%	100,00%
147	6,9619	0,8518	0,0775	0,1213	9,10%	14,24%
15	6,9883	0,8781	0,8781	0,8781	100,00%	100,00%
122	7,0014	0,8913	0,8166	0,8832	91,62%	99,09%
29	7,0313	0,9212	0,9212	0,9212	100,00%	100,00%
115	7,0325	0,9223	0,9223	0,9223	100,00%	100,00%
44	7,0390	0,9288	0,8333	0,8179	89,72%	88,06%
6	7,0456	0,9355	1,2821	0,9355	62,95%	100,00%
139	7,0517	0,9416	1,2418	0,9416	68,12%	100,00%
50	7,0805	0,9704	0,9704	0,9704	100,00%	100,00%
193	7,1822	0,9806	0,8764	0,8647	89,37%	88,18%
140	7,0909	0,9807	0,9807	0,9807	100,00%	100,00%
168	7,1031	0,9930	0,9930	0,9930	100,00%	100,00%
7	7,1471	1,0370	1,0370	1,0370	100,00%	100,00%
13	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
35	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
59	7,2700	1,1598	1,1476	1,1598	98,95%	100,00%
68	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
80	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
99	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
104	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
114	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
123	7,2700	1,1598	0,6695	0,4778	57,73%	41,20%
130	7,2700	1,1598	1,1598	1,1598	100,00%	100,00%
51	7,2748	1,1647	1,1647	1,1647	100,00%	100,00%
137	7,3254	1,2153	1,7551	1,2153	55,58%	100,00%
3	7,3478	1,2376	1,2376	1,2376	100,00%	100,00%
136	7,3573	1,2471	1,1964	1,2419	95,93%	99,58%
37	7,4143	1,3042	1,3042	1,3042	100,00%	100,00%
74	7,5383	1,4281	0,7641	0,9261	53,50%	64,85%
167	7,5760	1,4659	1,4659	1,4649	100,00%	99,93%
55	7,6896	1,5794	1,5794	1,5794	100,00%	100,00%
134	7,7024	1,5922	1,5922	1,5922	100,00%	100,00%
102	7,7844	1,6742	1,5188	1,4226	90,72%	84,97%

Continuación de Tabla II

157	7,7902	1,6801	1,7974	1,6801	93,02%	100,00%
209	7,8880	1,6864	1,6864	1,6864	100,00%	100,00%
95	7,8017	1,6916	1,1994	1,0814	70,90%	63,93%
161	7,8029	1,6927	1,8075	1,6927	93,22%	100,00%
192	7,9436	1,7420	1,7420	1,7420	100,00%	100,00%
138	7,9080	1,7978	0,9501	0,9449	52,85%	52,56%
118	8,0227	1,9125	1,9125	1,9125	100,00%	100,00%
170	8,0997	1,9896	1,9896	1,9896	100,00%	100,00%
158	8,1824	2,0723	2,0707	1,9970	99,92%	96,37%
4	8,2930	2,1829	2,1829	2,1829	100,00%	100,00%
57	8,2930	2,1829	2,6614	2,1829	78,08%	100,00%
90	8,2930	2,1829	2,1829	2,1829	100,00%	100,00%
111	8,2930	2,1829	2,1829	2,1829	100,00%	100,00%
60	8,3207	2,2106	2,2103	2,2054	99,99%	99,76%
93	8,3463	2,2362	2,2048	2,1934	98,60%	98,09%
188	8,6002	2,3986	2,3986	2,3986	100,00%	100,00%
49	8,6700	2,5599	2,5573	2,5599	99,90%	100,00%
164	8,8426	2,7324	2,7324	2,7324	100,00%	100,00%
67	9,2082	3,0980	1,3902	1,3961	44,87%	45,06%
76	9,2082	3,0980	3,0980	3,0980	100,00%	100,00%
69	10,0360	3,9259	3,1665	2,7881	80,66%	71,02%
113	10,0360	3,9259	0,7485	1,9685	19,07%	50,14%

	porcentaje de acercamiento de la tecnica1 a lo deseado	porcentaje de acercamiento de la tecnica2 a lo deseado
tecnica1 = tonos reservados con algoritmo gradiente	91,80%	94,43%
tecnica2= tonos reservados con pulso gaussiano		

Continuación de Tabla II

A simple vista revisando los valores de la tabla del APENDICE F o la mostrada aquí ya se observa (columnas CORRECCION1 CORRECCION2 y Numiteracones1 Numiteraciones2) que tanta corrección realizó cada técnica en particular para cada símbolo y por ejemplo en algoritmo gradiente (tecnica1) se ve que en repetidas ocasiones se llegó al límite de iteraciones (aunque a pesar de eso se logró una aceptable corrección), también se nota por ejemplo como en variadas ocasiones la tecnica2 (pulso gaussiano) produce un número de iteraciones menor que algoritmo gradiente.

En las 2 últimas columnas de la tabla anterior se ha agregado el cálculo de que tan cerca resultaron las corrección con respecto al ρ_{pr} que se debía reducir (EXCESO), se puede observar como la técnica de tone reservación con algoritmo gradiente resulta en promedio con aproximaciones de 91,80% mientras que con pulso gaussiano la aproximación a cumplir la meta es en promedio 94,43%, superando aunque en poco valor al otro algoritmo.

CAPÍTULO 5

5.- Análisis de Resultados

5.1.- resumen

La modulación OFDM/DMT, una de las mas prometedoras para las tecnologías de comunicación a implementarse en la actualidad posee un gran problema de Peak.to-average power ratio demasiados altos y este problema se ve agravado a medida que un número mayor de portadoras se utilice (como se observó en la sección 1.2) ya que aumenta según el número de portadoras, dichos valores altos pueden dar origen a distintos problemas como en los amplificadores de potencia los cuales se ven en la necesidad de trabajar en una mayor zona lineal y por ende elevar sus costos y sus consumos de potencia (mas IBO), porque de no hacerlo el amplificador podría limitar la señal recortándola y ocasionando

pérdida de información y ruido fuera de banda. Por esto se han implementado distintos métodos para efectuar la reducción de dichos picos y en nuestro trabajo hemos decidido examinar la técnica de Tone Reservation, una técnica que permite generar una señal reductora de esos picos, pero dicha señal se genera con un set de tonos ya reservados para esa finalidad.

La elaboración de esa señal reductora a su vez puede conseguirse mediante diversos algoritmos utilizando en este caso el algoritmo del gradiente (ligado a la derivada direccional de la señal) y la creación de un pulso gaussiano.

Se han realizado pruebas de simulación en MATLAB elaborando un código que imite el comportamiento de la transmisión OFDM/DMT para lo cual se debió escoger un Estándar y en este caso el 802.11a, es así como se puede comprobar finalmente las virtudes de la técnica de tonos reservados dado que la figura 4.10 muestra como con ambos algoritmos se logra limitar la probabilidad de que el PAPR supere ciertos valores, así como con la reorganización de los datos obtenidos se puede observar un promedio de reducciones de 91,80% y 94,43%, lo cual indica que los objetivos de reducir el papr hasta ciertos valores se cumplen con bastante cercanía.

5.2.- Respecto al número de tonos:

Un valor que no se revisó con anterioridad pero que marca el comportamiento de la técnica de tonos reservados es el número de tonos donde se crea la señal c , puede venir a la mente la idea de que un mayor número de tonos formaría una señal correctora c más exacta y por ende una corrección mejor, la verdad el aumento de tonos reservados sí puede ligarse a la obtención de ventajas en la técnica. Por ejemplo para el caso de algoritmo gradiente, tal como se hizo con anterioridad para otros parámetros, podemos utilizar nuestro código principal de matlab y hacer las variaciones respectivas creando un código que nos muestre los efectos de variar el número de tonos, dicha variación se hizo, para observar que tan rápida era la corrección de tonos al variar el número de ellos (el código también se encuentra en el APENDICE E 4to código) y en tres corridas del código se muestra las gráficas siguientes:

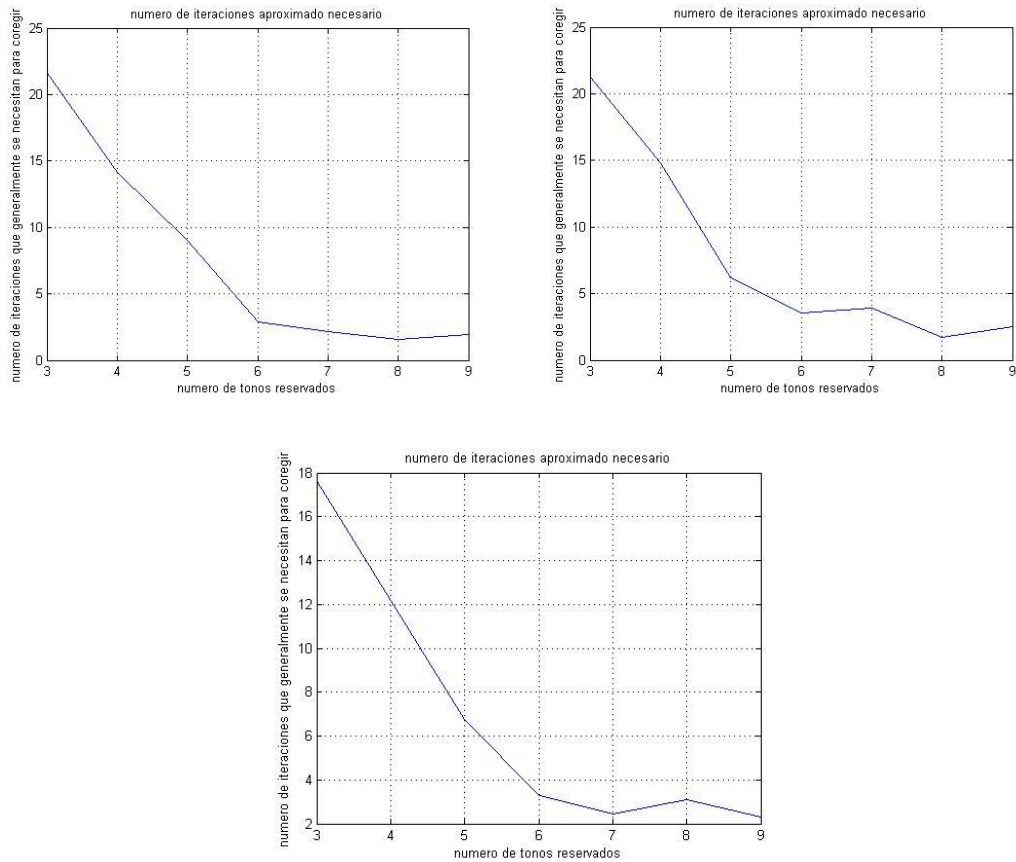


Fig. 5.1.- Número de iteraciones promedio para realizar corrección según ntonos

Se puede observar que para el algoritmo gradiente con los valores que se establecieron anteriormente del UMBRAL y NIU, al variar tonos desde un numero de 3 hasta 9 , el numero de iteraciones va disminuyendo, es decir provocaría que el algoritmo tenga una convergencia mas rápida.

No se ha realizado el caso de pulso gaussiano, pero sin embargo se puede notar algo ligado a esta técnica, y es con respecto al ancho del pulso gaussiano (la varianza) , para esto nótese la figura 3.4 que se halla en la sección de pulso gaussiano (3.2) se puede observar como con el pulso gaussiano creado en el set de tonos reservados se crea una señal angosta en el tiempo para el número de 4 muestras (por sacar la ifft de esas 4 subportadoras) esto es debido a que si se observa las formulas 3.32 y 3.33 de dicha sección, con la ifft el valor M pasa de numerador a denominador, y dado que este valor está ligado a la varianza lleva al efecto de que el pulso en el tiempo tenga una varianza que se relaciona de forma inversa con la varianza en el plano de frecuencia, es decir, si la varianza del pulso en frecuencia tiene un valor alto (tal como se ve en dicha figura ya que el pulso ocupa con su ancho todos los tonos) entonces la varianza en el tiempo tiene un valor pequeño y resulta la forma angosta de pulso que se observa en la grafica.

Dicho caso ya no sucede al colocar nuestro pulso en su ubicación respectiva con respecto a las demás subportadoras, es decir, ya no será un pulso que ocupe toda la gama de subportadoras, sino que, por ejemplo para nuestro caso solo ocupara un ancho de 4 subportadoras de las 64 que están, eso implica que se vea un pulso angosto en

frecuencia (menos varianza) y por lo tanto, al estar relacionado inversamente, la varianza del pulso en el tiempo se incrementa formando un pulso mas ancho en el tiempo.

Para visualizar este efecto vamos a hacer al igual que en la sección 3.2 la sección de código que crea el pulso gaussiano en frecuencia pero esta vez lo ubicaremos en el lugar que debe entre el total de subportadoras para después obtener el pulso en el tiempo por la ifft.

El código es el siguiente:

```
clear
nToneRes=4;
nFFTSIZE=64 ;
tonereservIndex=[8 9 10 11];
kk2 = 21;
n=[1:1:nFFTSIZE]';
xfinal2(kk2)=1.2 ;

L = nToneRes ;
alpha = 1;
m=(1:1:L);
p_gaussf= exp (-0.5*( (alpha * ( m-(L/2)-1 ) / ((L-1)/2 ) ).^2 )) ; %creacion de pulso
gaussiano
CF2=zeros(nFFTSIZE,1);
CF2(tonereservIndex+nFFTSIZE/2+1)=p_gaussf ; %s colocando el pulso en los tonos
reservados de entre todas las subportadoras
plot(n,abs(CF2))
title('Pulso en el dominio de la frecuencia')
xlabel('subportadora')
ylabel('magnitud')

ct2=ifft(CF2); % paso a pulso en el tiempo
ct2=ifftshift(ct2);
figure(2)
plot(n,abs(ct2))
title('Pulso en el dominio del tiempo')
xlabel('muestra')
ylabel('amplitud')
```

Y las graficas obtenidas:

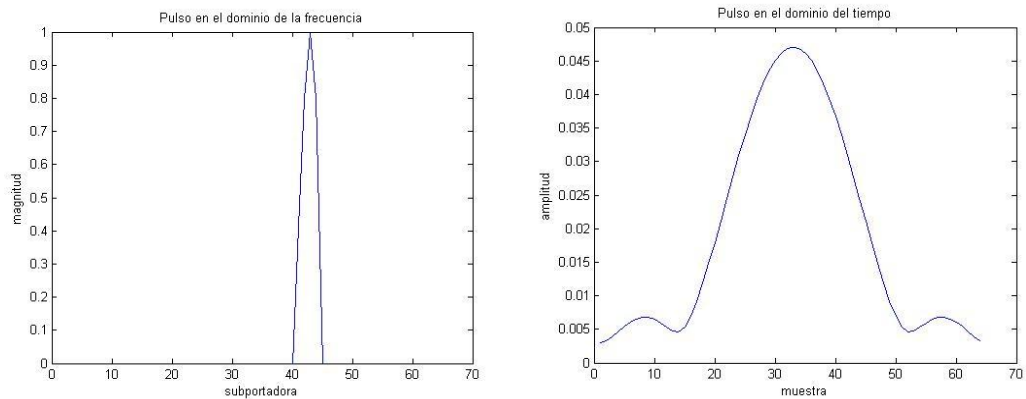


Fig. 5.2.- Pulso gaussiano de la señal correctora c en frecuencia y tiempo

Y ahora se observa que la grafica no es igual a la figura 3.4, es lo contrario, ya que esta vez como los tonos reservados están en solo una pequeña parte del total de tonos la varianza es pequeña y el pulso en frecuencia se ve angosto, y en el tiempo se vuelve lo contrario, o sea un pulso ancho, eso indica que el pulso gaussiano en el tiempo se expande a un mayor numero de muestras, por lo que al colocarlo en un pico puede afectar las muestras aledañas a el pudiendo causar el crecimiento de nuevos picos.

Si ya se tiene establecido el numero de tonos reservados la única forma de minimizar este efecto es usarlos en su totalidad con el pulso creado, es decir la varianza debe ocupar lo mayor posible, porque si el pulso se crea angosto tal que no ocupe todos los tonos reservados la varianza será tan pequeña que el pulso en el tiempo se ensanchara aun mas.

Para este caso también se puede pensar en el efecto de aumentar el número de tonos reservados, los resultados serían que pudiendo usar un mayor número de tonos para nuestro pulso en frecuencia la varianza (ancho del pulso) aumentaría, trayendo su respectiva disminución del ancho del pulso en el dominio del tiempo para que el mismo no interfiera tanto con las demás muestras aledañas al pico a corregir.

Entonces podría pensarse que finalmente aumentar el número de tonos reservados es lo deseado, menor iteración con algoritmo gradiente, mejor señal correctora con pulso gaussiano, es decir trae ventajas ligadas a ello y sería lo correcto, pues no necesariamente.

La alternativa no sería muy viable en este caso, no se puede elevar grandemente el número de tonos reservados porque está ligado a la pérdida de tasa de datos, los tonos reservados se toman de entre los tonos (subportadoras) que tenemos, y como se vio en el estándar para nuestro caso (802.11a) son 52 las subportadoras que llevan información por lo que si escogemos 4 de ellas tendríamos $4/52 \cdot 100 = 7,69\%$ menos en nuestra capacidad de enviar bits de información útil y a medida que incrementamos tonos nuestra capacidad disminuirá más, dado que se desea es enviar mayor información entonces el incremento

de los tonos no sería una opción muy indicada, lo cual es mas viable en caso de otro sistemas con un numero de subportadoras mucho mayor.

5.3.- Existe relación entre la cantidad de PAPR excesivo a corregir y el numero de iteraciones?

Una observación interesante y que uno podría llegar a pensar es que si el pico de x_t se excede cada vez mas del UMBRAL entonces se va a demorar mas la corrección, aprovechando los datos que se reordenaron en el archivo Excel (APENDICE F) (que también se encuentran en `tablal` y `tablall`), podemos obtener una curva de cuantas iteraciones se efectuaron dependiendo de cuanto se excedía el pico, dicha curva se muestra a continuación.

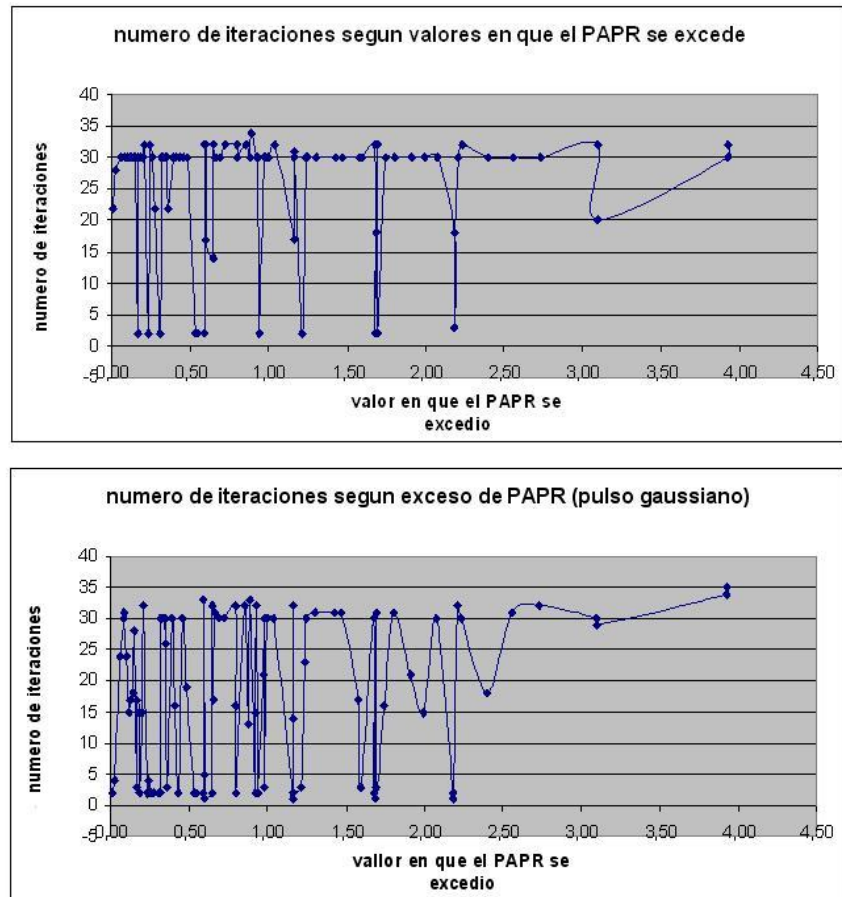


Fig. 5.3.- Gráfica de numero de tonos con respecto al exceso de PAPR que se corrigió para los algoritmo gradiente (arriba) y pulso gaussiano (abajo)

Puede notarse que por lo menos para valores de PAPR que se han excedido desde 0.006 hasta por ejemplo 3,5 (no hay demasiadas muestras mas allá de esos valores y la probabilidad de que ocurran variaciones mas grandes cada vez es menor debido a la distribución de probabilidad que tiene el PAPR), no se puede concluir que si el PAPR se excedió mas necesitó mas numero de iteraciones para corregirse, no es valida la afirmación ya que se muestra que las curvas toman valores

variados de número de iteraciones sin importar el valor en que el PAPR se halla sobrepasado, El incremento en un numero de iteraciones probablemente se incremente es por la forma del símbolo, es decir, el número de picos que se encuentren en un mismo símbolo, a que distanciamiento están los picos, y que picos nuevos se crean al ir haciendo las correcciones.

5.4.- Comparación de resultados con ambos algoritmos

Al mostrar resultados de 2 técnicas es lógico interesarse en cual de ellas esta logrando mejores resultados. Anteriormente se mostró que los resultados de disminución de papr con ambos métodos son parecidos, y luego se obtuvo que en promedio el algoritmo gradiente produjo un 91,80% de efectividad de corrección hasta el valor de papr deseado, mientras que con el pulso gaussiano se obtuvo un 94,43% , lo que permite notar un mejor resultado de este ultimo con 2,63% superior que el otro, pero quizás tampoco un valor decisivo al respecto.

A continuación se siguen usando los resultados obtenidos (datos de Excel del apéndice F) que fueron ordenados en la **tabla II** y aprovechando dichos resultados se usaran para en la **tabla III** visualizar otros datos de comparación de ambos algoritmos.

Como se expreso antes la tecnica1 es tonos reservados con algoritmo gradiente y la 2 tonos reservados con pulso gaussiano.

simbolo enviado	numero de iteraciones en hacer la correccion 1	numero de iteraciones en hacer la correccion 2	porcentaje de acercamiento de la correccion1 a lo deseado	porcentaje de acercamiento de la correccion2 a lo deseado	coreccion1 mejor que 2	coreccion2 mejor que 1	Correccion 2 = 1	numero de iteraciones 1 menor que el de tecnica2	numero de iteraciones 2 menor que el de tecnica1	numero de iteraciones 1 igual al de tecnica 2
ii	Numiteraciones1'	Numiteraciones2'	porcentaje aproximacion con tecnica1	porcentaje aproximacion con tecnica2	coreccion1 mejor que 2	coreccion2 mejor que 1	Correccion 2 = 1	iteraciones1 menor que2	iteraciones 2 menor que 1	iteraciones 1 igual a 2
98	22	2	100,00%	100,00%	0	0	1	0	1	0
175	28	4	100,00%	100,00%	0	0	1	0	1	0
131	30	24	100,00%	100,00%	0	0	1	0	1	0
190	30	30	100,00%	100,00%	0	0	1	0	0	1
32	30	31	81,18%	83,67%	0	1	0	1	0	0
110	30	30	100,00%	100,00%	0	0	1	0	0	1
9	30	24	100,00%	100,00%	0	0	1	0	1	0
171	30	15	100,00%	100,00%	0	0	1	0	1	0
86	30	17	100,00%	100,00%	0	0	1	0	1	0
53	30	18	100,00%	100,00%	0	0	1	0	1	0
135	30	28	100,00%	100,00%	0	0	1	0	1	0
16	2	3	90,72%	100,00%	0	1	0	1	0	0
178	30	17	100,00%	100,00%	0	0	1	0	1	0
145	30	15	100,00%	100,00%	0	0	1	0	1	0
143	30	2	100,00%	100,00%	0	0	1	0	1	0
105	30	15	100,00%	100,00%	0	0	1	0	1	0
204	32	32	33,14%	17,12%	1	0	0	0	0	1
70	2	2	39,31%	100,00%	0	1	0	0	0	1
64	32	4	100,00%	100,00%	0	0	1	0	1	0
22	30	2	100,00%	100,00%	0	0	1	0	1	0
189	22	2	100,00%	100,00%	0	0	1	0	1	0
52	2	2	84,82%	100,00%	0	1	0	0	0	1
31	30	2	100,00%	100,00%	0	0	1	0	1	0
159	30	30	100,00%	100,00%	0	0	1	0	0	1
166	30	30	100,00%	100,00%	0	0	1	0	0	1
146	30	30	99,56%	100,00%	0	1	0	0	0	1
38	30	26	100,00%	100,00%	0	0	1	0	1	0
185	30	30	99,89%	100,00%	0	1	0	0	0	1
183	22	3	100,00%	100,00%	0	0	1	0	1	0

TABLA III: reorganización de resultados de simulación

56	30	30	92,88%	95,89%	0	1	0	0	0	1
149	30	16	100,00%	100,00%	0	0	1	0	1	0
174	30	2	100,00%	100,00%	0	0	1	0	1	0
20	30	30	86,63%	97,11%	0	1	0	0	0	1
176	30	19	100,00%	100,00%	0	0	1	0	1	0
120	2	2	89,20%	100,00%	0	1	0	0	0	1
103	2	2	95,02%	100,00%	0	1	0	0	0	1
201	2	2	97,24%	100,00%	0	1	0	0	0	1
117	32	33	81,44%	75,40%	1	0	0	1	0	0
197	32	5	99,66%	100,00%	0	1	0	0	1	0
181	17	1	100,00%	100,00%	0	0	1	0	1	0
107	14	2	100,00%	100,00%	0	0	1	0	1	0
203	32	32	99,95%	100,00%	0	1	0	0	0	1
18	30	17	100,00%	100,00%	0	0	1	0	1	0
206	30	31	100,00%	100,00%	0	0	1	1	0	0
126	30	30	99,94%	100,00%	0	1	0	0	0	1
77	32	30	85,33%	94,97%	0	1	0	0	1	0
42	32	32	96,76%	99,70%	0	1	0	0	0	1
82	30	16	100,00%	100,00%	0	0	1	0	1	0
75	30	2	100,00%	100,00%	0	0	1	0	1	0
147	32	32	9,10%	14,24%	0	1	0	0	0	1
15	30	13	100,00%	100,00%	0	0	1	0	1	0
122	34	33	91,62%	99,09%	0	1	0	0	1	0
29	30	2	100,00%	100,00%	0	0	1	0	1	0
115	30	15	100,00%	100,00%	0	0	1	0	1	0
44	30	32	89,72%	88,06%	1	0	0	1	0	0
6	2	2	62,95%	100,00%	0	1	0	0	0	1
139	2	2	68,12%	100,00%	0	1	0	0	0	1
50	30	21	100,00%	100,00%	0	0	1	0	1	0
193	30	30	89,37%	88,18%	1	0	0	0	0	1
140	30	3	100,00%	100,00%	0	0	1	0	1	0
168	30	30	100,00%	100,00%	0	0	1	0	0	1
7	32	30	100,00%	100,00%	0	0	1	0	1	0
13	17	1	100,00%	100,00%	0	0	1	0	1	0
35	17	1	100,00%	100,00%	0	0	1	0	1	0
59	31	14	98,95%	100,00%	0	1	0	0	1	0
68	31	1	100,00%	100,00%	0	0	1	0	1	0
80	17	1	100,00%	100,00%	0	0	1	0	1	0
99	17	1	100,00%	100,00%	0	0	1	0	1	0
104	17	1	100,00%	100,00%	0	0	1	0	1	0
114	17	1	100,00%	100,00%	0	0	1	0	1	0
123	30	32	57,73%	41,20%	1	0	0	1	0	0
130	17	1	100,00%	100,00%	0	0	1	0	1	0
51	30	2	100,00%	100,00%	0	0	1	0	1	0

Continuación de Tabla III

137	2	3	55,58%	100,00%
3	30	23	100,00%	100,00%
136	30	30	95,93%	99,58%
37	30	31	100,00%	100,00%
74	30	31	53,50%	64,85%
167	30	31	100,00%	99,93%
55	30	17	100,00%	100,00%
134	30	3	100,00%	100,00%
102	32	30	90,72%	84,97%
157	2	2	93,02%	100,00%
209	18	1	100,00%	100,00%
95	32	31	70,90%	63,93%
161	2	3	93,22%	100,00%
192	30	16	100,00%	100,00%
138	30	31	52,85%	52,56%
118	30	21	100,00%	100,00%
170	30	15	100,00%	100,00%
158	30	30	99,92%	96,37%
4	18	1	100,00%	100,00%
57	3	2	78,08%	100,00%
90	18	1	100,00%	100,00%
111	18	1	100,00%	100,00%
60	30	32	99,99%	99,76%
93	32	30	98,60%	98,09%
188	30	18	100,00%	100,00%
49	30	31	99,90%	100,00%
164	30	32	100,00%	100,00%
67	32	30	44,87%	45,06%
76	20	29	100,00%	100,00%
69	30	34	80,66%	71,02%
113	32	35	19,07%	50,14%

0	1	0
0	0	1
0	1	0
0	0	1
0	1	0
1	0	0
0	0	1
0	0	1
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0
0	0	1
1	0	0
0	0	1
0	0	1
1	0	0
0	0	1
0	1	0
0	0	1
0	0	1
1	0	0
1	0	0
0	0	1
0	1	0
0	0	1
0	0	1
1	0	0
0	0	1
0	1	0
0	0	1
0	1	0
1	0	0
0	1	0

1	0	0
0	1	0
0	0	1
1	0	0
1	0	0
1	0	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	1
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	0	1
0	1	0
0	1	0
0	1	0
0	1	0
0	0	1
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
0	1	0
1	0	0
1	0	0
1	0	0

veces en que tecnica 1 se acerca o mas que 2	veces en que tecnica 2 se acerca o mas que 1	veces en que tecnica 1 es igual que 2
13%	29%	59%

veces en que tecnica 1 es mas rapida que 2	veces en que tecnica 2 es mas rapida que 1	veces en que tecnica 1 es igual que 2
17%	59%	24%

Continuación de Tabla III

Se han obtenido nuevos valores de comparación, que mostrándose gráficamente serían:

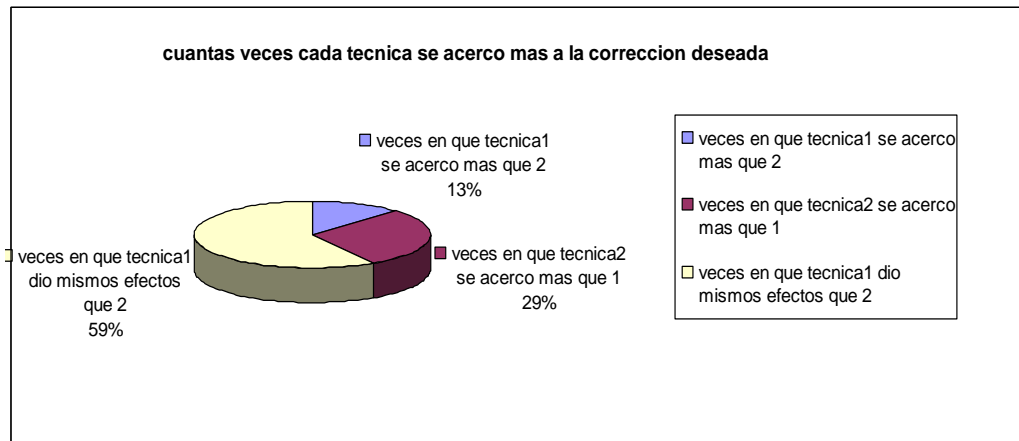


Fig. 5.4.- Cuantas veces cada técnica se acercó mas a la corrección deseada

Y se nota que en mas del 50% de ocasiones ambas técnicas se acercan por igual a la corrección deseada, pero sin embargo del total de veces es la técnica 2 (pulso gaussiano) es la que se acerca en mas ocasiones a la corrección deseada en un 16% mas sobre las veces que la supero tonos reservados..

Pero quizás el valor en el que más se nota diferencia es en el número de iteraciones que realizo cada uno.

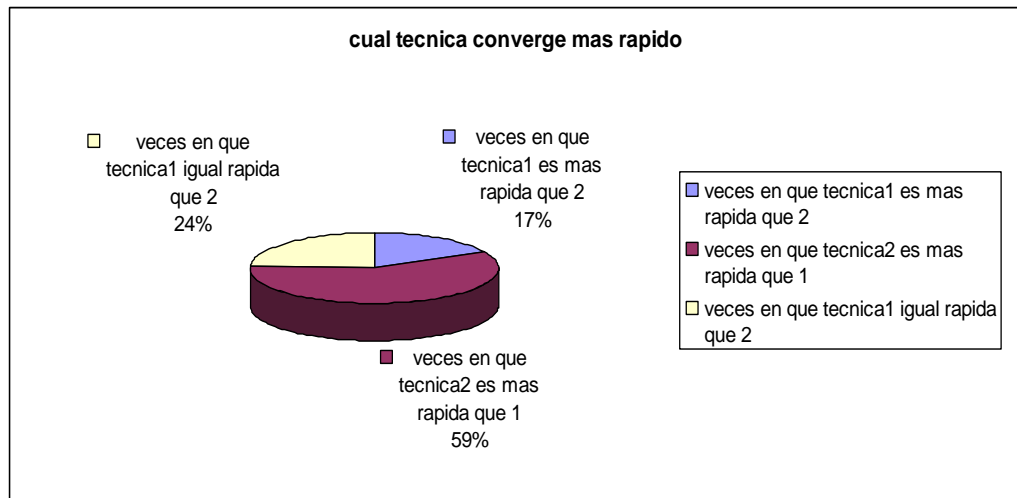


Fig. 5.5.- veces en que cada técnica converge mas rápido

Se nota que en mas del 50% de todas las correcciones es el pulso gaussiano el que se acerca a la corrección mas rápidamente, siendo una notable diferencia para su consideración, en la mayoría de casos el algoritmo de pulso gaussiano converge mas rápido que algoritmo gradiente (con los valores de Niu , Umbral y numero de tonos reservados que se trabajaron).

Para finalizar puede indicarse que la técnica de tone reservation , como se mostró, es una técnica que a pesar de traer la desventaja de utilizar tonos que ya no servirán para enviar información (causando disminución en el numero de bits que se transmite por símbolo, por ejemplo en nuestro caso del 7,69%) , se compensa por su buen porcentaje de corrección de mas del 90% en promedio de cercanía al

PAPR deseado teniendo la ventaja de no tener que hacer cambios drásticos en el receptor (simplemente no tomar en cuenta los tonos reservados) , y aunque introduce complejidad esta puede manejarse escogiendo un algoritmo apropiado u optimizando el algoritmo que se use u otros parámetros, por ejemplo en nuestro caso se mostró que en este sistema (con los parámetros escogidos) el algoritmo de pulso gaussiano en bastantes casos produce pocas iteraciones, lo cual da mas dinamismo y rapidez al proceso de corrección. Tone Reservation es una técnica con bastantes posibilidades de cambios y pruebas y buenos resultados lo que la ha llevado a ser bastante considerada en los estudios de disminución de PAPR.

CONCLUSIONES

1. Los valores de peak to average power ratio excesivamente altos pueden ocasionar problemas de eficiencia en el hardware de convertidores A/D D/A y HPA's debido al amplio rango que deben manejar para valores ocasionales (picos), por lo que se necesita de una técnica que ayude a controlar dichos niveles de PAPR.
2. Para realizar la disminución de PAPR, se ha usado la técnica de tonos reservados, pero al tener varias formas de implementarla se decidió hacerlo a través de 2 algoritmos, uno es el algoritmo gradiente y el otro a través de pulso gaussiano, para observar sus efectos reductores y comparación de ambas.
3. Se creo un código de matlab para simulación pero se encontró la necesidad de ajustar parámetros para realizar una adecuada obtención

de datos y analizar los resultados, es por eso que se ajusta el código para visualizar las reacciones al variar valores como: el nivel de umbral de saturación, el número de tonos reservados y el tamaño de paso en el algoritmo gradiente.

4. Se pudo encontrar que el nivel de ajuste del PAPR a su reajuste deseado era superior al 90% con este método implementado con ambos algoritmos, lo que hace la técnica bastante viable y nos da la ventaja de no necesitar ajustes extra en el receptor.
5. Al comparar ambos algoritmos se observa que en el 50% de las correcciones el algoritmo de pulso gaussiano logra converger más rápidamente que la de algoritmo gradiente.

RECOMENDACIONES

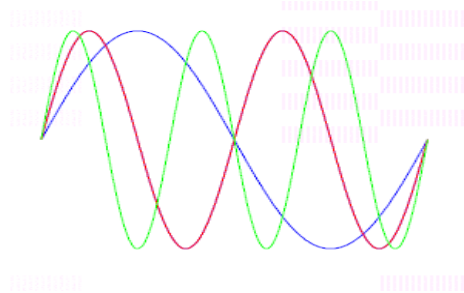
1. En el algoritmo pueden ajustarse distintas variables según se desee observar los efectos, estos son numero de bits a transmitir, valor de umbral, máximo numero de iteraciones, numero de tonos y tamaño de paso del algoritmo gradiente
2. No se ha llevado a cabo una búsqueda de optimización del tamaño de paso en algoritmo gradiente dado a la extensión en el estudio sobre el tema, personas adentradas en el mismo o con intención de observar mas a fondo el caso podrán reajustar dicho parámetro y realizar una nueva comparación.
3. El algoritmo también puede servir de base para comparar el uso de tonos reservados con otra técnica de corrección del PAPR.

APENDICES

APENDICE A

Transmisión OFDM/DMT

En contraposición a lo que vemos en las típicas comunicaciones monoportadora o *Single Carrier* (SC), donde cada símbolo se transmite serialmente (uno a la vez) ocupando todo el ancho de banda disponible, en una modulación multiportadora se envían los símbolos paralelamente en subportadoras adyacentes, es decir, usando algún tipo de multiplexaje por división de frecuencias (FDM)

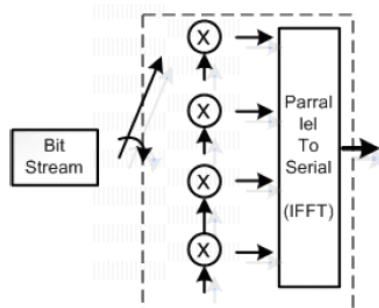


OFDM –Ortogonal Frequency-Division Multiplexing, aparece como una solución para no incrementar la frecuencia de la portadora, sino utilizar un conjunto de N subportadoras.

La frecuencia primaria se divide en cada subportadora que ahora transmite símbolos más largos a alta velocidad permitiendo aumentar el desempeño del sistema.

Siendo N subportadoras ortogonales las que se utilizan para un sistema OFDM cualquiera, éstas estarán separadas en frecuencia justamente por el valor correspondiente a la inversa del tiempo útil del símbolo OFDM o T_u , y durante este período se transmitirán N símbolos independientes codificados por QPSK, 16-QAM, 64-QAM o cualquier otro tipo de modulación I/Q.

En el grafico podemos apreciar que los símbolos que entran al sistema son convertidos en sub-tramas paralelas, este proceso de separación o división en sub-símbolos o bloques de información es llevada a cabo por un convertidor Serie-Paralelo.



Los símbolos a la entrada del convertidor Serie-Paralelo son recibidos a una tasa de datos específica, estos símbolos son divididos en subcadenas o tramas de símbolos cada una de ellas con una menor tasa de datos al

final todos estos símbolos de menor tasa de datos se suman para poder obtener una tasa de datos equivalente a la original

Tomaremos como ejemplo que deseamos transmitir la siguiente cadena de bits:

1, 1, -1, -1, 1, 1, 1, -1, 1, -1, -1, -1, -1, 1, -1, -1, -1, 1,

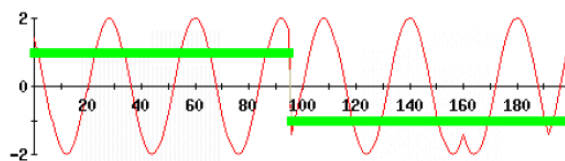
Esta secuencia la deseamos transmitir en OFDM usando 4 subportadoras. La señal tiene una tasa de símbolos de 1 y frecuencia de muestreo de 1 muestra por símbolo, entonces cada transmisión es de 1 bit.

A continuación los datos son convertidos de serial a paralelo para asignarlo a cada subportadora

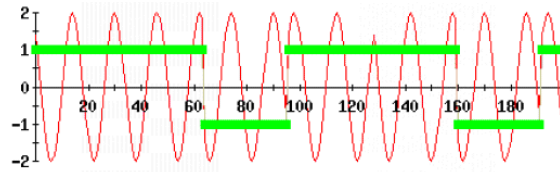
c1	c2	c3	c4
1	1	-1	-1
1	1	1	-1
1	-1	-1	-1
-1	1	-1	-1
-1	1	1	-1
-1	-1	1	1

Cada columna representa los bits que serán asignados a cada subportadora y cada fila es un símbolo que se enviara.

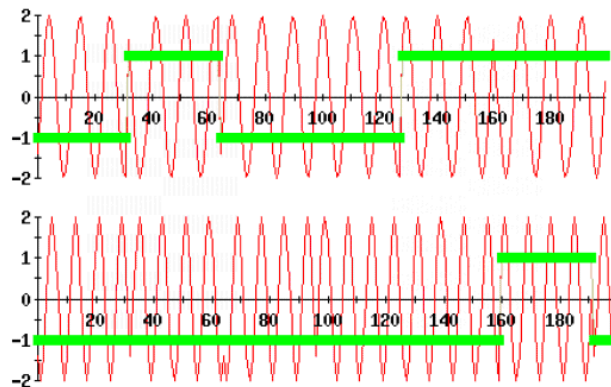
Transmitiendo la primera columna que es para la subportadora 1, utilizando BPSK con una frecuencia de portadora de 1Hz (por conveniencia), obtenemos:



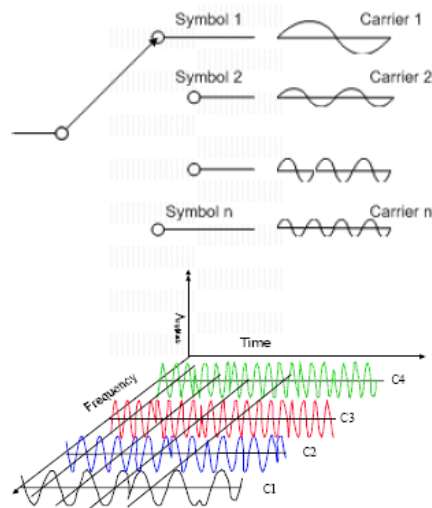
La siguiente columna c2 que es para subportadora 2 sera para una frecuencia de 2Hz que es la siguiente orthogonal/harmonic a la frecuencia de 1Hz de la primera subportadora, con los datos de la segunda columna quedaría:



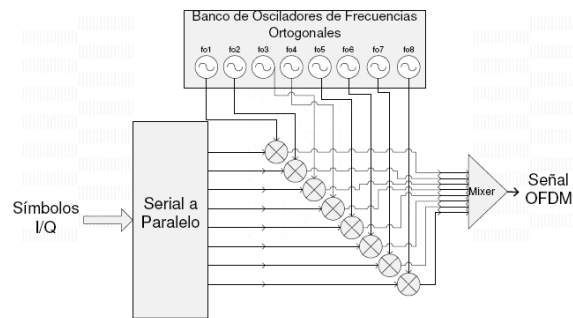
Así mismo los datos de la columna 3 y 4 son modulados para las siguientes subportadoras 3 y 4 a frecuencias de 3Hz y 4 Hz



Así se ha distribuido la cadena de bits para cuatro subportadoras, la siguiente grafica nos muestra este proceso de haber distribuido la cadena de datos para las 4 subportadoras en las que la transmitiríamos.

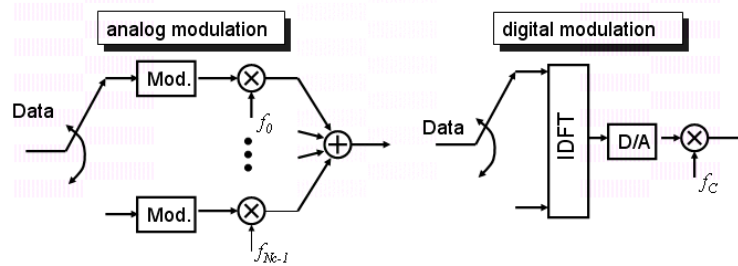


A continuación se observa un esquema de lo que sería una implementación de OFDM con apenas 8 frecuencias ortogonales, realizada totalmente en la etapa de radiofrecuencia (RF).



Se observa la dependencia del sistema de la precisión de cada oscilador y demás elementos que pueden introducir espurias como mixers y divisores, y debemos contar con las no linealidades de filtros y amplificadores que suponen una demodulación coherente en el receptor. Si tomamos en cuenta que normalmente se requerirían muchas más subportadores, es fácil presumir la complejidad y costos prohibitivos en que incurriría este tipo de dispositivo.

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 desmodulación se realizan en tiempo discreto mediante la IDFT y la DFT respectivamente.



Para comprender su utilización deberemos partir del concepto mismo de la Transformada de Fourier y su inversa, así conocidos, teniendo establecida cualquier señal continua $x(t)$ en el dominio del tiempo, la transformada de Fourier $F\{x(t)\}$ esta dada por

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

Siempre recordando la relación entre la frecuencia en radianes con la frecuencia en hertzios $\omega=2\pi f$

En un sistema digital, como lo sería nuestro esquema OFDM, el paso al mundo analógico se da mediante convertidores analógico a digital (DACs) y digital a analógico (ADCs) ubicados en la misma frontera del sistema que maneja la modulación en banda base

En vista de esto, las señales que manejamos en realidad están muestreadas, y son así mismo, señales discretas.

Por otro lado, incluso para los análisis realizados con la ayuda de una computadora, como al utilizar MATLAB, jamás tendremos realmente señales continuas, sino muestras de ellas. Y para realizar el análisis necesario en frecuencia, y luego entender los métodos numéricos que se deben usar

comenzaremos por contemplar la utilización de la transformada de Fourier de una señal muestreada, de este modo, siendo la señal continua en el tiempo $x(t)$, siendo n el índice de las muestras de dicha señal, $x(nT_s)$ es para cada valor de n , una de las muestras de la señal $x(t)$ y T_s es el periodo de muestreo.

la versión computable de Fourier es la denominada Transformada Discreta de Fourier o DFT. Ésta es un algoritmo donde se usa un número limitado de muestras en el tiempo, y como consecuencia obtendremos así mismo la misma cantidad de muestras en el dominio de la frecuencia. Se define $\hat{x}(n)$ como la versión truncada de $x(nT_s)$, $\hat{x}(n) = x(nT_s)$ para un número limitado de muestras N_s . La DFT para N_s muestras esta definida por:

$$\hat{X}(k) = \sum_{n=0}^{N_s-1} \hat{x}(n) e^{-j \frac{2\pi nk}{N_s}}$$

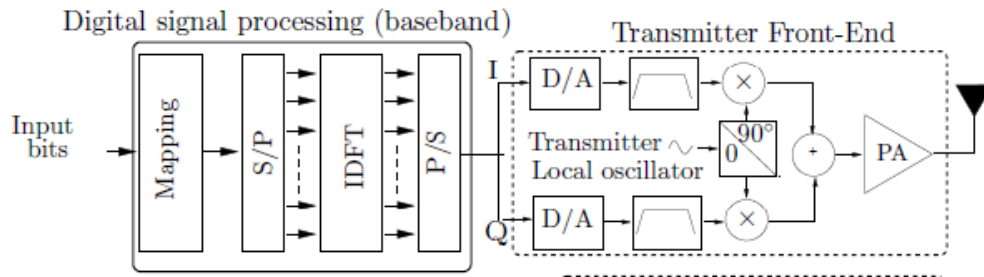
la relación inversa está dada básicamente por la misma relación, en vista que tan solo tendríamos que cambiar los índices, sin embargo se considera incluir un factor de compensación a la sumatoria. Entonces la IDFT se define:

$$\hat{x}(n) = \frac{1}{N_s} \sum_{k=0}^{N_s-1} \hat{X}(k) e^{j \frac{2\pi nk}{N_s}} \quad n = 0, 1, 2, \dots, N_s - 1$$

El cálculo de la DFT o su inversa implica para su cálculo la realización de sucesivas multiplicaciones y sumas, además de los gastos de memoria que crecen de acuerdo al valor de muestras. Con el objeto básico de reducir dicho número de multiplicaciones, fue diseñada la denominada

“Transformada Rápida de Fourier” o FFT, en el cual el número de multiplicaciones se reduce.

A continuación se muestra el esquema de un transmisor OFDM



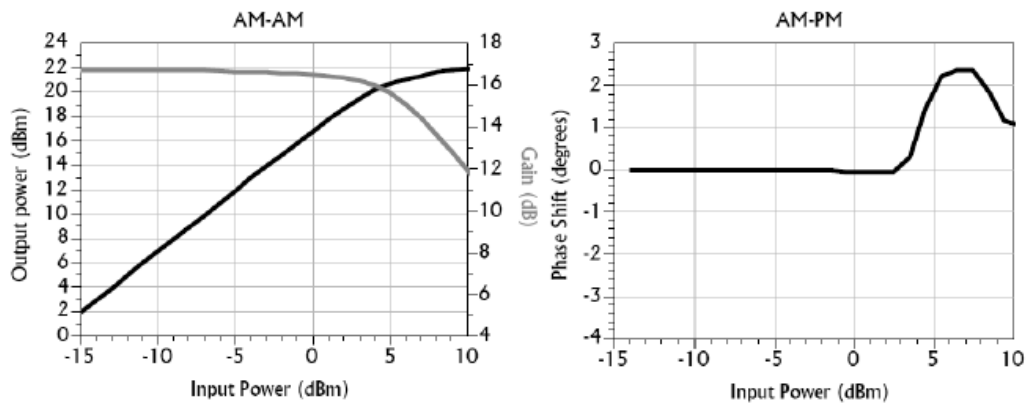
La entrada de bits es mapeada a símbolos apropiados dependiendo de la modulación usada en las subportadoras, luego se procede a hacer la transformada discreta de Fourier inversa (IDFT), donde la salida de la misma es paralela y debe ser convertida a cadenas seriales de símbolos OFDM y se separa en sus componentes real e imaginario (en fase y cuadratura) y se utiliza un convertidor D/A para pasar la señal digital a una analógica de banda base que es filtrada y finalmente será trasladada a la frecuencia de portadora y amplificada.

APENDICE B

Modelo de distorsión no lineal de los amplificadores de potencia

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.

En la Figura siguiente se observa la gráfica AM-AM que muestra relación entre la potencia de entrada y la potencia de salida del amplificador. La gráfica AM-PM muestra la distorsión de fase que introduce el AP dependiendo de la potencia de entrada. Existe un límite en la potencia de entrada (aproximadamente 3dBm en el AP de la Figura) donde comienza a saturar y a introducir distorsión de fase y es por lo tanto hasta donde el AP trabaja en zona lineal.



Curvas características AM-AM y AM-PM

Si consideramos un modelo genérico de una señal banda base $x(t)$ y, simplificando la notación, dejamos implícita la dependencia temporal, la entrada al HPA estaría dada por (en coordenadas polares):

$$x = |x|e^{j\alpha} = ue^{j\alpha}$$

Donde:

$$u = |x|$$

α la componente de fase instantánea de la señal x

De este modo tenemos la salida distorsionada:

$$g(x) = A[u]e^{j(\alpha+\phi[u])}$$

Donde

$A[u]$ representa la distorsión de amplitud de la señal de entrada AM/AM

$\phi[u]$ representa la distorsión de la fase de la señal de entrada AM/PM

Estas dos características dependen particularmente del tipo de HPA, como por ejemplo están los TWTA que son los amplificadores de potencia que

tienen como base traveling-wave tube amplifiers (por eso sus siglas) y los SSPA son los solid state power amplifier, las características de ambos son diferentes por ejemplo la distorsión en fase de los SSPA es mucho menor que la de los TWTA. Para los SSPA los valores de distorsión de magnitud y fase se aproximados se expresan como:

$$A[x] = \frac{|x(n)|}{\left[1 + \left(\frac{|x(n)|}{A_s}\right)^{2p}\right]^{1/2p}}$$

$$\phi[x] = 0$$

Donde p es un valor que se relaciona con la suavidad de la transición desde la operación lineal del amplificador hasta su zona de operación en saturación, y A_s es la amplitud de la salida en saturación.

Cuando el valor p toma valores largos (mayor suavidad al pasar de zona lineal a saturación) este modelo de la distorsión en magnitud $A[x]$ se aproxima al modelo llamado soft limiter model (el modelo de limitado suave) que es el mas usado comúnmente para la representación de la operación de clipping de una señal, así, si en la ecuación anterior de la distorsión de magnitud hacemos que el valor de p tienda a valores grandes (infinito) podemos obtener:

$$A[x] = \begin{cases} |x|, & |x| \leq A_s \\ A_s, & |x| > A_s \end{cases}$$

$$\phi[x] = 0$$

Como la distorsión de fase cero al remplazar en la expresión de $g(x)$ podemos obtener

$$g(x) = A[|x|]e^{j(\alpha+\phi[|x|])}$$

$$g(x) = A[|x|]e^{j(\alpha)}$$

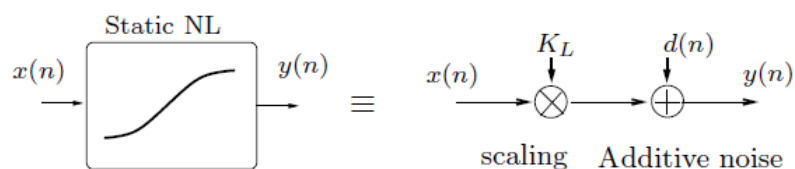
$$g(x) = \begin{cases} |x|, & |x| \leq A \\ A_s e^{j\alpha}, & |x| > A \end{cases}$$

Donde , como se describió anteriormente, A_s es el valor del amplificador en saturación y α es la componente de fase de x cuando se lo expreso en forma polar, es decir el argumento de x .

Existe también un modelo de la salida no lineal del amplificador que se suele utilizar en casos discretos, cuando la señal $x(n)$ que excita al PA tiene una distribución gaussiana, esta representación de la salida no lineal se da como:

$$y_{NL}(n) = K_L x(n) + d(n)$$

Donde el termino de distorsión $d(n)$ no esta relacionado con $x(n)$ y K_L es un factor de atenuación, la siguiente grafica muestra una ilustración en diagramas de bloques.



APENDICE C

Estándar 802.11a

Aquí se revisarán algunas de las características que se usan para el estándar 802.11a específicamente a lo correspondiente al símbolo OFDM sin profundizar en detalles del estándar completo.

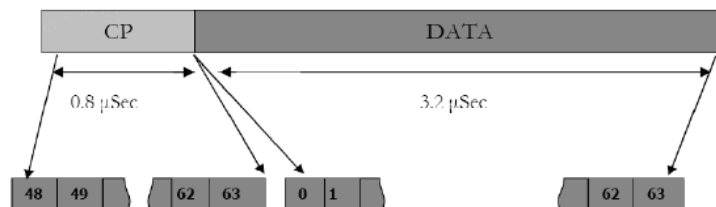
Las especificaciones del ensamblaje del símbolo OFDM entre otras:

- FFT e IFFT de 64 puntos.
- Los primeros 6 y los últimos 5 símbolos en el dominio de la frecuencia no son usados y se “rellenan” con cero.
- Índice de frecuencia 0 se usará con cero también.
- Los últimos 16 símbolos se usarán como Prefijo Cíclico.
- El uso de los símbolos que se marcan como “cero” en el dominio de la frecuencia se utilizan para minimizar la necesidad del “ventaneo” necesario para evitar la interferencia entre canales, así mismo eliminando el uso del índice de frecuencia cero nos ayuda a evitar el uso de la componente DC en la operación de muestreo.

De los 64 símbolos que estamos procesando luego de las consideraciones arriba expuestas tendremos las frecuencias de la -26 a la -1 y de 1 a 26, con un total de 52 componentes espectrales se podrán usar para transmisión.

Adicionalmente el estándar 802.11a contempla el uso de las componentes en frecuencia con índice -21, -7, 7 y 21 como símbolos pilotos para estimación del canal (4), las consideraciones de las componentes espectrales que se han detallado se muestran en la figura siguiente.

En la Figura A se muestra la inserción del prefijo cíclico en el símbolo OFDM y en la Figura B se muestra el símbolo OFDM en el dominio del estándar 802.11a al finalizar la construcción del mismo, resaltando las componentes del prefijo cíclico y la duración en el dominio del tiempo.



Símbolo OFDM 802.11a en el dominio del tiempo

APENDICE D

Transformada de Fourier de pulso Gaussiano de la forma

$$f(x) = \exp\left(-\frac{ax^2}{2}\right) \quad \text{Siempre que } a > 0$$

Para hallar la transformada de Fourier de esta expresión recurriremos al uso de 2 propiedades de la transformada de Fourier que son:

1.- transformada de la derivada: $\longrightarrow F(f'(x)) = ikF(f(x)) = ikF(k)$

2.- transformada de $x \cdot f(x)$: $\longrightarrow F(xf(x)) = iF'(f(x)) = iF'(k)$

Comencemos derivando la expresión:

$$f(x) = \exp\left(-\frac{ax^2}{2}\right) \Rightarrow f'(x) = -ax \exp\left(-\frac{ax^2}{2}\right) = -axf(x)$$

Utilizando las 2 primeras propiedades:

$$\left. \begin{array}{l} F(f'(x)) = ikF(f(x)) = ikF(k) \\ F[xf(x)] = iF'(f(x)) = iF'(k) \end{array} \right\} \Rightarrow ikF(k) = -iaF'(k)$$

$$ikF(k) = -iaF'(k) \Rightarrow F(k) = Be^{-\frac{k^2}{2a}}$$

$$F(k) = Be^{-\frac{k^2}{2a}} = \int_{-\infty}^{\infty} e^{-\frac{ax^2}{2}} e^{-ikx} dx$$

$$\begin{aligned} B = F(0) &= \int_{-\infty}^{\infty} e^{-\frac{ax^2}{2}} dx = \sqrt{\frac{2}{a}} \int_{-\infty}^{\infty} e^{-u^2} du = 2\sqrt{\frac{2}{a}} \int_0^{\infty} e^{-u^2} du \\ &= \sqrt{\frac{2}{a}} \int_0^{\infty} \frac{e^{-t}}{\sqrt{t}} dt = \sqrt{\frac{2\pi}{a}} \end{aligned}$$

$$F(k) = \sqrt{\frac{2\pi}{a}} e^{-\frac{k^2}{2a}}$$

$$f(x) = \exp\left(-\frac{ax^2}{2}\right) \xrightarrow{F} F(k) = \sqrt{\frac{2\pi}{a}} e^{-\frac{k^2}{2a}}$$

Si tomamos $a/2$ como una constante M podemos expresar:

$$f(x) = \exp(-Mx^2) \xrightarrow{F} F(k) = \sqrt{\frac{\pi}{M}} e^{-\frac{k^2}{4M}}$$

APENDICE E

Códigos matlab para las simulaciones hechas

Código de matlab utilizado para simular corrección de PAPR con tonos reservados (usando algoritmo gradiente y usando pulso gaussiano)

```
1 % Author : Juan Carlos Pinto y Miguel Cevallos, (tomado como punto de partida algoritmo de graficacion de Krishna)
2 % Version : 1.0
3 % Date :
4 % *****
5
6 % Script for computing the per symbol peak to average PAPR for
7 % an OFDM transmit waveform (loosely based on IEEE 802.11A
8 % specifications)
9 % PAPR reduction with tone reservation ( algoritmo gradiente y pulso
10 % gaussiano)
11
12 clear
13 nFFTSsize = 64; % tamaño de la transformada inversa de fourier , para este caso lo haremos coincidir con el numero
14 % de subportadoras
15 % for each symbol bits a1 to a52 are assigned to subcarrier
16 nBit = 10000; % numero de bits a transmitir
17 ip = rand(1,nBit) > 0.5; % generating 1's and 0's
18 % ip = zeros(1,nBit)+1 ; %este es un caso maximo para que todas las portadoras estan en fase
19 nportadoras = 64; % numero de subportadoras
20 nToneRes = 4; % numero de tonos reservados
21 modulacion = 1; % tipo de modulacion 1=BPSK
22 UMBRAL = 14 ; % maximo valor que se va a permitir a la entrada del HPA
23 maxit = 30 ; % maximo numero de iteraciones del algoritmo gradiente
24 Niu = 14 ; % tamaño de paso del algoritmo gradiente
25 picoExistentes = 0 ; % indica cuantos picos van apareciendo en un mismo simbolo OFDM
26 numiterac=0 ; % numero de iteraciones en las que se hizo la correccion del papr en un simbolo
27 picoExistentes2 = 0 ; % indica cuantos picos van apareciendo en un mismo simbolo OFDM
28 numiterac2=0 ; % numero de iteraciones en las que se hizo la correccion del papr en un simbolo
29 segraficoofdm=0 ; %indica si se grafico un simbolo con la correccion del papr
30 segraficoofdm2=0 ; %indica si se grafico un simbolo con la correccion del papr
31 ubicacionp=0 ; %muestra donde esta un pico
32
33
```

```

34 % matriz Q
35 N = nFFTSize ; % tamaño de la matriz IFFT
36 n= 1 ; % subindice de las filas
37 m = 1 ; % subindice de las columnas
38
39 for n = 1:N
40 for m = 1:N
41
42 Q(n,m) = exp(0+i*(2*pi/N)*(n-1)*(m-1)) ;
43
44 end
45 end
46 Q= (1/(N^(0.5)))*Q ;
47
48
49 % a continuación se mapea para bpsk
50
51 nBitPerSymbol = nportadoras-12- nToneRes; % numero de Portadoras que llevaran información (nportadoras totales
52 % menos 12 que se llenan con cero en el standard 802.11a menos las que se usan como tonos reservados)
53 nSymbol = ceil(nBit/nBitPerSymbol); % se reparten los 10000 datos entre las
54 % portadoras que llevan información
55
56 % BPSK modulation
57 % bit0 --> -1
58 % bit1 --> +1
59 ipMod = 2*ip - 1; % los 0 y 1 se convierten en -1 y +1
60 ipMod = [ipMod zeros(1,nBitPerSymbol*nSymbol-nBit)]; % se completa con ceros al final
61 % para formar un numero Redondo para las portadoras
62 ipMod = reshape(ipMod,nSymbol,nBitPerSymbol); % divide los datos en los símbolos que se enviaran
63
64
65 % modulando en OFDM
66 % st = []; % empty vector, aquí se iran guardando los resultados del papr de cada simbolo ofdm
67
68 index = [-32:31] ; % sin contar prefijo ciclico
69 % index [-26 to -1 1 to 26] , indices que se manejan segun standard 802.11a
70 subcarrierIndex = [-26:-1 1:7 (8+nToneRes):26]; %indice de las portadoras que llevan informacion
71 % en 802.11a se usan el 7 y el 21 como pilotos asi que escogeremos tonos
72 % reservados a partir del 8 y en los tonos reservados no se envia informacion
73 tonereservIndex = [8:nToneRes-1] ; % como la 7 es piloto cogemos tonos reserv a partir de la 8
74
75 Qred = Q(:,tonereservIndex+33) ; %matriz reducida de Q para los tonos reservados que usaremos
76
77
78 % todo el siguiente proceso se realiza para cada simbolo OFDM
79
80 for ii = 1:nSymbol
81
82
83 Xf = zeros(nFFTSize,1);
84 % asignando bits a1 1 to a52 a los subcarriers que llevaran informacion
85 Xf(subcarrierIndex+nFFTSize/2+1) = ipMod(ii,:); %vector X a la entrada de la ifft
86 xt = 64*ifft(Xf,nFFTSize); %salida d la transformada inversa de fourier
87
88
89 % computing the peak to average power ratio for each symbol
90 meanSquareValue = xt'*xt/length(xt);
91 peakValue = max(xt.*conj(xt)) ;
92 paprSymbol(ii) = peakValue/meanSquareValue;
93 paprSymboldB(ii) = 10*log10(paprSymbol(ii));
94
95 EXCESO(ii) = (paprSymboldB(ii) - 10*log10((UMBRAL^2)/meanSquareValue)) ;
96 if EXCESO(ii)>0
97 EXCESO(ii) = EXCESO(ii) ;
98 else
99 EXCESO(ii) = 0 ;

```

```

100 - end
101
102
103
104
105 % correccion del PAPR algoritmo gradiente
106
107 - c = zeros(nFFTSize,1); % c es la señal correctora que va en los tonos reservados
108 - numiterac = 0 ; % contara el num de iteraciones del algoritmo (cuantas veces revisa el mismo simbolo y corrige)
109 - numiteracFin = 0 ; % guarda el numero total de iteraciones que le hizo a un simbolo ofdm
110 - xfinal = xt + c ; % nuestra x resultante es la suma de la x original con los tonos reservados
111
112
113 - while numiterac < maxit ; %mientras no se pase del numero maximo de iteraciones
114
115 -     kk = 0 ; % contador de las muestras de la matriz x del simbolo actual analizandose
116 -     picoExistentes = 0 ;
117 -     cs = zeros(nFFTSize,1) ; % señal correctora para la respectiva muestra
118
119
120 -     for kk = 1 : nFFTSize ;
121 -         if abs(xfinal(kk))>UMBRAL ;
122 -             picoExistentes= picoExistentes+1 ;
123 -             qrow = Qred(kk,:) ;
124 -             cs = cs + ((xfinal(kk))-(UMBRAL*exp(i*angle(xfinal(kk)))))*Qred*qrow') ; %correccion con algoritmo gradiente
125 -             numiterac=numiterac+1 ;
126 -         end
127 -     end
128
129
130 -     if picoExistentes > 0 ; %si se encontraron picos se hizo correccion en ese simbolo , se llevaria 1 iteracion mas
131 -         numiteracFin = numiterac ; % el numero de iteraciones en que se realizo la reduccion total
132 -         c = c - ( Niu * cs ) ; %señal correctora total para el respectivo simbolo OFDM
133 -         xfinal = xt + c ;
134
135
136 -     else
137 -         numiterac = maxit ; % como ya no hay picos en el smbolo picoExistentes=0 directamente le da el valor para que
138 -             % salga del lazo
139
140 -     end
141
142
143 - end
144
145
146 - if (segraficoofdm <= 0 ) % para realizar una sola grafica , la del primer simbolo OFDM donde se hace correccion
147 -     if (numiteracFin > 0)
148 -         close all;
149 -         figure (1)
150 -         subplot(3,1,1), plot(1:64,abs(xt)) ;
151 -         ylabel('x')
152 -         ylim([0 26]) %se ha buscado una escala comoda para visualizar resultados
153 -         hold all ;
154 -         subplot(3,1,1), plot(1:64, UMBRAL ) ;
155 -         subplot(3,1,2), plot(1:64,abs(c)) ;
156 -         ylabel('c')
157 -         ylim([0 26])
158 -         subplot(3,1,3), plot(1:64,abs(xfinal)) ;
159 -         ylim([0 26])
160 -         hold all ;
161 -         subplot(3,1,3), plot(1:64, UMBRAL )
162 -         ylabel('x+c')
163 -         xlabel('numero de muestra discreta en el tiempo') % son 64 muestras por ser la ifft de 64
164 -         segraficoofdm=1 ;
165

```

```

166
167 - figure(2) %se graficara frecuencias
168 - subplot(3,1,1), plot(index,abs(fft(xt))) ;
169 - ylabel('x')
170 - hold all ;
171 - subplot(3,1,2), plot(index,abs(fft(c))) ;
172 - ylabel('c')
173 - subplot(3,1,3), plot(index,abs(fft(xfinal))) ;
174 - hold all ;
175 - ylabel('x+c')
176 - xlabel('subportadora')
177
178 - end
179 - end
180
181
182 % computing the peak to average power ratio ya aplicada la tecnica de crreccion for each symbol
183 - peakValue = max(xfinal.*conj(xfinal));
184 - paprMODSymbol(ii) = peakValue/meanSquareValue; % Papr con la correccion hecha
185 - paprMODSymboldB(ii) = 10*log10(paprMODSymbol(ii));
186 - Numiteraciones1(ii) = numiteracFin ;
187
188
189
190
191
192
193
194 % correccion por pulso Gaussiano
195
196 - c2 = zeros(nFFTSize,1); % c es la señal correctora que va en los tonos reservados
197 - numiterac2 = 0 ; % numero de iteraciones del algoritmo
198 - numiteracFin2 = 0 ; % guarda el numero total de iteraciones que le hizo a un simbolo ofdm
199 - xfinal2 = xt + c2 ; % nuestra x resultante es la suma de la x original con los tonos reservados
200
201 - while numiterac2 < maxit ; %mientras no se pase del numero maximo de iteraciones
202
203 - kk2 = 0 ; % contador de las muestras de la matriz x del simbolo actual analizandose
204 - picoExistentes2 = 0 ;
205
206 - for kk2 = 1 : nFFTSize ;
207
208 - if abs(xfinal2(kk2))>UMBRAL ;
209 - picoExistentes2= picoExistentes2+1 ;
210 - L = nToneRes ;
211 - alpha = 1;
212 - nm=1:64 ;
213 - m=(1:1:L);
214 - pgaussf=0.01*(exp (-0.5*( (alpha * ( m-(L/2)-1 ) / ((L-1)/2) ).^2 ))) ; %creacion de pulso gaussiano
215 - CF2=zeros(nFFTSize,1); % pulso corrector en el dominio de la frecuencia
216 - CF2(tonereservIndex+nFFTSize/2+1)=pgaussf ; %señal correctora c en el dominio de la frecuencia
217 - CF2=(exp(-i*(kk2/nFFTSize)*2*pi*nm')) .* CF2 ; % para ubicarlo en la muestra de tiempo que tiene el pico
218 - ct2=ifft(CF2); % paso a pulso en el tiempo
219 - ct2=((abs(xfinal2(kk2))-UMBRAL)/(abs(ct2(kk2))))*exp(i*(angle(xfinal2(kk2))-angle(ct2(kk2))+pi)) * ct2 ;
220 - % cuantificando a la cantidad que se necesita para disminuir el pico
221 - c2= c2 + ct2 ; %señal correctora por pulso gaussiano
222 - xfinal2 = xt + c2 ;
223 - numiterac2=numiterac2+1 ;
224 - end
225
226 - end
227
228
229 - if picoExistentes2 > 0 ; %si se encontraron picos se hizo correccion en ese simbolo , se llevaria 1 iteracion mas
230 - numiteracFin2 = numiterac2 ; % el numero de iteraciones en que se realizo la reduccion total
231

```

```

232 - else
233 -     numiterac2 = maxit ; % como ya no hay picos en el simbolo picoExistentes=0 para que salga del lazo
234 -
235 - end
236 -
237 -
238 -
239 - end
240 -
241 -
242 - if (segraficoofdm2 <= 0 ) % para realizar una sola grafica ,la del primer simbolo OFDM donde se hace correccion
243 -     if (numiteracFin2 > 0)
244 -         figure (3)
245 -         subplot(3,1,1), plot(1:64,abs(xt)) ;
246 -         ylabel('x')
247 -         ylim([0 26]) %se ha buscado una escala comoda para visualizar resultados
248 -         hold all ;
249 -         subplot(3,1,1), plot(1:64, UMBRAL ) ;
250 -         subplot(3,1,2), plot(1:64,abs(c2)) ;
251 -         ylabel('c')
252 -         ylim([0 26])
253 -         subplot(3,1,3), plot(1:64,abs(xfinal2)) ;
254 -         ylim([0 26])
255 -         hold all ;
256 -         subplot(3,1,3), plot(1:64, UMBRAL )
257 -         ylabel('x+c')
258 -         xlabel('numero de muestra discreta en el tiempo') % son 64 muestras por ser la ifft de 64
259 -         segraficoofdm2=1 ;
260 -
261 -
262 -         figure(4) %se graficara frecuencias
263 -         subplot(3,1,1), plot(index,abs(fft(xt))) ;
264 -         ylabel('x')
265 -         hold all ;
266 -         subplot(3,1,2), plot(index,abs(fft(c2))) ;
267 -         ylabel('c')
268 -         subplot(3,1,3), plot(index,abs(fft(xfinal2))) ;
269 -         hold all ;
270 -         ylabel('x+c')
271 -         xlabel('subportadora')
272 -
273 -     end
274 - end
275 -
276 -
277 - % computing the peak to average power ratio ya aplicada la tecnica de crreccion for each symbol
278 - peakValue = max(xfinal2.*conj(xfinal2));
279 - paprMOD2Symbol(ii) = peakValue/meanSquareValue; % Papr con la correccion hecha
280 - paprMOD2SymboldB(ii) = 10*log10(paprMOD2Symbol(ii));
281 - Numiteraciones2(ii) = numiteracFin2 ; % se guarda cuantas iteraciones se han hecho en el actual simbolo
282 -
283 -
284 -
285 - end
286 -
287 -
288 -
289 -
290 -
291 - % graficando distribucion del Papr :
292 -
293 - CORRECCION1 = paprSymboldB - paprMOD2SymboldB ;
294 - CORRECCION2 = paprSymboldB - paprMOD2SymboldB ;
295 -
296 -
297 - figure(5)

```

```

298
299 - [a1 x1] = hist(paprSymboldB,[0:0.5:15]);
300 - plot(x1,1-cumsum(a1)/nSymbol,'LineWidth',2, 'Color', 'r')
301 - xlabel('papr (dB)')
302 - ylabel('Probability, PPPR >= x')
303 - title('CCDF plots of PAPR from an IEEE 802.11a Tx with BPSK modulation')
304 - grid on
305
306 - hold all
307
308 - [a2 x2] = hist(paprMOD2SymboldB,[0:0.5:15]); %con la correccion se pondra en azul
309 - plot(x2,1-cumsum(a2)/nSymbol,'LineWidth',2, 'Color', 'b')
310
311
312
313 - figure(6)
314 - plot(x1,1-cumsum(a1)/nSymbol,'LineWidth',2, 'Color', 'r')
315 - xlabel('papr (dB)')
316 - ylabel('Probability, PPPR >= x')
317 - title('CCDF plots of PAPR from an IEEE 802.11a Tx with BPSK modulation')
318 - grid on
319
320 - hold all
321
322 - [a3 x3] = hist(paprMOD2SymboldB,[0:0.5:15]); %con la correccion se pondra en verde
323 - plot(x3,1-cumsum(a3)/nSymbol,'LineWidth',2, 'Color', 'g')
324
325
326
327
328
329

```



```
Q(n,m) = exp(0+i*(2*pi/N)*(n-1)*(m-1)) ;
```

```
end
```

```
end
```

```
Q= (1/(N^(0.5)))*Q ;
```

```
% a continuación se mapea para bpsk
```

```
nBitPerSymbol = nportadoras-12- nToneRes; % numero de Portadoras que llevaran  
información (nportadoras totales  
% menos 12 que se llenan con cero en el standard 802.11a menos las que se usan como tonos  
reservados)
```

```
nSymbol = ceil(nBit/nBitPerSymbol); % se reparten los 10000 datos entre las  
% portadoras que llevan información
```

```
% BPSK modulation
```

```
% bit0 --> -1
```

```
% bit1 --> +1
```

```
ipMod = 2*ip - 1; % los 0 y 1 se convierten en -1 y +1
```

```
ipMod = [ipMod zeros(1,nBitPerSymbol*nSymbol-nBit)]; % se completa con ceros al final  
% para formar un numero Redondo para las portadoras
```

```
ipMod = reshape(ipMod,nSymbol,nBitPerSymbol); % divide los datos en los simbolos que  
se enviaran
```

```
% modulando en OFDM
```

```
% st = []; % empty vector, aqui se iran guardando los resultados del papr de cada simbolo  
ofdm
```

```
index = [-32:31] ; % sin contar prefijo ciclico
```

```
% index [-26 to -1 1 to 26] , indices que se manejan segun standard 802.11a
```

```
subcarrierIndex = [-26:-1 1:7 (8+nToneRes):26]; % indice de las portadoras que llevan  
informacion
```

```
% en 802.11a se usan el 7 y el 21 como pilotos asi que escogeremos tonos
```

```
% reservados a partir del 8 y en los tonos reservados no se envia informacion
```

```
tonereservIndex = [8:8+nToneRes-1] ; % como la 7 es piloto cogemos tonos resrv a partir de  
la 8
```

```
Qred = Q(:,tonereservIndex+33) ; %matriz reducida de Q para los tonos reservados que  
usaremos
```

```
zz=0;
```

```
UMBRAL = 8:22 ;
```

```
for zz = 1:length(UMBRAL) ;
```

```
numCorrecciones=0;
```

```

% todo el siguiente proceso se realiza para cada simbolo

for ii = 1:nSymbol

Xf = zeros(nFFTSize,1);
% asignando bits a1 to a52 a los subcarriers que llevaran informacion
Xf(subcarrierIndex+nFFTSize/2+1) = ipMod(ii,:); % vector X a la entrada de la ifft
xt = 64*ifft(Xf,nFFTSize); % salida d la transformada inversa de fourier

% computing the peak to average power ratio for each symbol
meanSquareValue = xt*xt/length(xt);
peakValue = max(xt.*conj(xt)) ;
paprSymbol(ii) = peakValue/meanSquareValue;
paprSymboldB(ii) = 10*log10(paprSymbol(ii));

EXCESO(ii) = (paprSymboldB(ii) - 10*log10((UMBRAL(zz)^2)/meanSquareValue) ) ;
if EXCESO(ii)>0
    EXCESO(ii) = EXCESO(ii) ;
else
    EXCESO(ii) = 0 ;
end

% correccion del PAPR algoritmo gradiente

c = zeros(nFFTSize,1); % c es la señal correctora que va en los tonos reservados
numiterac = 0 ; % contara el numero de iteraciones del algoritmo (cuantas veces revisa el
mismo simbolo y corrige)
numiteracFin = 0 ; % guarda el numero total de iteraciones que le hizo a un simbolo ofdm
xfinal = xt + c ; % nuestra x resultante es la suma de la x original con los tonos reservados

while numiterac < maxit ; %mientras no se pase del numero maximo de iteraciones

    kk = 0 ; % contador de las muestras de la matriz x del simbolo actual analizandose
    picoExistentes = 0 ;
    cs = zeros(nFFTSize,1) ;

    for kk = 1 : nFFTSize ;
        if abs(xfinal(kk))>UMBRAL(zz) ;

```

```

    picoExistentes= picoExistentes+1 ;
    qrow = Qred(kk,:);
    cs = cs + (((xfinal(kk))-(UMBRAL(zz)*exp(i*angle(xfinal(kk)))))*Qred*qrow');
%correccion en base a algoritmo gradiente
    numiterac=numiterac+1 ;
end
end

if picoExistentes > 0 ; %si se encontraron picos se hizo correccion en ese simbolo , se
llevaria 1 iteracion mas
    numiteracFin = numiterac ; % el numero de iteraciones en que se realizo la reduccion
total
    c = c - ( Niu * cs ) ;
    xfinal = xt + c ;

else
    numiterac = maxit ; % como ya no hay picos en el smbolo picoExistentes=0 directamente
le da el valor para que salga del lazo

end

end

if (numiteracFin > 0)
    numCorrecciones = numCorrecciones+1 ; %este valor me dira cuantos simbolos tenian
problemas de papr

end

% computing the peak to average power ratio ya aplicada la tecnica de crreccion for each
symbol
peakValue = max(xfinal.*conj(xfinal));
paprMODSymbol(ii) = peakValue/meanSquareValue; % Papr con la correccion hecha
paprMODSymboldB(ii) = 10*log10(paprMODSymbol(ii));
Numiteraciones1(ii) = numiteracFin ;

```

```
% adding cyclic prefix of 16 samples
% xfinal_with_CP = [xfinal(49:64) xfinal];
% concatenating the symbols to form the final output
% st = [st xfinal_with_CP];
```

```
end
```

```
CORRECCION1 = paprSymboldB - paprMODSymboldB ;
numCorrec(zz)=numCorrecciones;
PromIteraciones(zz)= mean(Numiteraciones1)
```

```
end %final del for de la linea 81 usada para pruebas de varias corridas del codigo
```

```
figure(1)
plot(UMBRAI,numCorrec)
xlabel('valores de UMBRAI')
ylabel('numero de simbolos con correccion')
title('numero de simbolos OFDM corregidos')
grid on
```

```
figure(2)
plot(UMBRAI,PromIteraciones)
xlabel('valores de UMBRAI')
ylabel('numero de iteraciones que generalmente se necesitan para coregir')
title('numero de iteraciones aproximado necesario')
grid on
```

Variación del Código de matlab utilizado para observar resultados al cambiar valores de “Niu” en la técnica tonos reservados utilizando algoritmo gradiente

```
% Author : Juan Carlos Pinto y Miguel Cevallos,(tomando como punto de
% partida algoritmo de Krishna)
% Version : 1.0
% Date :
%
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

% codigo tonos reservados por algoritmo gradiente variando valor "Niu"

clear
nFFTSize = 64; % tamaño de la transformada inversa de fourier , para este caso lo haremos
coincidir con el numero de subportadoras
% for each symbol bits a1 to a52 are assigned to subcarrier
nBit = 10000; % numero de bits a transmitir
ip = rand(1,nBit) > 0.5; % generating 1's and 0's
% ip = zeros(1,nBit)+1 ; %este es un caso maximo para que todas las portadoras estan en fase
nportadoras = 64; % numero de subportadoras
nToneRes = 4; % numero de tonos reservados
modulacion = 1; % tipo de modulacion 1=BPSK
UMBRAL = 14 ; % maximo valor que se va a permitir a la entrada del HPA
maxit = 30 ; % maximo numero de iteraciones del algoritmo gradiente
Niu = 14 ; % tamaño de paso del algoritmo gradiente
picoExistentes = 0 ; % indica cuantos picos van apareciendo en un mismo simbolo OFDM
numiterac=0 ; % % numero de iteraciones en las que se hizo la correccion del papr en un
simbolo
picoExistentes2 = 0 ; % indica cuantos picos van apareciendo en un mismo simbolo OFDM
numiterac2=0 ; % % numero de iteraciones en las que se hizo la correccion del papr en un
simbolo
segraficoofdm=0 ; %indica si se grafico un simbolo con la correccion del papr
segraficoofdm2=0 ; %indica si se grafico un simbolo con la correccion del papr

% matriz Q
N = nFFTSize ; % tamaño de la matriz IFFT
n= 1 ; % subindice de las filas
m = 1 ; % subindice de las columnas

for n = 1:N
for m = 1:N
```

```
Q(n,m) = exp(0+i*(2*pi/N)*(n-1)*(m-1)) ;
```

```
end
```

```
end
```

```
Q= (1/(N^(0.5)))*Q ;
```

```
% a continuación se mapea para bpsk
```

```
nBitPerSymbol = nportadoras-12- nToneRes; % numero de Portadoras que llevaran  
información (nportadoras totales  
% menos 12 que se llenan con cero en el standard 802.11a menos las que se usan como tonos  
reservados)
```

```
nSymbol = ceil(nBit/nBitPerSymbol); % se reparten los 10000 datos entre las  
% portadoras que llevan información
```

```
% BPSK modulation
```

```
% bit0 --> -1
```

```
% bit1 --> +1
```

```
ipMod = 2*ip - 1; % los 0 y 1 se convierten en -1 y +1
```

```
ipMod = [ipMod zeros(1,nBitPerSymbol*nSymbol-nBit)]; % se completa con ceros al final  
% para formar un numero Redondo para las portadoras
```

```
ipMod = reshape(ipMod,nSymbol,nBitPerSymbol); % divide los datos en los simbolos que  
se enviaran
```

```
% modulando en OFDM
```

```
% st = []; % empty vector, aqui se iran guardando los resultados del papr de cada simbolo  
ofdm
```

```
index = [-32:31] ; % sin contar prefijo ciclico
```

```
% index [-26 to -1 1 to 26] , indices que se manejan segun standard 802.11a
```

```
subcarrierIndex = [-26:-1 1:7 (8+nToneRes):26]; % indice de las portadoras que llevan  
informacion
```

```
% en 802.11a se usan el 7 y el 21 como pilotos asi que escogeremos tonos
```

```
% reservados a partir del 8 y en los tonos reservados no se envia informacion
```

```
tonereservIndex = [8:8+nToneRes-1] ; % como la 7 es piloto cogemos tonos resrv a partir de  
la 8
```

```
Qred = Q(:,tonereservIndex+33) ; %matriz reducida de Q para los tonos reservados que  
usaremos
```

```
zz=0;
```

```
NIU = 5:15 ;
```

```
for zz = 1:length(NIU) ;
```

```
numCorrecciones=0;
```

```

% todo el siguiente proceso se realiza para cada simbolo

for ii = 1:nSymbol

Xf = zeros(nFFTSize,1);
% asignando bits a1 to a52 a los subcarriers que llevaran informacion
Xf(subcarrierIndex+nFFTSize/2+1) = ipMod(ii,:); % vector X a la entrada de la ifft
xt = 64*ifft(Xf,nFFTSize); % salida d la transformada inversa de fourier

% computing the peak to average power ratio for each symbol
meanSquareValue = xt*xt/length(xt);
peakValue = max(xt.*conj(xt)) ;
paprSymbol(ii) = peakValue/meanSquareValue;
paprSymboldB(ii) = 10*log10(paprSymbol(ii));

EXCESO(ii) = (paprSymboldB(ii) - 10*log10((UMBRAL^2)/meanSquareValue) ) ;
if EXCESO(ii)>0
    EXCESO(ii) = EXCESO(ii) ;
else
    EXCESO(ii) = 0 ;
end

% correccion del PAPR algoritmo gradiente

c = zeros(nFFTSize,1); % c es la señal correctora que va en los tonos reservados
numiterac = 0 ; % contara el numero de iteraciones del algoritmo (cuantas veces revisa el
mismo simbolo y corrige)
numiteracFin = 0 ; % guarda el numero total de iteraciones que le hizo a un simbolo ofdm
xfinal = xt + c ; % nuestra x resultante es la suma de la x original con los tonos reservados

while numiterac < maxit ; %mientras no se pase del numero maximo de iteraciones

    kk = 0 ; % contador de las muestras de la matriz x del simbolo actual analizandose
    picoExistentes = 0 ;
    cs = zeros(nFFTSize,1) ;

    for kk = 1 : nFFTSize ;
        if abs(xfinal(kk))>UMBRAL ;

```



```

    picoExistentes= picoExistentes+1 ;
    qrow = Qred(kk,:);
    cs = cs + (((xfinal(kk))-(UMBRAL*exp(i*angle(xfinal(kk)))))*Qred*qrow');
%correccion en base a algoritmo gradiente
    numiterac=numiterac+1 ;
end
end

if picoExistentes > 0 ; %si se encontraron picos se hizo correccion en ese simbolo , se
llevaria 1 iteracion mas
    numiteracFin = numiterac ; % el numero de iteraciones en que se realizo la reduccion
total
    c = c - ( NIU(zz) * cs ) ;
    xfinal = xt + c ;

else
    numiterac = maxit ; % como ya no hay picos en el smbolo picoExistentes=0 directamente
le da el valor para que salga del lazo

end

end

if (numiteracFin > 0)
    numCorrecciones = numCorrecciones+1 ; %este valor me dira cuantos simbolos tenian
problemas de papr

end

% computing the peak to average power ratio ya aplicada la tecnica de crreccion for each
symbol
peakValue = max(xfinal.*conj(xfinal));
paprMODSymbol(ii) = peakValue/meanSquareValue; % Papr con la correccion hecha
paprMODSymboldB(ii) = 10*log10(paprMODSymbol(ii));
Numiteraciones1(ii) = numiteracFin ;

% adding cyclic prefix of 16 samples
% xfinal_with_CP = [xfinal(49:64) xfinal];
% concatenating the symbols to form the final output

```

```
% st = [st xfinal_with_CP];
```

```
end
```

```
CORRECCION1 = paprSymboldB - paprMODSymboldB ;  
EXACTITUD1= (1-((abs(EXCESO-CORRECCION1))/EXCESO))*100;  
EXAC(zz) = mean(EXACTITUD1);  
PromIteraciones(zz)= mean(Numiteraciones1) ;
```

```
end %final del for de la linea 81 usada para pruebas de varias corridas del codigo
```

```
figure(1)  
plot(NIU,EXAC)  
xlabel('valores de NIU')  
ylabel('promedio de aproximacion %')  
title('promedio de aproximacion segun el valor Niu')  
grid on
```

```
figure(2)  
plot(NIU,PromIteraciones)  
xlabel('valores de NIU')  
ylabel('numero de iteraciones que generalmente se necesitan para coregir')  
title('numero de iteraciones aproximado necesario segun el valor Niu')  
grid on
```

Variación del Código de matlab utilizado para observar resultados al cambiar valores del numero de tonos en la técnica tonos reservados utilizando algoritmo gradiente

```
% Author      : Juan Carlos Pinto y Miguel Cevallos, (tomando como
punto de
% partida algoritmo de Krishna)
% Version     : 1.0
% Date       :
% %%%%%%%%%%

% codigo tonos reservados por algoritmo gradiente variando numero
de tonos

clear
nFFTSize = 64; % tamaño de la transformada inversa de fourier ,
para este caso lo haremos coincidir con el numero de subportadoras
% for each symbol bits al to a52 are assigned to subcarrier
nBit = 10000; % numero de bits a transmitir
ip = rand(1,nBit) > 0.5; % generating 1's and 0's
% ip = zeros(1,nBit)+1 ; %este es un caso maximo para que todas las
poradoras estan en fase
nportadoras = 64; % numero de subportadoras
nToneRes = 4; % numero de tonos reservados
modulacion = 1; % tipo de modulacion 1=BPSK
UMBRAL = 14 ; % maximo valor que se va a permitir a la entrada del
HPA
maxit = 40 ; % maximo numero de iteraciones del algoritmo gradiente
Niu = 14 ; % tamaño de paso del algoritmo gradiente
picoExistentes = 0 ; % indica cuantos picos van apareciendo en un
mismo simbolo OFDM
numiterac=0 ; % % numero de iteraciones en las que se hizo la
correccion del papr en un simbolo
picoExistentes2 = 0 ; % indica cuantos picos van apareciendo en un
mismo simbolo OFDM
numiterac2=0 ; % % numero de iteraciones en las que se hizo la
correccion del papr en un simbolo
segraficoofdm=0 ; %indica si se grafico un simbolo con la correccion
del papr
segraficoofdm2=0 ; %indica si se grafico un simbolo con la
correccion del papr

zz=0;
nToneRes = 3:9 ;
for zz = 1:length(nToneRes) ;
numCorrecciones=0;
```

```

% matriz Q
N = nFFTSsize ; % tamaño de la matriz IFFT
n = 1 ; % subíndice de las filas
m = 1 ; % subíndice de las columnas

for n = 1:N
for m = 1:N

Q(n,m) = exp(0+i*(2*pi/N)*(n-1)*(m-1)) ;

end
end
Q= (1/(N^(0.5))) * Q ;

% a continuación se mapea para bpsk

nBitPerSymbol = nportadoras-12- nToneRes(zz); % número de
Portadoras que llevarán información (nportadoras totales
% menos 12 que se llenan con cero en el standard 802.11a menos las
que se usan como tonos reservados)
nSymbol = ceil(nBit/nBitPerSymbol); % se reparten los 10000 datos
entre las
% portadoras que llevan
información

% BPSK modulation
% bit0 --> -1
% bit1 --> +1
ipMod = 2*ip - 1; % los 0 y 1
se convierten en -1 y +1
ipMod = [ipMod zeros(1,nBitPerSymbol*nSymbol-nBit)]; % se
completa con ceros al final
% para formar un número
Redondo para las portadoras
ipMod = reshape(ipMod,nSymbol,nBitPerSymbol); % divide los datos en
los símbolos que se enviarán

% modulando en OFDM
% st = []; % empty vector, aquí se irán guardando los resultados del
papr de cada símbolo ofdm

index = [-32:31] ; % sin contar prefijo cíclico
% index [-26 to -1 1 to 26] , índices que se manejan según
standard 802.11a
subcarrierIndex = [-26:-1 1:7 (8+nToneRes(zz)):26]; % índice de las
portadoras que llevan información

```

```

% en 802.11a se usan el 7 y el 21 como pilotos asi que escogeremos
tonos
% reservados a partir del 8 y en los tonos reservados no se envia
informacion
tonereservIndex = [8:8+nToneRes(zz)-1] ; % como la 7 es piloto
cogemos tonos resrv a partir de la 8

Qred = Q(:,tonereservIndex+33) ; %matriz reducida de Q para los
tonos reservados que usaremos

% todo el siguiente proceso se realiza para cada simbolo

for ii = 1:nSymbol

Xf = zeros(nFFTSsize,1);
% asignando bits al to a52 a los subcarriers que llevaran
informacion
Xf(subcarrierIndex+nFFTSsize/2+1) = ipMod(ii,:); %vector X a la
entrada de la ifft
xt = 64*ifft(Xf,nFFTSsize); %salida d la transformada inversa de
fourier

% computing the peak to average power ratio for each symbol
meanSquareValue = xt'*xt/length(xt);
peakValue = max(xt.*conj(xt)) ;
paprSymbol(ii) = peakValue/meanSquareValue;
paprSymboldB(ii) = 10*log10(paprSymbol(ii));

EXCESO(ii) = (paprSymboldB(ii) -
10*log10((UMBRAL^2)/meanSquareValue) ) ;
if EXCESO(ii)>0
    EXCESO(ii) = EXCESO(ii) ;
else
    EXCESO(ii) = 0 ;
end

% correccion del PAPR algoritmo gradiente

c = zeros(nFFTSsize,1); % c es la señal corectora que va en los tonos
reservados
numiterac = 0 ; % contara el numero de iteraciones del algoritmo
(cuantas veces revisa el mismo simbolo y corrige)
numiteracFin = 0 ; % guarda el numero total de iteraciones que le
hizo a un simbolo ofdm

```

```
xfinal = xt + c ; % nuestra x resultante es la suma de la x original
con los tonos reservados
```

```
while numiterac < maxit ; %mientras no se pase del numero maximo de
iteraciones
```

```
    kk = 0 ; % contador de las muestras de la matriz x del simbolo
actual analizandose
    picoExistentes = 0 ;
    cs = zeros(nFFTSize,1) ;
```

```
    for kk = 1 : nFFTSize ;
        if abs(xfinal(kk))>UMBRAL ;
            picoExistentes= picoExistentes+1 ;
            grow = Qred(kk,:) ;
            cs = cs + ((xfinal(kk))-
(UMBRAL*exp(i*angle(xfinal(kk)))))*Qred*grow') ; %correccion en base
a algoritmo gradiente
            numiterac=numiterac+1 ;
        end
    end
```

```
    if picoExistentes > 0 ; %si se encontraron picos se hizo
correccion en ese simbolo , se llevaria 1 iteracion mas
        numiteracFin = numiterac ; % el numero de iteraciones en que
se realizo la reduccion total
        c = c - ( Niu * cs ) ;
        xfinal = xt + c ;
```

```
    else
        numiterac = maxit ; % como ya no hay picos en el smbolo
picoExistentes=0 directamente le da el valor para que salga del lazo
    end
```

```
end
```

```
    if (numiteracFin > 0)
        numCorrecciones = numCorrecciones+1 ; %este valor me dira
cuantos simbolos tenian problemas de papr
```

```
    end
```

```
% computing the peak to average power ratio ya aplicada la tecnica
de crreccion for each symbol
peakValue = max(xfinal.*conj(xfinal));
paprMODSymbol(ii) = peakValue/meanSquareValue; % Papr con la
correccion hecha
paprMODSymboldB(ii) = 10*log10(paprMODSymbol(ii));
Numiteraciones1(ii) = numiteracFin ;
```

```
end
```

```
CORRECCION1 = paprSymboldB - paprMODSymboldB ;
numCorrec(zz)=numCorrecciones;
PromIteraciones(zz)= mean(Numiteraciones1)
```

```
end %final del for de la linea 32 usada para pruebas de varias
corridad del codigo
```

```
figure(1)
plot(nToneRes,PromIteraciones)
xlabel('numero de tonos reservados')
ylabel('numero de iteraciones que generalmente se necesitan para
coregir')
title('numero de iteraciones aproximado necesario')
grid on
```

APENDICE F

Tabla de Resultados tomados en una corrida del algoritmo

Tecnica1 = tonos reservados con algoritmo gradiente

Tecnica2 = tonos reservados con pulso gaussiano

símbolo OFDM enviado	PAPR	exceso sobre el nivel del PAPR máximo	corrección con algoritmo gradiente	corrección con pulso gaussiano	numero de iteraciones en hacer la correccion1	numero de iteraciones en hacer la correccion 2
ii	paprSymboldB'	EXCESO'	CORRECCION1'	CORRECCION2'	Numiteraciones1'	Numiteraciones2'
1	5,1939	0,0000	0,0000	0,0000	0	0
2	5,5579	0,0000	0,0000	0,0000	0	0
3	7,3478	1,2376	1,2376	1,2376	30	23
4	8,2930	2,1829	2,1829	2,1829	18	1
5	4,7712	0,0000	0,0000	0,0000	0	0
6	7,0456	0,9355	1,2821	0,9355	2	2
7	7,1471	1,0370	1,0370	1,0370	32	30
8	5,8288	0,0000	0,0000	0,0000	0	0
9	6,2060	0,0959	0,0959	0,0959	30	24
10	5,5797	0,0000	0,0000	0,0000	0	0
11	5,8177	0,0000	0,0000	0,0000	0	0
12	5,8397	0,0000	0,0000	0,0000	0	0
13	7,2700	1,1598	1,1598	1,1598	17	1
14	5,9285	0,0000	0,0000	0,0000	0	0
15	6,9883	0,8781	0,8781	0,8781	30	13
16	6,2750	0,1648	0,1801	0,1648	2	3
17	5,9341	0,0000	0,0000	0,0000	0	0
18	6,7659	0,6558	0,6558	0,6558	30	17
19	6,1020	0,0000	0,0000	0,0000	0	0
20	6,5664	0,4563	0,3953	0,4431	30	30
21	6,0477	0,0000	0,0000	0,0000	0	0
22	6,3664	0,2563	0,2563	0,2563	30	2
23	6,1101	0,0000	0,0000	0,0000	0	0
24	5,6028	0,0000	0,0000	0,0000	0	0
25	5,6634	0,0000	0,0000	0,0000	0	0

26	5,9963	0,0000	0,0000	0,0000	0	0
27	5,6609	0,0000	0,0000	0,0000	0	0
28	5,4191	0,0000	0,0000	0,0000	0	0
29	7,0313	0,9212	0,9212	0,9212	30	2
30	5,5663	0,0000	0,0000	0,0000	0	0
31	6,4274	0,3172	0,3172	0,3172	30	2
32	6,1946	0,0845	0,0686	0,0707	30	31
33	5,3360	0,0000	0,0000	0,0000	0	0
34	5,2826	0,0000	0,0000	0,0000	0	0
35	7,2700	1,1598	1,1598	1,1598	17	1
36	5,0810	0,0000	0,0000	0,0000	0	0
37	7,4143	1,3042	1,3042	1,3042	30	31
38	6,4547	0,3446	0,3446	0,3446	30	26
39	6,0749	0,0000	0,0000	0,0000	0	0
40	4,7712	0,0000	0,0000	0,0000	0	0
41	5,7344	0,0000	0,0000	0,0000	0	0
42	6,9039	0,7937	0,7680	0,7913	32	32
43	5,7252	0,0000	0,0000	0,0000	0	0
44	7,0390	0,9288	0,8333	0,8179	30	32
45	4,6784	0,0000	0,0000	0,0000	0	0
46	4,5996	0,0000	0,0000	0,0000	0	0
47	5,5251	0,0000	0,0000	0,0000	0	0
48	5,3610	0,0000	0,0000	0,0000	0	0
49	8,6700	2,5599	2,5573	2,5599	30	31
50	7,0805	0,9704	0,9704	0,9704	30	21
51	7,2748	1,1647	1,1647	1,1647	30	2
52	6,4191	0,3090	0,3559	0,3090	2	2
53	6,2478	0,1377	0,1377	0,1377	30	18
54	5,1428	0,0000	0,0000	0,0000	0	0
55	7,6896	1,5794	1,5794	1,5794	30	17
56	6,4991	0,3889	0,3612	0,3729	30	30
57	8,2930	2,1829	2,6614	2,1829	3	2
58	5,2377	0,0000	0,0000	0,0000	0	0
59	7,2700	1,1598	1,1476	1,1598	31	14
60	8,3207	2,2106	2,2103	2,2054	30	32
61	5,3941	0,0000	0,0000	0,0000	0	0
62	5,1573	0,0000	0,0000	0,0000	0	0
63	3,9602	0,0000	0,0000	0,0000	0	0
64	6,3547	0,2445	0,2445	0,2445	32	4
65	5,7626	0,0000	0,0000	0,0000	0	0
66	5,8424	0,0000	0,0000	0,0000	0	0
67	9,2082	3,0980	1,3902	1,3961	32	30
68	7,2700	1,1598	1,1598	1,1598	31	1
69	10,0360	3,9259	3,1665	2,7881	30	34
70	6,3437	0,2335	0,3752	0,2335	2	2
71	5,7129	0,0000	0,0000	0,0000	0	0

72	4,9374	0,0000	0,0000	0,0000	0	0
73	6,0609	0,0000	0,0000	0,0000	0	0
74	7,5383	1,4281	0,7641	0,9261	30	31
75	6,9095	0,7993	0,7993	0,7993	30	2
76	9,2082	3,0980	3,0980	3,0980	20	29
77	6,8354	0,7252	0,6188	0,6887	32	30
78	5,0229	0,0000	0,0000	0,0000	0	0
79	5,3672	0,0000	0,0000	0,0000	0	0
80	7,2700	1,1598	1,1598	1,1598	17	1
81	4,1189	0,0000	0,0000	0,0000	0	0
82	6,9070	0,7968	0,7968	0,7968	30	16
83	6,0681	0,0000	0,0000	0,0000	0	0
84	6,0381	0,0000	0,0000	0,0000	0	0
85	5,6110	0,0000	0,0000	0,0000	0	0
86	6,2384	0,1282	0,1282	0,1282	30	17
87	5,5964	0,0000	0,0000	0,0000	0	0
88	5,6027	0,0000	0,0000	0,0000	0	0
89	6,1101	0,0000	0,0000	0,0000	0	0
90	8,2930	2,1829	2,1829	2,1829	18	1
91	6,0923	0,0000	0,0000	0,0000	0	0
92	5,9769	0,0000	0,0000	0,0000	0	0
93	8,3463	2,2362	2,2048	2,1934	32	30
94	4,7153	0,0000	0,0000	0,0000	0	0
95	7,8017	1,6916	1,1994	1,0814	32	31
96	4,9019	0,0000	0,0000	0,0000	0	0
97	5,7403	0,0000	0,0000	0,0000	0	0
98	6,1159	0,0058	0,0058	0,0058	22	2
99	7,2700	1,1598	1,1598	1,1598	17	1
100	5,1884	0,0000	0,0000	0,0000	0	0
101	5,9573	0,0000	0,0000	0,0000	0	0
102	7,7844	1,6742	1,5188	1,4226	32	30
103	6,6605	0,5504	0,5778	0,5504	2	2
104	7,2700	1,1598	1,1598	1,1598	17	1
105	6,3111	0,2009	0,2009	0,2009	30	15
106	5,7465	0,0000	0,0000	0,0000	0	0
107	6,7594	0,6493	0,6493	0,6493	14	2
108	4,5230	0,0000	0,0000	0,0000	0	0
109	5,8408	0,0000	0,0000	0,0000	0	0
110	6,1965	0,0863	0,0863	0,0863	30	30
111	8,2930	2,1829	2,1829	2,1829	18	1
112	5,6708	0,0000	0,0000	0,0000	0	0
113	10,0360	3,9259	0,7485	1,9685	32	35
114	7,2700	1,1598	1,1598	1,1598	17	1
115	7,0325	0,9223	0,9223	0,9223	30	15
116	4,8837	0,0000	0,0000	0,0000	0	0
117	6,7037	0,5936	0,4834	0,4476	32	33

118	8,0227	1,9125	1,9125	1,9125	30	21
119	5,5780	0,0000	0,0000	0,0000	0	0
120	6,6446	0,5345	0,5922	0,5345	2	2
121	5,5731	0,0000	0,0000	0,0000	0	0
122	7,0014	0,8913	0,8166	0,8832	34	33
123	7,2700	1,1598	0,6695	0,4778	30	32
124	4,8720	0,0000	0,0000	0,0000	0	0
125	5,4750	0,0000	0,0000	0,0000	0	0
126	6,7973	0,6872	0,6868	0,6872	30	30
127	4,5890	0,0000	0,0000	0,0000	0	0
128	5,7188	0,0000	0,0000	0,0000	0	0
129	5,3870	0,0000	0,0000	0,0000	0	0
130	7,2700	1,1598	1,1598	1,1598	17	1
131	6,1666	0,0565	0,0565	0,0565	30	24
132	5,1415	0,0000	0,0000	0,0000	0	0
133	5,5891	0,0000	0,0000	0,0000	0	0
134	7,7024	1,5922	1,5922	1,5922	30	3
135	6,2637	0,1535	0,1535	0,1535	30	28
136	7,3573	1,2471	1,1964	1,2419	30	30
137	7,3254	1,2153	1,7551	1,2153	2	3
138	7,9080	1,7978	0,9501	0,9449	30	31
139	7,0517	0,9416	1,2418	0,9416	2	2
140	7,0909	0,9807	0,9807	0,9807	30	3
141	6,1101	0,0000	0,0000	0,0000	0	0
142	4,6666	0,0000	0,0000	0,0000	0	0
143	6,2947	0,1846	0,1846	0,1846	30	2
144	5,9849	0,0000	0,0000	0,0000	0	0
145	6,2891	0,1790	0,1790	0,1790	30	15
146	6,4484	0,3382	0,3367	0,3382	30	30
147	6,9619	0,8518	0,0775	0,1213	32	32
148	5,3547	0,0000	0,0000	0,0000	0	0
149	6,5170	0,4068	0,4068	0,4068	30	16
150	5,7465	0,0000	0,0000	0,0000	0	0
151	6,0923	0,0000	0,0000	0,0000	0	0
152	4,4232	0,0000	0,0000	0,0000	0	0
153	5,6727	0,0000	0,0000	0,0000	0	0
154	5,5469	0,0000	0,0000	0,0000	0	0
155	3,9588	0,0000	0,0000	0,0000	0	0
156	5,4097	0,0000	0,0000	0,0000	0	0
157	7,7902	1,6801	1,7974	1,6801	2	2
158	8,1824	2,0723	2,0707	1,9970	30	30
159	6,4296	0,3194	0,3194	0,3194	30	30
160	5,9348	0,0000	0,0000	0,0000	0	0
161	7,8029	1,6927	1,8075	1,6927	2	3
162	5,7769	0,0000	0,0000	0,0000	0	0
163	5,2980	0,0000	0,0000	0,0000	0	0

164	8,8426	2,7324	2,7324	2,7324	30	32
165	6,0311	0,0000	0,0000	0,0000	0	0
166	6,4321	0,3219	0,3219	0,3219	30	30
167	7,5760	1,4659	1,4659	1,4649	30	31
168	7,1031	0,9930	0,9930	0,9930	30	30
169	4,7296	0,0000	0,0000	0,0000	0	0
170	8,0997	1,9896	1,9896	1,9896	30	15
171	6,2238	0,1136	0,1136	0,1136	30	15
172	4,4603	0,0000	0,0000	0,0000	0	0
173	5,7296	0,0000	0,0000	0,0000	0	0
174	6,5424	0,4322	0,4322	0,4322	30	2
175	6,1360	0,0259	0,0259	0,0259	28	4
176	6,5904	0,4803	0,4803	0,4803	30	19
177	5,0525	0,0000	0,0000	0,0000	0	0
178	6,3717	0,1701	0,1701	0,1701	30	17
179	5,8517	0,0000	0,0000	0,0000	0	0
180	4,9935	0,0000	0,0000	0,0000	0	0
181	6,8008	0,5993	0,5993	0,5993	17	1
182	6,0093	0,0000	0,0000	0,0000	0	0
183	6,5613	0,3597	0,3597	0,3597	22	3
184	5,8147	0,0000	0,0000	0,0000	0	0
185	6,5541	0,3525	0,3521	0,3525	30	30
186	5,7692	0,0000	0,0000	0,0000	0	0
187	5,9596	0,0000	0,0000	0,0000	0	0
188	8,6002	2,3986	2,3986	2,3986	30	18
189	6,4726	0,2711	0,2711	0,2711	22	2
190	6,2833	0,0818	0,0818	0,0818	30	30
191	5,5579	0,0000	0,0000	0,0000	0	0
192	7,9436	1,7420	1,7420	1,7420	30	16
193	7,1822	0,9806	0,8764	0,8647	30	30
194	4,1069	0,0000	0,0000	0,0000	0	0
195	4,9395	0,0000	0,0000	0,0000	0	0
196	6,0099	0,0000	0,0000	0,0000	0	0
197	6,7959	0,5943	0,5923	0,5943	32	5
198	4,1214	0,0000	0,0000	0,0000	0	0
199	5,5371	0,0000	0,0000	0,0000	0	0
200	5,5298	0,0000	0,0000	0,0000	0	0
201	6,7929	0,5913	0,6076	0,5913	2	2
202	5,7311	0,0000	0,0000	0,0000	0	0
203	6,8531	0,6516	0,6513	0,6516	32	32
204	6,4088	0,2073	0,0687	0,0355	32	32
205	4,7248	0,0000	0,0000	0,0000	0	0
206	6,8697	0,6682	0,6682	0,6682	30	31
207	5,9861	0,0000	0,0000	0,0000	0	0
208	5,8140	0,0000	0,0000	0,0000	0	0
209	7,8880	1,6864	1,6864	1,6864	18	1

BIBLIOGRAFÍA

[1] ANDGART NIKLAS, "Peak and Power Reduction in Multicarrier Communication System" , Department of Information Technology Lund University, Londres, 2005, 219p

[2] ANDREWS JEFFREY , "Orthogonal Frequency Division Multiplexing (OFDM)", 2006, http://wncg.org/ee381k2/lecturenotes/Chapter4_rev3.pdf, 35p

[3] SANKAR KRISHNA, blog de internet
<http://www.dsplog.com/2008/02/24/peak-to-average-power-ratio-for-ofdm/> , 2009

[4] TELLADO JOSE, "Multicarrier modulation with low PAR", Kluwer Academic Publishers, Dordrecht, 2000, 152p

[5] LOPEZ ALBERTO, "Overview on Crest Factor Reduction Techniques to reduce Peak to Average Power ratio in OFDM Signals", Escola Politècnica Superior de Castelldefels, Castelldefels, 2009, 62p

[6] GREGORIO FERNANDO, "Amalysis and Compensation of Nonlinear Power Amplifier Effects in Multi-Antenna System", Signal Processing Laboratory of Electrical and Communications Engineering Department of HELSINKI UNIVERSITY, helsinki, 2007, 133p

[7] JAYALATH DHAMMIKA, Presentation at ACoRN OFDM Workshop University of Melbourne "Peak Power of OFDM Signals.ppt" , Queensland University of Technology, Queensland , 2008

[8] DEVLIN COROLE, ZHU ANDING, BRAZIL THOMAS, "Gaussian Pulse Based Tone Reservation for Reducing PAPR of OFDM Signals", School of Electrical, Electronic and Mechanical Engineering of University College Dublín, Ireland, 2007, <http://www.pubzone.org/dblp/conf/vtc/DevlinZB07>