

Trabajo Fin de Máster Máster Universitario en Ingeniería de Telecomunicación

Esquemas de aprendizaje directo en predistorsión digital de amplificadores de potencia

Autor: Ildelfonso Jiménez Silva

Tutores: Juan Antonio Becerra González

María José Madero Ayora

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020



Trabajo Fin de Máster
Máster Universitario en Ingeniería de Telecomunicación

Esquemas de aprendizaje directo en predistorsión digital de amplificadores de potencia

Autor:

Ildefonso Jiménez Silva

Tutores:

Juan Antonio Becerra González

Profesor Sustituto Interino

María José Madero Ayora

Profesora Titular

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2020

Trabajo Fin de Máster: Esquemas de aprendizaje directo en predistorsión digital de amplificadores de potencia

Autor: Ildelfonso Jiménez Silva
Tutores: Juan Antonio Becerra González
María José Madero Ayora

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Agradecimientos

En primer lugar, me gustaría agradecer a mis padres, quienes siempre me han apoyado y que nunca dudaron de que conseguiría las metas que me propusiera.

También quiero dar las gracias a Cristina, mi mayor apoyo, quien siempre ha sido capaz de motivarme en los momentos más difíciles, que me ha ofrecido su ayuda en todo momento y que siempre ha estado a mi lado, alegrándose de mis logros tanto o incluso más que yo.

Otro pilar fundamental durante esta carrera lo componen los que hoy, más que compañeros, son unos de mis mejores amigos: Abraham, Daniel, Diego y Guillermo. Ha sido un placer compartir todos estos años, aprendiendo juntos y atesorando grandes momentos en común. Estoy seguro de que, aunque nuestros caminos se separen, llegaréis muy lejos y, por supuesto, siempre seguiremos siendo un gran equipo.

Finalmente, quiero agradecer a los tutores de este trabajo, Juan Antonio y María José. Por un lado, por su continua ayuda a lo largo del desarrollo de este trabajo, con las productivas y, en muchas ocasiones, divertidas reuniones. Por otro lado, por haber confiado en mí para solicitar la *Undergraduate/Pre-Graduate Scholarship* de la IEEE MTT-S, ya que sin su apoyo no la habría conseguido.

Ildefonso Jiménez Silva

Sevilla, 2020

Resumen

En el presente trabajo se han llevado a cabo una serie de experimentos con *hardware* para realizar una comparativa de diferentes técnicas de aprendizaje directo en predistorsión digital (DPD) de amplificadores de potencia (PA). Para ello, se introducen conceptos teóricos acerca de la no linealidad y la distorsión que producen los amplificadores de potencia para poder, a continuación, revisar el modelado del comportamiento de los amplificadores mediante series de Volterra.

Con ello de base, es posible introducir la predistorsión digital, así como las diferentes arquitecturas que se pueden emplear para, posteriormente, introducir los distintos métodos y algoritmos utilizados para comparar diferentes técnicas de aprendizaje directo, analizando las ventajas y los inconvenientes de aplicar cada una de ellas.

Abstract

Various experiments using hardware have been carried out in the present work with the aim of comparing different direct learning techniques in digital predistortion (DPD) of power amplifiers (PA). For this purpose, theoretical concepts about non-linearity and distortion effects produced by power amplifiers have been introduced in order to be able to review the power amplifiers behaviour modelling through Volterra series.

Based on this, it is possible to present digital predistortion as well as possible architectures that can be implemented. Then, some direct learning algorithms are described in order to make some experiments and to compare the different techniques, analysing the advantages and disadvantages of using each of them.

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
1 Introducción	1
2 Amplificadores de potencia	3
2.1 Sistemas lineales y no lineales	3
2.2 Distorsión no lineal	4
2.3 Caracterización de amplificadores	5
3 Modelado del comportamiento de los amplificadores	11
3.1 Series de Volterra	12
3.2 Modelos basados en series de Volterra	14
3.3 Notación matricial	16
3.4 Selección de componentes: DOMP	16
4 Linealización de amplificadores y predistorsión digital	19
4.1 Linealización de un amplificador	19
4.2 Predistorsión digital (DPD)	19
4.3 Aprendizaje indirecto y aprendizaje directo	22
5 Técnicas y experimentos	27
5.1 Configuración de los experimentos	27
5.2 Aprendizaje directo con bucle cerrado	30
5.3 Aprendizaje directo de base adaptable	37
5.4 Aprendizaje directo con matriz de covarianza	42
5.5 Aprendizaje directo con selección de componentes	45
5.6 Aplicación de CFR y normalización	49
5.7 Pruebas de estabilidad y coste computacional	54
6 Conclusiones	57
6.1 Conclusión	57
6.2 Líneas futuras	58
Bibliografía	59
<i>Índice de Figuras</i>	61
<i>Índice de Tablas</i>	63

Glosario

65

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
<i>Notación</i>	XI
1 Introducción	1
2 Amplificadores de potencia	3
2.1 Sistemas lineales y no lineales	3
2.2 Distorsión no lineal	4
2.3 Caracterización de amplificadores	5
2.3.1 Figuras de mérito	5
Punto de compresión de 1 dB	5
Relación de pico a valor medio (PAPR)	5
Relación de intermodulación	6
Relación de potencia de canal adyacente (ACPR)	6
Magnitud del vector error (EVM)	7
Error cuadrático medio normalizado (NMSE)	7
2.3.2 Características AM-AM y AM-PM	8
2.3.3 Efectos de memoria	9
3 Modelado del comportamiento de los amplificadores	11
3.1 Series de Volterra	12
3.1.1 Modelo general de Series de Volterra	12
3.1.2 Series de Volterra como desarrollo de Series de Taylor	12
3.1.3 Series de Volterra como desarrollo de un sistema lineal	13
3.1.4 Series de Volterra en tiempo discreto	14
3.2 Modelos basados en series de Volterra	14
3.2.1 Modelo completo de Volterra (FV)	14
3.2.2 Modelo polinomial con memoria (MP)	14
3.2.3 Modelo polinomial con memoria generalizado (GMP)	15
3.3 Notación matricial	16
3.4 Selección de componentes: DOMP	16
4 Linealización de amplificadores y predistorsión digital	19
4.1 Linealización de un amplificador	19
4.2 Predistorsión digital (DPD)	19
4.2.1 Diseño y funcionamiento de un DPD	20
4.2.2 Evaluación del DPD	21
4.3 Aprendizaje indirecto y aprendizaje directo	22
4.3.1 Controlador de autoajuste vs. arquitecturas de aprendizaje directo e indirecto	22

4.3.2	Arquitectura de aprendizaje indirecto	23
4.3.3	Arquitectura de aprendizaje directo	24
5	Técnicas y experimentos	27
5.1	Configuración de los experimentos	27
5.1.1	WebLab de Chalmers	27
5.1.2	Características de la señal	28
5.1.3	Modelado del amplificador	28
5.1.4	Análisis de las medidas	29
5.2	Aprendizaje directo con bucle cerrado	30
5.2.1	Funcionamiento del algoritmo	30
5.2.2	Experimento 1: Variación del factor de ponderación	32
5.3	Aprendizaje directo de base adaptable	37
5.3.1	Funcionamiento del algoritmo	37
5.3.2	Experimento 2: Variación del factor de ponderación con base adaptable	37
5.4	Aprendizaje directo con matriz de covarianza	42
5.4.1	Funcionamiento del algoritmo	42
5.4.2	Experimento 3: Análisis de las prestaciones usando la matriz de covarianza	43
5.4.3	Covarianza de Tyler	44
5.5	Aprendizaje directo con selección de componentes	45
5.5.1	Introducción al experimento	45
5.5.2	Experimento 4: Análisis del efecto de la selección de componentes	45
5.6	Aplicación de CFR y normalización	49
5.6.1	Técnicas utilizadas	49
5.6.2	Experimento 5: Aplicando CFR y normalizando	49
5.7	Pruebas de estabilidad y coste computacional	54
5.7.1	Introducción al experimento	54
5.7.2	Experimento 6: Número de condicionamiento y tiempo de aprendizaje	54
6	Conclusiones	57
6.1	Conclusión	57
6.2	Líneas futuras	58
	Bibliografía	59
	<i>Índice de Figuras</i>	61
	<i>Índice de Tablas</i>	63
	<i>Glosario</i>	65

Notación

Re	Parte real
Im	Parte imaginaria
sen	Función seno
cos	Función coseno
e	Número e
e^{jx}	Exponencial compleja
$e^{j2\pi x}$	Exponencial compleja con 2π
e^{-jx}	Exponencial compleja negativa
$e^{-j2\pi x}$	Exponencial compleja negativa con 2π
\mathbf{A}^\top	Transpuesto de \mathbf{A}
\mathbf{A}^{-1}	Inversa de la matriz \mathbf{A}
\mathbf{A}^\dagger	Matriz pseudoinversa de la matriz \mathbf{A}
\mathbf{A}^H	Transpuesto y conjugado de \mathbf{A}
\mathbf{A}^*	Conjugado de \mathbf{A}
$\ \mathbf{A}\ _2$	Norma l_2 de \mathbf{A}
\emptyset	Conjunto vacío
\otimes	Producto de Kronecker
\mathbf{C}_x	Matriz de covarianza de x

1 Introducción

The day you stop learning is the day you begin decaying.

ISAAC ASIMOV

La reducción del consumo de energía eléctrica se trata actualmente de una de las grandes preocupaciones a nivel mundial, no únicamente por el ahorro económico que ello implicaría, sino también por la reducción de la contaminación asociada a la producción de electricidad. Además, entre el 2007 y el 2011, el consumo eléctrico de las redes de comunicaciones representaba aproximadamente el 10% del consumo total en todo el mundo, además de estar incrementándose anualmente [1]. Esto se debe a la continua instalación de nuevas estaciones base para ofrecer nuevos servicios de banda ancha inalámbricos.

Concretamente, las estaciones base emplean grandes cantidades de energía para transmitir información y, en particular, el dispositivo que requiere un mayor consumo se trata del amplificador de potencia o, en inglés, *Power Amplifier* (PA). Estos presentan una característica importante a la hora de operar, puesto que su eficiencia es mayor cuando se trabaja con altos niveles de potencia. Sin embargo, a ciertos niveles de potencia se comienza a producir un efecto indeseable en la señal de salida: la distorsión.

Por otro lado, los nuevos estándares de comunicaciones incluidos en la cuarta y quinta generación de comunicaciones móviles emplean señales que cuentan con altos picos respecto a su valor medio que, junto con la distorsión producida por los amplificadores, hacen que sea necesario reducir el punto de operación y, consecuentemente, reducir la eficiencia del dispositivo de comunicaciones.

Como se presentará a lo largo de este trabajo, este problema ha sido objeto de investigación en los últimos años, y la técnica denominada como predistorsión digital o, en inglés, *Digital Predistortion* (DPD) se ha postulado como una de las mejores soluciones. Por este motivo, existe una gran variedad de investigaciones con el objetivo de diseñar métodos y algoritmos para conseguir una predistorsión digital de modo que se obtenga un sistema de comunicaciones con las mejores prestaciones posibles, manteniendo eficiencia, altos niveles de potencia y una reducida distorsión.

Además, dentro de la predistorsión digital existen diferentes arquitecturas, dentro de las cuales se encuentran las de aprendizaje directo, a las que se prestará una mayor atención en el presente trabajo. Así, en el Capítulo 2, se introducirá la problemática de los amplificadores de potencia haciendo una revisión de las no linealidades que estos presentan y definiendo una serie de parámetros que pueden ser utilizados para caracterizarlos. Tras ello, en el Capítulo 3, se explicarán algunas de las formas más comunes de modelar el comportamiento de un amplificador mediante las series de Volterra. A continuación, en el Capítulo 4, se explicará en qué consiste la linealización de un amplificador y, particularmente, la predistorsión digital, así como las principales arquitecturas que existen. En el Capítulo 5, se expondrán los experimentos realizados haciendo uso de *hardware* en el presente trabajo, presentando en primer lugar la configuración de los experimentos, los diferentes métodos utilizados y mostrando los resultados obtenidos para cada uno de ellos. Finalmente, el Capítulo 6 expondrá las conclusiones finales que se han podido obtener con la realización de este proyecto, así como algunas de las líneas futuras que podrían sucederse tras la finalización del presente trabajo.

2 Amplificadores de potencia

I learned very early the difference between knowing the name of something and knowing something.

RICHARD P. FEYNMANN

Los amplificadores de potencia de radiofrecuencia (RF), representan un componente esencial de los sistemas de comunicaciones, concretamente en los transmisores, con la función de aumentar el nivel de potencia de la señal de entrada y que será transmitida mediante una antena. Sin embargo, en la realidad los amplificadores de potencia no son tan ideales, puesto que presentan no linealidades inherentes al propio dispositivo que producen efectos no deseados, como la distorsión armónica de la señal, distorsión de intermodulación, distorsión de fase, compresión de ganancia, interferencia con el canal adyacente, etc. Estos efectos reducen la eficiencia espectral de la transmisión, de modo que no solo causan distorsión en la banda de la señal de entrada, sino que también existe un recrecimiento fuera de la banda.

Como se ha comentado, los PA son empleados en la última etapa de un transmisor con el objetivo de producir una señal con una potencia adecuada para su transmisión mediante una antena. Para condiciones de pequeña señal, el amplificador de potencia se comporta como un dispositivo lineal, sin embargo, para los altos niveles de señal que interesan en un transmisor, se producen comportamientos no lineales. Concretamente, a partir de un cierto valor de potencia de señal de entrada, la potencia de salida deja de incrementarse, entrando el amplificador en un punto de saturación.

Los amplificadores de potencia presentan una mayor eficiencia cerca de este punto de saturación. Esto hace que, a pesar de trabajar en un punto con mejor eficiencia, se produzcan efectos de distorsión no lineal en la señal de salida, tanto en la propia banda, como en canales adyacentes. Debido a ello, siempre deberá existir un balance entre trabajar con una buena eficiencia o un nivel deseable de distorsión no lineal. Por este motivo, es de vital importancia conocer las características de los efectos y comportamientos de la distorsión no lineal [2].

2.1 Sistemas lineales y no lineales

Ya que a lo largo de este documento se trabajará con sistemas no lineales, es apropiado realizar una breve revisión de las propiedades de este tipo de sistemas. Los sistemas no lineales son aquellos que no cumplen las características de los sistemas lineales, por lo que será conveniente recordar las propiedades de un sistema lineal.

Como se explica en [3], “un sistema lineal es aquel que posee la propiedad de superposición, es decir, si una entrada consiste en la suma ponderada de varias señales, entonces la salida es la superposición o suma ponderada de las respuestas del sistema a cada una de estas señales”. Expresando este concepto con notación matemática en tiempo continuo, sea $y_1(t)$ la respuesta del sistema a una entrada $x_1(t)$, sea $y_2(t)$ la respuesta del sistema a una entrada $x_2(t)$, y sean a y b constantes complejas cualesquiera, un sistema lineal puede definirse como aquel que cumple:

$$ax_1(t) + bx_2(t) \rightarrow ay_1(t) + by_2(t) \quad (2.1)$$

Por tanto, cualquier sistema que no cumpla la propiedad de superposición descrita por la Ecuación (2.1), se tratará de un sistema no lineal.

Volviendo a los amplificadores de potencia, es importante aclarar que, aunque en las etapas académicas se suelen introducir como un sistema lineal, estos no suelen serlo. Para demostrar esto de forma sencilla, se pueden definir cuatro términos distintos de potencia en torno a un PA: las potencias de las señales de entrada P_{in} y de salida P_{out} , la potencia DC entregada al PA como alimentación P_{dc} , y la potencia disipada total P_{dis} .

Definiendo la ganancia del amplificador como la relación entre la potencia de salida y de entrada, y teniendo en cuenta la conservación de la energía, se podrían definir la Ecuación (2.2) y la Ecuación (2.3), respectivamente.

$$G_P = \frac{P_{out}}{P_{in}} \quad (2.2)$$

$$P_{out} + P_{dis} = P_{dc} + P_{in} \quad (2.3)$$

Sustituyendo la segunda ecuación en la primera, la ganancia del PA quedaría finalmente como se indica en la Ecuación (2.4). Analizando esta última, se observa claramente que el PA no podrá mantener una ganancia constante a pesar de que la potencia de entrada sea cada vez mayor, puesto que P_{dc} está limitada por la potencia disponible de la fuente.

$$G_P = 1 + \frac{P_{dc} - P_{dis}}{P_{in}} \quad (2.4)$$

2.2 Distorsión no lineal

La distorsión no lineal es una forma de perturbación de señal originada por las no linealidades de un sistema. Una perturbación de señal puede deberse a la adición de nuevas componentes a la señal o a la modificación de las propias características de la señal original [4]. En el primer caso, se pueden incluir componentes de ruido aleatorio aditivo y las interferencias deterministas aditivas. El otro conjunto de perturbaciones englobaría la distorsión de señal que consiste en una modificación de señal, estando asociada a esta, por lo que no pueden ser detectadas cuando no se dispone de la fuente de señal.

Además, la distorsión puede ser lineal, como un cambio de escala de la señal, lo cual no suele ser de gran relevancia en un sistema de telecomunicaciones; y no lineal, que puede producir un mayor número de modificaciones a la señal como, por ejemplo, generar armónicos o componentes totalmente incorreladas con la señal original, que sí son más perjudiciales.

Normalmente, las señales con las que se trabajan en telecomunicaciones son sinusoidales, con una amplitud y una fase variables, como se muestra en la Ecuación (2.5).

$$x(t) = A(t) \cos [2\pi f_c t + \theta(t)] \quad (2.5)$$

En algunos casos, la respuesta de un sistema lineal a la señal $x(t)$ se podría representar como en la Ecuación (2.6), donde se aprecia un término que modifica la amplitud, un retraso y un desfase.

$$y_L(t) = a_1 A(t - \tau_1) \cos [2\pi f_c t + \theta(t - \tau_1) - \phi_1] \quad (2.6)$$

Por otro lado, con esta señal como entrada a un sistema no lineal, se podría representar la salida como en la Ecuación (2.7), donde se ha truncado en el tercer orden.

$$y_{NL}(t) = a_1 x(t - \tau_1) + a_2 x(t - \tau_2)^2 + a_3 x(t - \tau_3)^3 + \dots \quad (2.7)$$

Además, esta se podría extender como sigue:

$$\begin{aligned} y_{NL}(t) = & a_1 A(t - \tau_1) \cos [2\pi f_c t + \theta(t - \tau_1) - \phi_1] \\ & + a_2 A(t - \tau_2)^2 \cos [2\pi f_c t + \theta(t - \tau_2) - \phi_2]^2 \\ & + a_3 A(t - \tau_3)^3 \cos [2\pi f_c t + \theta(t - \tau_3) - \phi_3]^3 \end{aligned} \quad (2.8)$$

Como se puede comprobar, en el caso no lineal se obtiene un número mayor de términos, con orden cada vez mayor. Estos se podrían agrupar, de modo que se visualizaría más claramente que estos términos serían equivalentes a cosenos de frecuencias múltiplos de la fundamental, es decir, $2 \cdot 2\pi f_c$, $3 \cdot 2\pi f_c$, etc. Por tanto,

estos términos adicionales introducen nuevas componentes espectrales y, por consiguiente, producen un recrecimiento espectral. De este modo, como se dice en [4], los sistemas no lineales pueden modificar el espectro de señal, tanto eliminando como generando componentes frecuenciales.

2.3 Caracterización de amplificadores

Los amplificadores de potencia y, en general, los dispositivos electrónicos, pueden ser caracterizados por una serie de parámetros o figuras de mérito. En el caso de los PA, la más conocida se trata de la ganancia de potencia, que ofrece una relación entre la potencia de la señal de salida y la de entrada.

Sin embargo, existen algunas métricas que también son muy útiles y que se usarán a lo largo del presente trabajo para medir las no linealidades del amplificador.

2.3.1 Figuras de mérito

Punto de compresión de 1 dB

Como se ha comentado anteriormente, cuando la potencia de la señal de entrada supera un cierto valor, la ganancia del amplificador empieza a decrecer. De hecho, a partir del punto de saturación, el incremento de la potencia de salida puede llegar a ser inapreciable a pesar de una mayor potencia de entrada. Por este motivo, suele ser muy útil conocer el nivel de potencia en el cual empieza a producirse esta compresión de la ganancia. El punto de compresión de 1 dB (P_{1dB}) es una medida que define el punto de potencia de entrada en el que la ganancia cae un decibelio respecto a la ganancia lineal. En la Figura 2.1 se representa una medida real donde se ha indicado el punto de compresión de 1 dB. En esta figura se muestra también, en línea de puntos la respuesta ideal lineal, así como una línea discontinua que representa el punto de saturación del amplificador en cuestión, donde el aumento de la potencia de salida sería prácticamente nulo.

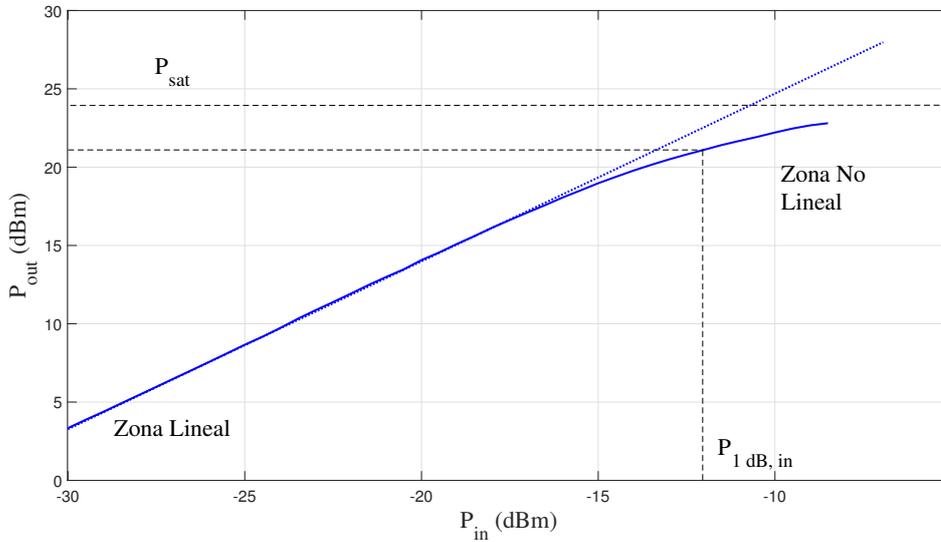


Figura 2.1 Punto de compresión de 1 dB.

Relación de pico a valor medio (PAPR)

La relación de pico a valor medio de una señal $x(t)$, también conocida en inglés como *Peak to Average Power Ratio* (PAPR), se define como la relación entre el máximo instantáneo de la potencia de dicha señal y el valor promedio de su potencia, como se muestra en la siguiente ecuación:

$$PAPR[x(t)] = \frac{\max_{0 \leq t \leq NT} [|x(t)|^2]}{P_{av}} \tag{2.9}$$

donde N es el número de símbolos, T es el período de símbolo y P_{av} es la potencia media de la señal $x(t)$.

La PAPR de una señal podrá determinar cuánto afectará la amplificación no lineal a la señal transmitida, puesto que si este parámetro es demasiado elevado en la señal de entrada, el amplificador puede entrar en regiones no lineales para determinados puntos de la señal.

Concretamente, este parámetro es ampliamente utilizado para medir señales de sistemas OFDM para tener en cuenta los rangos de potencia de las señales que se van a introducir en un amplificador ya que, al presentar valores altos de PAPR, el PA puede producir diferente ganancia para diferentes instantes de la señal y, por tanto, generar distorsión. Debido a que este tipo de sistemas son muy usados en la actualidad, en este trabajo se tratará con señales OFDM, por lo que este parámetro tendrá cierta relevancia a la hora de medir las no linealidades del amplificador en diferentes experimentos.

Relación de intermodulación

Algunas figuras de mérito se miden poniendo a la entrada del amplificador dos tonos, es decir, dos señales de diferente frecuencia. Al introducir dos o más tonos, un amplificador no lineal genera componentes de intermodulación, destacando los de tercer orden (IM3). Para medir los efectos producidos por esta intermodulación, se determina el *Intermodulation Ratio* (IMR), el cual se define como la relación entre la potencia de la componente fundamental y la salida de la potencia del *Intermodulation Distortion* (IMD), es decir, de las componentes que se originan debido a la intermodulación de dos o más tonos. Además, se puede medir el IMR_L inferior y el IMR_U superior, tomando de referencia a la frecuencia fundamental. Es importante tener en cuenta esta distinción, puesto que en algunas ocasiones los dispositivos pueden presentar asimetrías y, por tanto, estos valores serán diferentes.

Para el caso de las componentes de tercer orden, los valores de IMR_L y de IMR_U se calcularían como:

$$IMR_L = \frac{P_{fund}}{P_{IMD}} = \frac{P(f_1)}{P(2f_1 - f_2)} \quad (2.10)$$

$$IMR_U = \frac{P_{fund}}{P_{IMD}} = \frac{P(f_2)}{P(2f_2 - f_1)} \quad (2.11)$$

Relación de potencia de canal adyacente (ACPR)

Mientras se amplifica el canal de la señal introducida al PA, una parte de la potencia también se transfiere a los canales adyacentes debido a la distorsión de intermodulación. El ensanchado o recrecimiento espectral que se produce debido a la amplificación no lineal también se conoce como interferencia de canal adyacente o *Adjacent-Channel Interference* (ACI) en los sistemas de comunicaciones inalámbricas. Estas componentes frecuenciales no deseadas representarán una importante fuente de interferencia para sistemas que trabajen en dichas bandas.

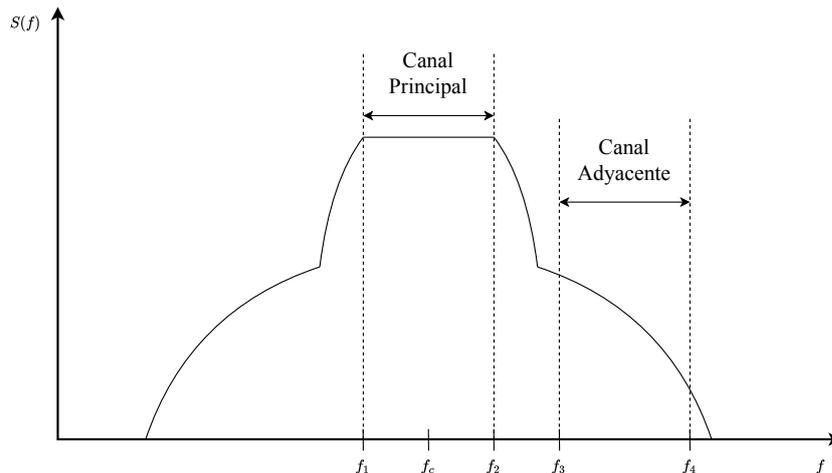


Figura 2.2 Definición de ACPR.

Como se comprobará más adelante, a lo largo de este trabajo será de vital importancia conocer el nivel de recrecimiento de los canales más cercanos a la frecuencia central. El parámetro conocido como *Adjacent-Channel Power Ratio* (ACPR) caracteriza el recrecimiento espectral producido por la no linealidad del PA. Este se define como la relación entre la potencia de salida en el canal fundamental y la potencia transferida

a los canales adyacentes a este, de modo que la ecuación se expresa en términos de densidad espectral de potencia o, en inglés, *Power Spectral Density* (PSD), como se indica a continuación:

$$ACPR = \frac{\int_{f_1}^{f_2} S_{yy}(f)df}{\int_{f_3}^{f_4} S_{yy}(f)df} \quad (2.12)$$

donde las frecuencias f_1 y f_2 son los límites de la canal principal y las frecuencias f_3 y f_4 representan los límites del canal adyacente, como se representa en la Figura 2.2.

Del mismo modo que ocurre con otras figuras de mérito, es habitual definir valores de ACPR tanto para canales superiores como inferiores, así como con canales que no sean estrictamente adyacentes. Concretamente, en este trabajo se tendrán en cuenta cuatro valores de ACPR, los correspondientes a los canales superior e inferior inmediatamente adyacentes al canal principal, y los canales superior e inferior contiguos a estos.

Magnitud del vector error (EVM)

Este parámetro, comúnmente conocido por su nombre en inglés, *Error Vector Magnitude* (EVM), representa una forma de cuantizar la distorsión no lineal de un sistema, existiendo muchos estándares de comunicaciones especifican un valor máximo [5]. El EVM consiste en medir la desviación de la constelación de señal respecto a la del caso ideal en un sistema de comunicaciones digitales y puede definirse como la distancia entre el vector de señal deseado y los realmente obtenidos, normalizando respecto a la amplitud de la señal.

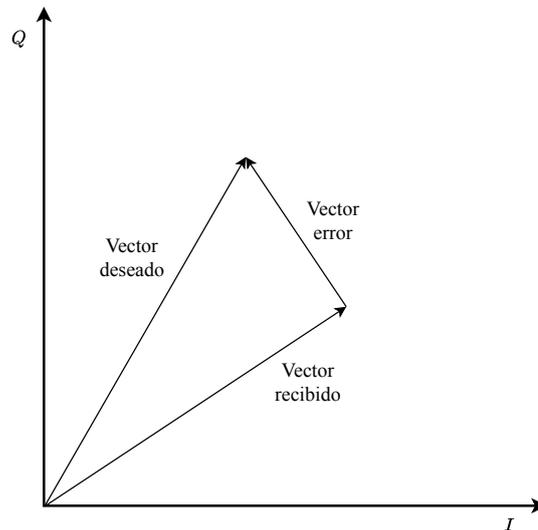


Figura 2.3 Definición de EVM.

El EVM cuantifica la distorsión dentro de la propia banda que implica errores en la recepción de los bits del conjunto de datos transmitidos. Por tanto, se deben cumplir las especificaciones de EVM para conseguir una comunicación adecuada.

En la Ecuación (2.13) se define matemáticamente el EVM, según [6], donde $z(n)$ sería la señal recibida, mientras que $x(n)$ sería la señal transmitida o de referencia.

$$EVM = \sqrt{\frac{E[|z(n) - x(n)|^2]}{E[|x(n)|^2]}} \quad (2.13)$$

Error cuadrático medio normalizado (NMSE)

Como se explicará en el siguiente capítulo, existen diversas formas para modelar matemáticamente el comportamiento de un amplificador. Además, muchos modelos suelen presentar términos polinómicos de alto orden y, aunque el error de modelado suele tender a ser menor en estos casos, nunca suele ser nulo.

Por este motivo, surge la necesidad de obtener una medida cuantitativa de la precisión de la aproximación de un modelo y, como se indica en [7], se define el error cuadrático medio normalizado o, en inglés, *Normalized Mean Squared Error* (NMSE). Concretamente, este parámetro se utiliza tanto para obtener la precisión (o el

error) de un modelo, como el parecido entre una señal original o deseada $y(t)$ y una estimada $\hat{y}(t)$, como se muestra en la siguiente ecuación

$$NMSE = 10 \log_{10} \left[\frac{\sum_{n=1}^N |y(t_n) - \hat{y}(t_n)|^2}{\sum_{n=1}^N |y(t_n)|^2} \right] \quad (2.14)$$

donde t_n indicaría el instante de tiempo o la muestra de la señal, y el valor de $NMSE$ estaría expresado en decibelios.

2.3.2 Características AM-AM y AM-PM

Además de las figuras de mérito, una forma de caracterizar un amplificador es obtener su característica *Amplitude Modulation - Amplitude Modulation* (AM-AM). Esta puede ser representada mostrando la ganancia instantánea o la potencia instantánea de la señal de salida frente a la potencia de entrada para una señal modulada. Un ejemplo del segundo caso se muestra en la Figura 2.4.

Normalmente, la característica AM-AM muestra que la ganancia del PA presenta un aumento constante, es decir, una ganancia lineal, al incrementar la potencia de entrada, hasta llegar a un punto de saturación en el que la ganancia empieza a descender, representándose comúnmente como una desviación expresada en dB/dB. Como se comenta en [5], esta saturación es una manifestación de la no linealidad del amplificador, donde la potencia de salida no sigue la potencia de entrada con una ganancia constante.

Esta característica también permite la evaluación de una figura de mérito muy importante, el punto de compresión de un decibelio P_{1dB} , el cual se definió en la sección anterior.

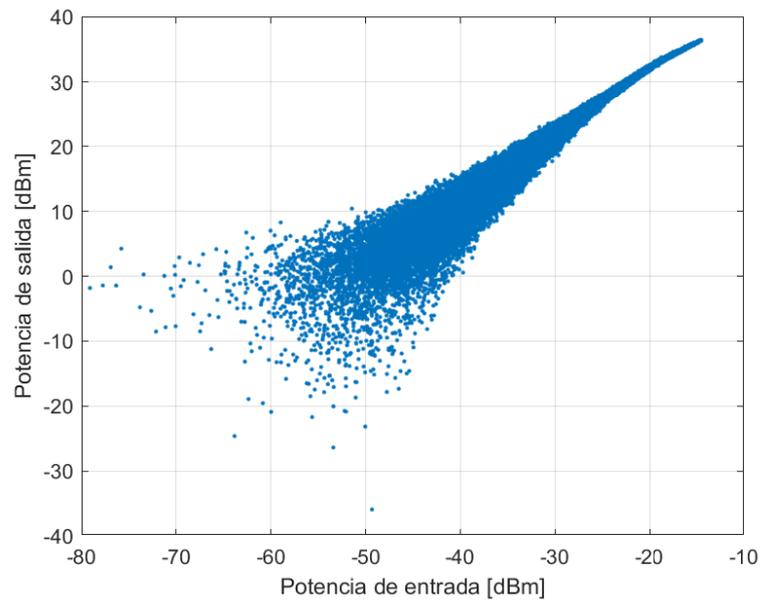


Figura 2.4 Ejemplo de característica AM-AM.

Otra manifestación de no linealidad se da en la fase del amplificador, de modo que la fase de la señal de salida presenta una desviación respecto a la fase de la señal de entrada, representada en grados/dB, la cual dependerá de la potencia de la señal de entrada. Esta característica se conoce como *Amplitude Modulation - Phase Modulation* (AM-PM) y, concretamente, mide la distorsión de fase introducida por el amplificador.

Tanto la Figura 2.4 como la Figura 2.5 representan las características AM-AM y AM-PM, respectivamente, de un amplificador de potencia real, concretamente de uno de los laboratorios de microondas más importantes del mundo, el *Microwave Electronics Laboratory* de Chalmers. En la primera figura se puede apreciar que la ganancia del amplificador no es, en absoluto, lineal, mientras que la desviación de la fase sufre un efecto similar. Este efecto de puntos dispersos no se debe únicamente a la no linealidad del amplificador, sino a los efectos de memoria que en este se producen.

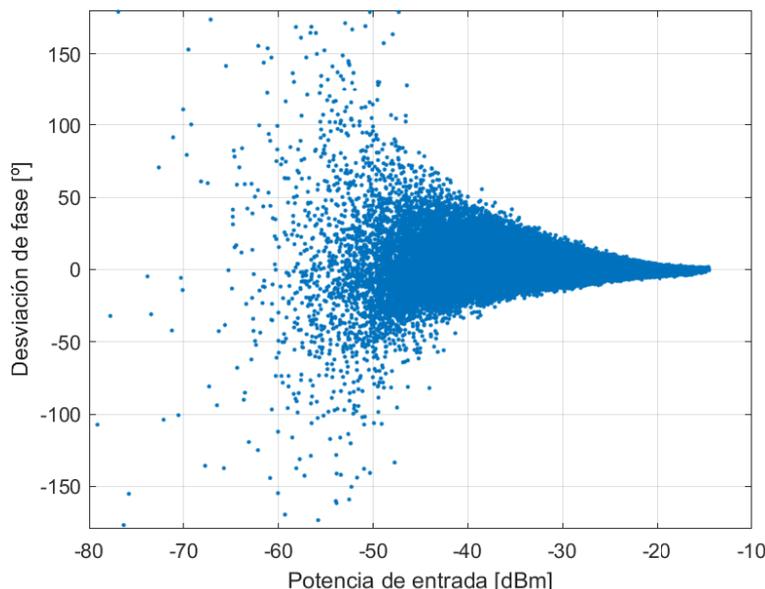


Figura 2.5 Ejemplo de característica AM-PM.

2.3.3 Efectos de memoria

Cuando se habla de los efectos de memoria producidos por algún tipo de dispositivo electrónico, se suele referir a la influencia de valores anteriores de una señal en el valor actual de otra señal de salida. De este modo, en el caso de los amplificadores de potencia, el valor de la señal de salida en un determinado instante no dependerá únicamente del valor de la señal de entrada en el mismo instante, sino que también influirán valores de instantes diferentes. Esto puede representarse en el tiempo discreto según la Ecuación (2.15), donde la señal de entrada sería $x[k]$ y la señal de salida del amplificador sería $y[k]$, la cual depende no solo del valor de la señal de entrada en el instante k , sino de versiones retrasadas de la misma.

$$y[k] = f(x[k], x[k-1], x[k-2], \dots, x[k-Q]) \quad (2.15)$$

Por otro lado, es habitual que la influencia de los efectos de memoria sean cada vez menores, es decir, el término de mayor influencia será el del mismo instante de tiempo, el del instante anterior tendrá una menor repercusión y así sucesivamente hasta que un cierto término tenga una influencia despreciable en la señal de salida. Este fenómeno es denominado *fading memory* o “memoria desvaneciente”, y el número de muestras Q que tienen repercusión en un instante k es llamado *memory depth* o “profundidad de la memoria”.

El hecho de que un amplificador de potencia, u otra clase de dispositivo electrónico, presente memoria es consecuencia directa de los componentes que forman su arquitectura. Estos sistemas suelen estar compuestos por elementos de electrónica básica, como pueden ser las bobinas o los condensadores, los cuales también presentan este tipo de efectos ya que, por ejemplo, un condensador ofrecerá una determinada tensión en función de la carga que haya tomado anteriormente.

Por otro lado, según [8], los efectos de memoria pueden ser clasificados en función del grado de influencia que tengan en comparación con el periodo de la señal:

- **Efectos de memoria a corto plazo:** en el caso de un amplificador, este tipo de efectos son debidos a las reactancias asociadas al transistor, así como a las redes de adaptación utilizadas, las cuales suelen incluir componentes reactivos o líneas de transmisión.
- **Efectos de memoria a largo plazo:** estos efectos son consecuencia de los efectos térmicos presentes para frecuencias menores a 100 kHz, efectos y variaciones producidos por imperfecciones de los semiconductores que componen el dispositivo y las redes de polarización que presentan baja impedancia para corrientes DC pero a su vez alta impedancia para señales de RF.

Además, si las constantes de tiempo de la memoria de un amplificador son menores que el máximo de la envolvente de la señal, el sistema se denomina “cuasi-sin-memoria” [5].

Finalmente, es importante comentar que si la característica AM-PM de un PA no es plana, este presenta comportamientos de memoria. Esto se debe a que, generalmente, los sistemas no lineales sin memoria solo causan distorsión de amplitud y nunca de fase, por lo que si un sistema presenta distorsión de fase, representada por la característica AM-PM, significa que también presentará efectos de memoria.

La forma más común de modelar los efectos de memoria de los amplificadores no lineales consisten en emplear análisis en series de Volterra, las cuales serán estudiadas en mayor detalle en el Capítulo 3.

3 Modelado del comportamiento de los amplificadores

Two things are infinite: the universe and human stupidity; and I'm not sure about the universe.

ALBERT EINSTEIN

Como se ha visto en la sección anterior, un circuito lineal produce una salida a partir de una entrada sin introducir distorsión. Sin embargo, los sistemas no lineales y, concretamente, los amplificadores de potencia, en determinados puntos de trabajo, pueden introducir distorsión. Este fenómeno no lineal no se trata de un simple cambio de la forma de onda de la señal en el dominio del tiempo, sino que también produce la adición de componentes frecuenciales que pueden ser perjudiciales para otros sistemas de comunicaciones [6]. Es por este motivo el hecho de que el modelado del comportamiento y la simulación de sistemas no lineales tengan un papel muy importante en la evaluación de un sistema de comunicaciones.

Para el modelado de amplificadores no lineales es común usar las características AM-AM y AM-PM, obtenidas a partir de una medida de un único tono. Con ello, se ajusta una función polinómica a los datos medidos, de modo que se obtenga un modelo que sea capaz de realizar predicciones de la distorsión no lineal que el sistema de comunicaciones pueda introducir en simulaciones. Sin embargo, las medidas de un único tono son demasiado simples como para poder ser empleadas en amplificadores no lineales de banda ancha con efectos de memoria, debido a la dependencia frecuencial de sus características [5].

Existen varios métodos para realizar el análisis de circuitos no lineales, lo cuales se pueden englobar en tres grupos, según [4]. En primer lugar, se podría incluir a los basados en la integración por paso temporal o *time-step*. A pesar de que este tipo de métodos no son tan adecuados para la predicción de distorsión multitono como otros tipos, fue el primero en ser introducido debido a que ofrecen una interpretación más sencilla, tratando de resolver las ecuaciones diferenciales no lineales obtenidas tras aplicar leyes de Kirchhoff.

Por otro lado, estaría el grupo compuesto por las técnicas de balance armónico. Este método se utiliza actualmente como método de simulación en el ámbito de las RF y las microondas, puesto que incluye el comportamiento del circuito, su excitación y su salida en dominio frecuencial.

Por último, pero no menos importante, se encontrarían los métodos de análisis en series de Volterra. Estos son los más comúnmente usados para modelar el comportamiento de sistemas no lineales y consisten en un procedimiento analítico que permite describir la respuesta de los sistemas mediante un conjunto de términos y operaciones matemáticas limitado. Puesto que este método permite representar sistemas haciendo uso de un conjunto de operadores no lineales, independientemente de la señal de excitación, puede aplicarse no solo al análisis, sino también al diseño de sistemas, por lo que cuenta con una gran importancia en el estudio de los efectos de la distorsión.

Sin embargo, el desarrollo de los *kernels* de Volterra característicos de este método se vuelve demasiado complicado a partir del tercer orden, y su implementación se hace ineficiente [5]. Debido a este inconveniente, una gran variedad de modelos han sido desarrollados para paliar la complejidad computacional y, aunque se basan en casos específicos de las series de Volterra, ofrecen una gran simplificación en términos de extracción de parámetros y de implementación del modelo en software.

Por tanto, en este capítulo se realizará una revisión del análisis de series de Volterra, puesto que es un método bastante usado en el estudio de la distorsión producida por amplificadores de potencia. Tras esto, se introducirán algunos de los modelos que parten del modelo de Volterra original, así como algunas de las técnicas utilizadas en la realización de experimentos para este trabajo.

3.1 Series de Volterra

Vito Volterra fue un matemático italiano que tuvo un papel muy importante en una rama de las matemáticas, el análisis funcional, donde se define el concepto de funcional como una función que depende de un conjunto de valores de otra función. Posteriormente, Norbert Wiener, profesor en el *Massachusetts Institute of Technology* (MIT), introdujo el uso de la teoría de funcionales de Volterra en el modelado de sistemas dinámicos no lineales e invariantes en el tiempo.

Muchos de los modelos analíticos de no linealidades se basan en el análisis en series de Volterra y la teoría que desarrolló Wiener para modelar sistemas no lineales con memoria. En el análisis en series de Volterra general, un modelo no lineal suele consistir en filtros (lineales o no) con un ancho de banda finito que determinan la selectividad en frecuencia del sistema. En el dominio del tiempo, estos filtros son representados por funciones denominadas *kernels*, que serán vistas a continuación y cuyo soporte representa la memoria en el sistema.

Actualmente, las técnicas de Volterra han vuelto a tomar presencia debido a la significativa preocupación sobre la distorsión en circuitos de comunicaciones modernos, con dispositivos de alta potencia y alta frecuencia. Adicionalmente, los efectos de memoria en los PA en estos sistemas son un gran problema, y las técnicas de series de Volterra han sido ampliamente aplicadas al modelado del comportamiento de los amplificadores de RF, y a su linealización usando técnicas de predistorsión [6].

Existen diferentes variantes del modelo general de Volterra, incluyendo modelos sin memoria. A continuación, se realizará una revisión de cómo se aplica la expansión en series de Volterra para un sistema dinámico no lineal, como podría ser un amplificador de RF, el cual debe ser débilmente no lineal para que sea posible aplicar análisis de Volterra, es decir, la respuesta del sistema debe ser continua y, por tanto, puede ser representada por una serie finita de términos.

3.1.1 Modelo general de Series de Volterra

Un modelo general de series de Volterra para un sistema no lineal se muestra en la Ecuación (3.1) como una expansión funcional de funciones continuas.

$$y(t) = \sum_{n=1}^{\infty} F_n(x(t)) = \sum_{n=0}^{\infty} y_n(t) \quad (3.1)$$

La función $F_n(x(t))$ se trata de la funcional de Volterra y se define como

$$F_n(x(t)) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(t; \tau_1, \tau_2, \dots, \tau_n) \prod_{i=1}^n x(\tau_i) d\tau_i \quad (3.2)$$

donde el término $h_n(\tau_1, \dots, \tau_n)$ se denomina *kernel* de Volterra n -dimensional o respuesta al impulso no lineal de orden n . Este determina la memoria del sistema, ya que define el intervalo de tiempo en el que las entradas pueden tener repercusión en la salida en el instante actual.

Si el sistema no lineal es causal e invariante en el tiempo, este puede ser expresado como la convolución de las potencias de la señal de entrada $x(t)$ con los *kernels* n -dimensionales:

$$\begin{aligned} F_n(x(t)) &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \dots, \tau_n) \prod_{i=1}^n x(t - \tau_i) d\tau_i \\ &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(t - \tau_1, \dots, t - \tau_n) \prod_{i=1}^n x(\tau_i) d\tau_i \end{aligned} \quad (3.3)$$

3.1.2 Series de Volterra como desarrollo de Series de Taylor

Como se demuestra en [6], las series de Volterra pueden expresarse a partir de un modelo no lineal instantáneo (sin memoria) descrito en expansión de series de Taylor, como el que se indica en la Ecuación (3.4). Los

coeficientes polinómicos a_n de esta expresión pueden calcularse como la relación de entrada y salida de las series de Taylor en torno al punto de operación x_0 , de modo que se obtendría la Ecuación (3.5).

$$y(t) = a_0 + \sum_{n=1}^N a_n x^n(t) \quad (3.4)$$

$$\begin{aligned} y(x(t)) &= y(x)|_{x=x_0} + \frac{1}{2!} \frac{dy}{dx} \Big|_{x=x_0} (x-x_0) + \frac{1}{3!} \frac{d^2y}{dx^2} \Big|_{x=x_0} (x-x_0)^2 + \dots \\ &= a_0 + a_1 x + a_2 x^2 + \dots \end{aligned} \quad (3.5)$$

Tras esto, si se considera la salida del sistema $y(t)$, que podría ser en este caso un amplificador, como una función que depende de la entrada $x(t)$ y de sus valores anteriores, podría expresarse como:

$$y(t) = f(x, x_1, x_2, \dots, x_n) \quad (3.6)$$

donde

$$x = x(t), \quad x_1 = x(t - \tau_1), \quad x_2 = x(t - \tau_2), \quad \dots, \quad x_n = x(t - \tau_n) \quad (3.7)$$

De este modo, se puede expresar esta función como expansión en series de Taylor, teniendo como resultado la Ecuación (3.8), la cual sería una serie multinomial.

$$\begin{aligned} y(x(t)) &= y(x)|_{x=x_0} + \frac{1}{2!} \frac{dy}{dx} \Big|_{x=x_0} (x-x_0) + \frac{1}{2!} \frac{dy}{dx_1} \Big|_{x=x_0} (x_1-x_0) + \dots \\ &+ \frac{1}{3!} \frac{d^2y}{dx^2} \Big|_{x=x_0} (x-x_0)^2 + \frac{1}{3!} \frac{d^2y}{dx_1^2} \Big|_{x=x_0} (x_1-x_0)^2 + \dots \\ &+ \frac{1}{3!} \frac{d^2y}{dx dx_1} \Big|_{x=x_0} (x-x_0)(x_1-x_0) + \dots \end{aligned} \quad (3.8)$$

En esta expresión, se pueden apreciar los términos de memoria $(x_1 - x_0)$ y los denominados términos cruzados de Volterra $(x - x_0)(x_1 - x_0)$.

Por tanto, se ha comprobado que las series de Volterra pueden describirse como un caso de series de Taylor con términos que tienen en cuenta la memoria de un sistema. Por otro lado, la convergencia de este modelo puede garantizarse para un número limitado de términos, es decir, las series pueden aproximarse al valor de la función original truncando en un cierto punto de la expresión, de modo que se cumpla una cierta tolerancia o un error mínimo respecto al valor verdadero de la función.

3.1.3 Series de Volterra como desarrollo de un sistema lineal

En la Sección 2.1 se hizo una revisión de cómo se define si un sistema es lineal o no. La salida de un sistema dinámico lineal puede expresarse como una convolución de su respuesta al impulso $h(t)$ y la señal de entrada $x(t)$:

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(t - \tau) x(\tau) d\tau \quad (3.9)$$

De nuevo, si se plantea un modelo dinámico no lineal en el que se tengan en cuenta los términos hasta el orden n y se parte de la definición de un sistema lineal en forma de convolución definida en la Ecuación (3.9), se puede expresar la salida de la siguiente forma:

$$y(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) x(t - \tau_1) x(t - \tau_2) \dots x(t - \tau_n) d\tau_1 d\tau_2 \dots d\tau_n \quad (3.10)$$

Esta expresión resulta útil para comprender a los *kernels* de Volterra $h_n(\tau_1, \tau_2, \dots, \tau_n)$ como los coeficientes de cada uno de los componentes de la expresión del modelo no lineal y, que la convolución que representa un sistema lineal se corresponde con un caso particular de la serie de Volterra.

3.1.4 Series de Volterra en tiempo discreto

Finalmente, es importante destacar que, en la práctica, la aproximación de la serie se mantiene solo en intervalos de tiempo finitos donde la señal de entrada es distinta de cero. Además, es muy habitual trabajar con muestras discretas debido a las limitaciones de los sistemas de computación por lo que tanto la señal como la expresión de la serie de Volterra se suelen expresar en tiempo discreto:

$$y[k] = \sum_{n=1}^N y_n[k] = \sum_{n=1}^N \sum_{m_1=0}^M \sum_{m_2=0}^M \cdots \sum_{m_n=0}^M h_n(m_1, m_2, \dots, m_n) \prod_{i=1}^n x[k - m_i] \quad (3.11)$$

siendo este un modelo con memoria de longitud $M + 1$.

3.2 Modelos basados en series de Volterra

En la sección anterior se ha realizado una revisión del modelo general del análisis en series de Volterra. Sin embargo, se ha comentado que existen varios modelos que se basan en el modelo general y aplican una serie de modificaciones, teniendo en cuenta una serie de consideraciones.

En esta sección, se revisarán algunos de los modelos que se basan en las series de Volterra que serán empleados posteriormente para la realización de experimentos. No obstante, es conveniente realizar alguna aclaración. En primer lugar, la teoría general de series de Volterra puede ser particularizada para señales de ancho de banda pequeño con respecto a su frecuencia central f_0 . Por ello, como las componentes cercanas a dicha frecuencia suelen ser las de mayor interés, una señal real $x(t)$ se suele representar como:

$$x(t) = \text{Re}\{e^{j2\pi f_0 t} \tilde{x}(t)\} = \frac{e^{j2\pi f_0 t} \tilde{x}(t) + e^{-j2\pi f_0 t} \tilde{x}^*(t)}{2} \quad (3.12)$$

donde $\tilde{x}(t)$ se corresponde con la representación compleja en banda base de la señal $x(t)$, también conocida como envolvente compleja de la señal. Sin embargo, para facilitar la lectura de este documento, de ahora en adelante se expresará la señal en banda base como $x(t)$.

Más concretamente, se trabajará con expresiones en tiempo discreto, con vistas a la hora de poner en práctica algunos de los modelos más adelante, por lo que se expresarán las ecuaciones en función de $x[k]$.

3.2.1 Modelo completo de Volterra (FV)

El primer modelo que se va a destacar se trata del modelo completo de Volterra, conocido en inglés como *Full-Volterra* (FV). Este consiste en implementar directamente las series de Volterra en función de la envolvente compleja [9]. La expresión de este modelo se recoge en la Ecuación (3.13), donde se debe destacar que los posibles regresores presentan siempre un orden impar y, como máximo, de $2N - 1$.

$$y_{FV}[k] = \sum_{n=1}^N \sum_{m_1=0}^M \sum_{m_2=m_1}^M \cdots \sum_{m_n=m_{n-1}}^M \sum_{m_{n+1}=0}^M \sum_{m_{n+2}=m_{n+1}}^M \cdots \quad (3.13)$$

$$\cdots \sum_{m_{2n-1}=m_{2n-2}}^M h_{2n-1}(m_1, \dots, m_{2n-1}) \times \prod_{i=1}^n x[k - m_i] \prod_{j=n+1}^{2n-1} x^*[k - m_j]$$

El hecho de que el orden sea impar es debido al propio desarrollo del modelo FV en términos de la envolvente compleja ya que, para que la frecuencia fundamental de la entrada y la salida coincidan, si en cada producto hay $2n - 1$ factores de x , $n - 1$ de los mismos deberán estar conjugados, motivo por el que el productorio tenga la mitad de términos [8].

3.2.2 Modelo polinomial con memoria (MP)

El modelo que se usará para la realización de los experimentos, explicados más adelante, se basa en el modelo polinomial con memoria (MP), por lo que será conveniente, en primer lugar, realizar una breve revisión de este.

El modelo MP consiste en una generalización del modelo de Hammerstein que, combinando diferentes filtros y usando coeficientes de series de potencia a_{nm} , puede ser expresado de la siguiente forma:

$$y_{GH}[k] = \sum_{n=1}^N \sum_{m=0}^{M-1} a_{nm} x^n[k-m] \quad (3.14)$$

Partiendo de la Ecuación (3.14), se puede particularizar para el caso de banda estrecha, de modo que si se seleccionan únicamente las combinaciones de la forma $x[k]|x[k-m]|^{n-1}$, se obtiene el modelo polinomial con memoria:

$$y_{MP}[k] = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} a_{nm} x[k-m] |x[k-m]|^n \quad (3.15)$$

Este modelo incorpora los efectos de memoria simples al modelo estático, de forma que los términos de memoria cruzados no se incluyen, pero sí se tienen en cuenta los términos “diagonales”, que suelen ser relevantes en el modelado de amplificadores de potencia. De hecho, este es el principal motivo, por el que se ha demostrado que el modelo MP es efectivo en el campo de la predistorsión digital.

Sin embargo, como se verá a continuación, existe una forma de generalizar más la expresión de la Ecuación (3.15), teniendo en cuenta un mayor número de términos, pero sin ser un modelo excesivamente complejo.

3.2.3 Modelo polinomial con memoria generalizado (GMP)

El modelo *Generalized Memory Polynomial* (GMP) fue presentado en [10], tomando como base el modelo MP. Tras haber revisado este, resulta una tarea sencilla analizar el modelo GMP, puesto que este mejora el MP teniendo en cuenta algunos términos cruzados que puedan ser de mayor importancia, como son los de las diagonales superiores e inferiores. Así, se podría definir el modelo GMP, el cual tiene en cuenta estos términos y tendría la siguiente expresión:

$$\begin{aligned} y_{GMP}[k] = & \sum_{n=0}^{N_a} \sum_{l=0}^{L_a} a_{nl} x[k-l] |x[k-l]|^n \\ & + \sum_{n=1}^{N_b} \sum_{l=0}^{L_b} \sum_{m=1}^{M_b} b_{nlm} x[k-l] |x[k-l-m]|^n \\ & + \sum_{n=1}^{N_c} \sum_{l=0}^{L_c} \sum_{m=1}^{M_c} c_{nlm} x[k-l] |x[k-l+m]|^n \end{aligned} \quad (3.16)$$

donde los sumatorios con subíndices l y m se corresponden con los términos de memoria cruzados y los coeficientes b_{nlm} y c_{nlm} son los *kernels* correspondientes.

En esta ecuación, los valores N_a y L_a representan el número de coeficientes para la señal alineada con la envolvente, mientras que los valores N_b , L_b y M_b son el número de coeficientes para la señal con la envolvente retrasada, y los valores N_c , L_c y M_c con la envolvente adelantada respecto a la señal.

La principal ventaja de estos términos cruzados es que los coeficientes aparecen relacionados de forma lineal, por lo que pueden ser estimados de una forma simple y robusta usando algoritmos de tipo *Least Squares* (LS), lo que favorece la estabilidad del algoritmo y la complejidad computacional.

Además, es muy habitual no utilizar todos los coeficientes que se muestran en la Ecuación (3.16), de modo que se puede utilizar un número reducido de términos y conseguir buenos resultados. Por tanto, se puede observar que, con el objetivo de aprovechar aún más el potencial de la predistorsión digital en aplicaciones actuales, el modelo GMP combina el modelo polinomial con memoria con términos cruzados entre la señal y componentes adelantados o retrasados, consiguiendo una mayor precisión en el modelado, pero sin requerir un excesivo coste computacional. De hecho, tanto el GMP como el MP son modelos que contienen un número de coeficientes mucho más razonable que el del modelo FV, en el que puede llegar a haber un número tan elevado de coeficientes que sea un modelo excesivamente complejo e inmanejable. De ahí que este tipos de modelos sean soluciones muy empleadas actualmente.

Finalmente, con vistas a explicaciones posteriores, se definirá el término *regresor*. Este se referirá a cada una de las distintas versiones de la señal de entrada atrasadas o adelantadas, de forma que los *kernels* de la serie de Volterra serán multiplicados a esta señal con diferentes desplazamientos temporales.

3.3 Notación matricial

Tras haber realizado una revisión de algunos de los modelos de comportamientos de amplificadores, es conveniente indicar cómo se suele trabajar con los mismos. En comunicaciones, es muy habitual utilizar un modelo matricial, como el de la Ecuación (3.17).

$$\mathbf{y} = \mathbf{X} \cdot \mathbf{h} + \mathbf{e} \quad (3.17)$$

En esta expresión, \mathbf{y} es un vector columna de longitud M con los valores de la señal de salida en banda base, \mathbf{X} es una matriz de dimensiones $M \times N$ cuyas N columnas se corresponden con los N regresores del modelo y cuyas filas son las M muestras de los mismos, mientras que \mathbf{h} representaría un vector columna compuesto por los N coeficientes estimados del modelo. Finalmente, el vector columna \mathbf{e} se trata de un vector de error, puesto que los modelos no son perfectos y las medidas pueden presentar un cierto ruido.

Como bien se comenta en [8], en la Ecuación (3.17) se aprecia que las señales $y[k]$ y $x[k]$ se relacionan de manera no lineal, mientras que $y[k]$ sí se relaciona linealmente con los coeficientes del modelo. Partiendo de dicha expresión, se podría calcular los valores de los *kernels* del modelo minimizando la suma del error cuadrático residual (RSS), llegando a la expresión de la Ecuación (3.18), donde se obtendría la estimación de los coeficientes $\hat{\mathbf{h}}$.

$$\hat{\mathbf{h}} = (\mathbf{X}^H \mathbf{X})^{-1} \mathbf{X}^H \mathbf{y} = \mathbf{X}^\dagger \mathbf{y} \quad (3.18)$$

El operador H se corresponde con la hermítica de la matriz y la \mathbf{X}^\dagger denota la matriz pseudoinversa de Moore-Penrose de la matriz \mathbf{X} . La solución de la Ecuación (3.18) de donde se obtendría la estimación de los coeficientes del modelo se corresponde con la solución LS.

Tras haber presentado la notación matricial que será utilizada en los diferentes experimentos de este trabajo, se presentarán las técnicas de selección de componentes, que tendrán también cabida en algunas pruebas que serán explicadas en posteriores capítulos.

3.4 Selección de componentes: DOMP

Realizar operaciones con modelos basados en series de Volterra suele suponer un elevado coste computacional. Por este motivo, actualmente existen estudios e investigaciones sobre métodos para mitigar este inconveniente. Una forma de conseguir un coste computacional menor consiste en utilizar técnicas de reducción de componentes que, básicamente, tienen como objetivo reducir el número de términos o coeficientes en los modelos de comportamiento de los amplificadores.

Las técnicas de reducción de componentes se utilizan en muchos campos de estudio diferentes y, en la actualidad, existe una gran variedad de ellas, siendo la más conocida la denominada análisis de componentes principales (PCA). Por otro lado, existe un tipo denominado algoritmo voraz o *greedy*, en inglés, en el cual se centrará esta sección. Este tipo de algoritmos no realiza decisiones *a priori* en la selección de componentes, lo que los hace adecuados para aplicarlos a las series de Volterra.

Un tipo de algoritmo voraz es la técnica denominada *Orthogonal Matching Pursuit* (OMP), que consiste en elegir la componente con mayor correlación normalizada entre los regresores de la matriz \mathbf{X} y un vector de residuo \mathbf{r} , el cual es redefinido en cada iteración como la diferencia de la señal de salida \mathbf{y} y la señal de salida modelada $\hat{\mathbf{y}}$.

Concretamente, en los experimentos que serán explicados en próximos capítulos de este trabajo, se utilizará una técnica derivada del OMP, la denominada *Doubly Orthogonal Matching Pursuit* (DOMP), que fue presentada en [11] en el año 2018. Esta consiste en mejorar la selección de coeficientes de modo que se consiga una solución *k-sparse*, es decir, una solución con un determinado número k de coeficientes no nulos.

El DOMP parte de una expresión similar a la Ecuación (3.17) y tiene como objetivo calcular el vector de *kernels* de Volterra \mathbf{h} mediante un algoritmo iterativo siguiendo la minimización de norma l_1 , cuyo pseudocódigo se muestra en el Algoritmo 1.

El resultado del algoritmo será un conjunto de soporte S , cuyos elementos indican el índice de los regresores y estarán ordenados de mayor a menor relevancia respecto a la salida. Por tanto, el algoritmo se inicializará con un conjunto de soporte vacío, $S^{(0)} = \emptyset$, al cual se le irán añadiendo las componentes en el orden indicado. En [11] también se define una matriz $\mathbf{Z}^{(0)} = \mathbf{X}$, la cual es usada para conservar la información de los regresores ortogonalizados y que, tras la ejecución del algoritmo, deberá ser igual al resultado de aplicar el proceso de Gram-Schmidt a los regresores \mathbf{X} del conjunto de soporte final $S^{(final)}$. Por último, existirá un vector

residual que será inicializado con el valor de la señal de salida $\mathbf{r}^{(0)} = \mathbf{y}$, que será utilizado para conservar las componentes restantes de la misma tras seleccionar los regresores del modelo.

Algoritmo 1 Resumen del Algoritmo DOMP

Entrada: $\mathbf{X} \in \mathbb{C}^{m \times n}$, $\mathbf{y} \in \mathbb{C}^m$

Salida: $S^{(t)}$

- 1: *Inicialización* : $\mathbf{Z}^{(0)} = \mathbf{X}$, $\mathbf{r}^{(0)} \leftarrow \mathbf{y}$, $S^{(0)} \leftarrow \emptyset$
 - 2: **for** $t = 1$ hasta que el criterio de parada se cumpla **do**
 - 3: $\mathbf{Z}_{\{i\}}^{(t-1)} \leftarrow \frac{\mathbf{Z}_{\{i\}}^{(t-1)}}{\|\mathbf{Z}_{\{i\}}^{(t-1)}\|_2}$
 - 4: $i^{(t)} \leftarrow \arg \max_{i \notin S^{(t-1)}} \left| \mathbf{Z}_{\{i\}}^{(t-1)\text{H}} \mathbf{r}^{(t-1)} \right|$
 - 5: $S^{(t)} \leftarrow S^{(t-1)} \cup i^{(t)}$
 - 6: $\mathbf{p}^{(t)} \leftarrow \mathbf{Z}_{\{i\}}^{(t-1)\text{H}} \mathbf{Z}^{(t-1)}$
 - 7: $\mathbf{Z}^{(t)} \leftarrow \mathbf{Z}^{(t-1)} - \mathbf{p}^{(t)} \otimes \mathbf{Z}_{\{i^{(t)}\}}^{(t-1)}$
 - 8: $\hat{\mathbf{h}} \leftarrow \mathbf{X}_{S^{(t)}}^\dagger \mathbf{y}$
 - 9: $\hat{\mathbf{y}}^{(t)} \leftarrow \mathbf{X}_{S^{(t)}} \hat{\mathbf{h}}$
 - 10: $\mathbf{r}^{(t)} \leftarrow \mathbf{y} - \hat{\mathbf{y}}^{(t)}$
 - 11: **end for**
-

En cada iteración t , el algoritmo realiza una normalización de las componentes de la base dividiendo cada columna de la matriz \mathbf{Z} entre su norma l_2 , como se indica en la línea 3 del Algoritmo 1. A continuación, se selecciona la componente con la mayor resolución escalar normalizada en la dirección del vector residual. Además, como el residuo es siempre ortogonal al conjunto de base seleccionado $S^{(t-1)}$, la condición $i \notin S^{(t-1)}$ siempre se va a cumplir. Esta condición reduce además la complejidad computacional del algoritmo en una implementación real.

Una vez se ha seleccionado la componente, se incluye en el conjunto de soporte (línea 5) y se realiza el proceso de ortogonalización, obteniendo las proyecciones $\mathbf{p}^{(t)}$ del regresor seleccionado en cada uno de las componentes de la base. Luego, como se indica en la línea 7, donde \otimes denota el producto de Kronecker, a cada regresor se le resta su proyección correspondiente y, por tanto, la componente seleccionada queda ortogonal al conjunto de base restante.

Con el proceso de Gram-Schmidt se consigue una doble ortogonalización. En primer lugar, la base seleccionada es siempre ortogonal al residuo y, a su vez, también es ortogonal a los regresores que no pertenecen al conjunto de soporte seleccionado. Tras esto, se calcula la estimación del vector de *kernel*s de Volterra mediante la solución LS, y con este, se obtiene la estimación de la salida $\hat{\mathbf{y}}^{(t)}$ para, finalmente, actualizar el vector de residuo $\mathbf{r}^{(t)}$ calculando la diferencia entre la señal de salida estimada y la original.

Tal y como se detalla en [11], utilizando este algoritmo, se consigue un modelo con menos componentes que el modelo original o clásico pero se mantiene un buen comportamiento, gracias a una adecuada selección de componentes. Este aspecto, como se verá en próximos capítulos, será muy útil en los procedimientos posteriores puesto que, al tener un menor número de componentes, la complejidad computacional será menor.

Como conclusión de este capítulo, se han revisado algunas de las herramientas que se pueden utilizar para realizar el modelado de sistemas no lineales y, concretamente, los amplificadores de potencia. Entre estos métodos, han sido destacadas las series de Volterra, así como algunos que, basados en estas, consiguen mejorar en algunos aspectos los modelos resultantes. Con todo ello, es posible introducir en qué consiste y cómo se lleva a cabo la técnica de linealización conocida como predistorsión, la cual utilizará los modelos explicados en este capítulo para conseguir su objetivo.

4 Linealización de amplificadores y predistorsión digital

What we know is a drop, what we don't know is an ocean.

ISAAC NEWTON

En los capítulos anteriores se ha realizado un estudio de la no linealidad en los amplificadores de potencia, revisando algunos comportamientos como la saturación de su respuesta o la distorsión que estos introducen, así como la forma en que son medidas algunas de sus características. Además, se han estudiado algunos de los métodos más conocidos para modelar los amplificadores de potencia.

Una vez que el comportamiento de un amplificador es caracterizado, se puede ir más allá y aplicar técnicas que permitan trabajar con el amplificador en las zonas no lineales, paliando los inconvenientes que estas producen. En el presente capítulo, se presentarán algunas técnicas de linealización, prestando una atención especial a una de las técnicas más utilizadas en la actualidad, la predistorsión digital.

4.1 Linealización de un amplificador

Los sistemas de comunicaciones digitales modernos emplean señales diseñadas para conseguir una alta eficiencia espectral, como las señales OFDM, y estas presentan altos niveles de PAPR consecuentemente. Además, se requieren amplificadores de RF que cumplan ciertas condiciones como una elevada potencia de salida, alto nivel de ganancia, amplio ancho de banda de señal, baja distorsión y una alta eficiencia.

Este tipo de arquitecturas, sin embargo, presentan comportamientos no lineales. De hecho, los niveles de distorsión permitidos en los canales de la señal y la interferencia en los canales adyacentes se especifican claramente en estándares como los de la ITU-T o la 3GPP. Por este motivo, las técnicas de linealización son esenciales para cumplir dichas especificaciones.

Las técnicas de linealización son aquellas que se utilizan para cubrir y compensar los efectos de la distorsión no lineal introducida por los dispositivos no lineales en los transmisores inalámbricos, de modo que al aplicarlas se consiga una amplificación de la señal transmitida más lineal.

Como se ha visto en la sección anterior, las series de Volterra permiten el modelado de dispositivos no lineales. Sin embargo, el último objetivo de analizar el comportamiento de un amplificador no es únicamente modelar el propio sistema existente, sino emplear dichos modelos a la hora de diseñar técnicas de linealización.

Existen diferentes técnicas de linealización, incluyendo técnicas analógicas de realimentación *feedback* y *feedforward* [6]. Sin embargo, este trabajo se centrará en las técnicas de linealización llevadas a cabo mediante la modulación digital de señales en banda base, la llamada predistorsión digital o, en inglés, *Digital Predistortion* (DPD).

4.2 Predistorsión digital (DPD)

La predistorsión digital es una de las técnicas de linealización más extendidas y efectivas actualmente y se basa en añadir un bloque no lineal que ofrece las características inversas al dispositivo no lineal de modo que,

al combinarse, la salida sea lineal.

Los métodos de DPD han sido objeto de gran parte de la investigación recientemente, debido a la disponibilidad de dispositivos DSP más rápidos, con capacidades para obtener una mayor precisión a la hora de realizar las funciones de DPD.

Se pueden encontrar referencias [12] en las que se toma a la predistorsión digital como una técnica complementaria a dispositivos de linealización analógicos, de modo que se utiliza un DSP para controlar los ajustes de amplitud y fase, en un sistema que monitorice el rendimiento global del sistema al cambiar las señales y el entorno, con el objetivo de conseguir un incremento de la eficiencia y cumplir las necesidades de potencia. No obstante, en este trabajo se prestará una atención principal a los métodos de DPD con el objetivo de operar independientemente de otros tipos de dispositivos analógicos.

4.2.1 Diseño y funcionamiento de un DPD

Para diseñar un predistorsionador digital o DPD, el primer paso sería obtener un modelo del amplificador mediante series de Volterra y, a continuación, obtener su función inversa para establecer las características que deberá tener el DPD, obteniendo matemáticamente el modelo no lineal del sistema inverso, que puede ser desarrollado a partir de medidas de entrada y salida del dispositivo. Luego, el segundo paso sería considerar diferentes métodos para configurar un DPD con las características adecuadas.

Como se ha explicado anteriormente, un amplificador no es lineal para todo el rango de potencias de entrada y, cuando se supera cierto valor de la potencia de entrada, se comienza a producir una compresión de la ganancia, introduciendo también distorsión. Además, a partir de un determinado valor, el amplificador entra en zona de saturación, de forma que no es posible obtener más potencia del mismo. Por otro lado, también se ha comentado que el amplificador presenta una mejor eficiencia y potencia a niveles altos de señal, motivo por el cual la predistorsión pretende mejorar dicha zona de trabajo.

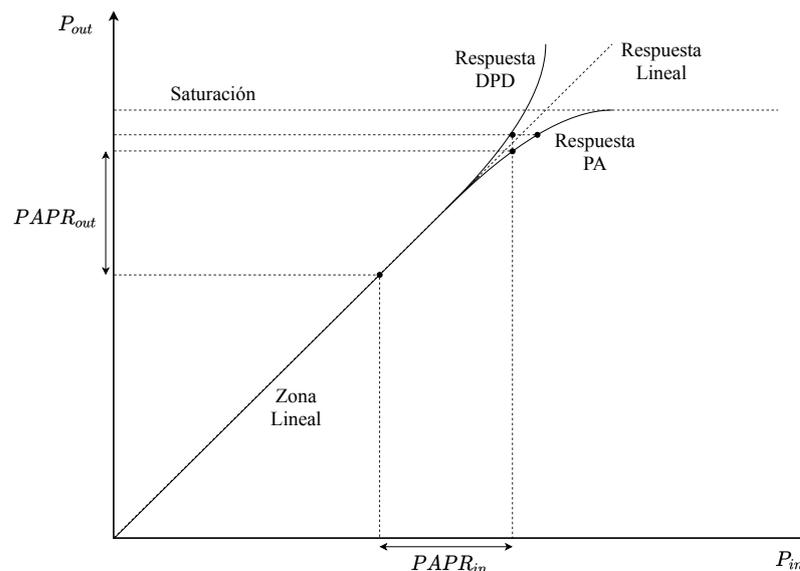


Figura 4.1 Característica entrada-salida de un amplificador y de un predistorsionador.

En la Figura 4.1, se muestra la respuesta de un amplificador con un punto de compresión determinado y la respuesta del predistorsionador adecuado. Como se puede comprobar, cuando la respuesta del amplificador tiende a comprimirse, la respuesta del DPD presenta el comportamiento de expansión opuesto al del PA. Al combinar ambas respuestas, el resultado sería generalmente lineal, representado en la figura con una línea de puntos. Un comportamiento similar se apreciaría en el caso de la característica AM-PM.

Es importante tener en cuenta que las características inversas solo funcionan hasta el punto de saturación del PA por lo que, en la práctica, se deberá limitar la entrada a dicho nivel para prevenir que el PA se sature.

Por otra parte, en la Figura 4.1 también se muestran los valores de PAPR de entrada y salida, observándose que, en altos valores de potencia, la PAPR de la señal de entrada sufriría una compresión a la salida del amplificador. Sin embargo, en la curva del DPD, la PAPR de salida sería mayor y, al ser la entrada del PA, podría producir la saturación del amplificador. Esto podría prevenirse haciendo que la potencia media de la señal de entrada sea menor.

Otra alternativa sería usar una técnica conocida como reducción del factor de cresta o, en inglés, *Crest Factor Reduction* (CFR). La función de CFR se situaría antes del predistorsionador y consiste en reducir o eliminar los picos de la señal digital con el objetivo de reducir su PAPR. Existen diferentes formas de aplicar técnicas de CFR, pudiendo ser aplicado directamente, limitando la señal a un determinado valor, o usando técnicas más sofisticadas [6], [13].

El comportamiento de un predistorsionador también puede analizarse en el dominio de la frecuencia. Ya se vio anteriormente que el amplificador introduce armónicos, productos de intermodulación y nuevas componentes frecuenciales, además del comportamiento de compresión. Para eliminar los productos armónicos se podrían utilizar filtros, pero para reducir o eliminar las componentes de intermodulación, que interfieren en canales adyacentes, es necesario emplear técnicas de linealización.

El DPD añade componentes adicionales de distorsión en la señal de entrada del PA, que se corresponden con los productos de distorsión de alto orden del amplificador descritos por el modelo, de forma que estos se cancelen con las componentes de distorsión introducidos por el propio amplificador a su salida, al presentar fases contrarias.

Por tanto, debido a dichos términos, la señal de salida del DPD será “altamente distorsionada” [12] y su ancho de banda será mayor que el de la señal de entrada original. Concretamente, mostrará una distorsión espectral similar a la de la señal de salida del amplificador sin linealizar. Esto puede tener un gran impacto respecto al ancho de banda requerido por el DPD, además de la velocidad del DSP.

Finalmente, es importante tener en cuenta algunos aspectos acerca de la predistorsión:

- El DPD mejorará la distorsión del amplificador al que preceda, pero no ofrecerá más potencia de salida de lo posible. De hecho, en la práctica, un predistorsionador será normalmente un dispositivo pasivo, de modo que la ganancia se conseguirá minimizando la atenuación introducida por el mismo.
- En algunas ocasiones, puede ser un inconveniente usar la función matemática inversa exacta del modelo del amplificador para el predistorsionador, siendo posible usar funciones más simples [6]. Por este motivo, se suelen utilizar modelos basados en series de Volterra con algún tipo de simplificación o emplear técnicas de reducción de componentes como las revisadas en la Sección 3.4 para reducir la complejidad computacional del algoritmo del DPD.
- La configuración en cascada de dos dispositivos no lineales puede dejar residuos de productos no lineales de alto orden, anteriormente ausentes en la respuesta del PA original. Por ello, el DPD no solo debe linealizar la salida del PA, sino que también debe reducir la influencia de estos términos.

4.2.2 Evaluación del DPD

El modelado del comportamiento de un sistema y la predistorsión coinciden en una serie de pasos a la hora de llevarlos a cabo. Sin embargo, su evaluación es bastante diferente, puesto que en el primer caso se comparan las salidas medidas y estimadas del dispositivo bajo prueba (DUT), que sería el amplificador que se desea linealizar, y el modelo, respectivamente. Por otro lado, el comportamiento del DPD normalmente se evalúa mediante medidas de linealidad como las anteriormente comentadas, como el ACPR o el EVM, así como las características AM-AM y AM-PM del DUT linealizado.

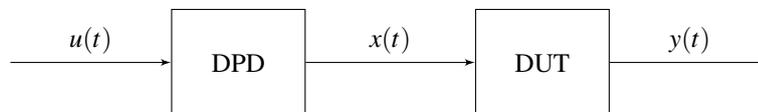


Figura 4.2 Diagrama de bloques genérico de un sistema de predistorsión digital.

Para asegurar que la cascada formada por el DPD y el DUT, como se muestra en la Figura 4.2, funciona como un sistema lineal amplificador, la señal de la salida $y(t)$ debe ser una réplica escalada de la señal de entrada al DPD $u(t)$, de modo que la relación entre la señal de salida y la de entrada será igual a la ganancia G del sistema linealizado, como se indica en la Ecuación (4.1).

$$G = \frac{y(t)}{u(t)} \quad (4.1)$$

Así, midiendo la entrada y la salida del DUT, se obtendrá información que puede ser usada en el entrenamiento del predistorsionador digital. Por tanto, el diseño del DPD puede verse como un problema de modelado, ya sea considerando el DPD como un sistema independiente o considerándolo en cascada junto

con el DUT [14]. En el primer caso, la evaluación del DPD se realizaría comparando la señal de salida con la salida que se esperaría idealmente, es decir, la señal que debería hacer de entrada al DUT.

En el segundo caso, el comportamiento del DPD se evaluaría considerando también el DUT. Usando la configuración en cascada, la señal $y(t)$ que saldría del DUT tendría que ser comparada con la señal de entrada $u(t)$ del DPD escalada con la ganancia ideal G puesto que, idealmente, la señal de salida tendría que ser una versión escalada de la señal de entrada, como se muestra en la siguiente ecuación.

$$y_{ideal}(t) = G \cdot u(t) \quad (4.2)$$

En la práctica, a la hora de evaluar un DPD, suele ser más fiable usar la segunda opción, es decir, considerando la cascada del DPD y el DUT, ya que el objetivo final es emplear el DPD junto con el DUT para conseguir un sistema lineal que satisfaga la Ecuación (4.2).

Hasta ahora se ha realizado una revisión de en qué consiste la predistorsión digital y cómo se evaluaría un predistorsionador. En la siguiente sección, se hará una comparación de las dos principales arquitecturas que existen para diseñar un predistorsionador.

4.3 Aprendizaje indirecto y aprendizaje directo

Actualmente existen dos tipos de arquitecturas para realizar el cálculo de los parámetros del predistorsionador: la arquitectura de aprendizaje indirecto (ILA) y la arquitectura de aprendizaje directo (DLA).

Ambas arquitecturas son casos particulares de sistemas de control inverso que, a su vez, son un caso particular del control por autoajuste o *auto-tuning*. En teoría de control, un sistema de autoajuste tiene la habilidad de realizar la optimización de sus propios parámetros, de forma que se maximice o minimice una función objetivo, siendo generalmente la minimización de un error.

Sin embargo, aunque ambas arquitecturas se basan en el autoajuste, funcionan de forma diferente. En primer lugar, el aprendizaje directo tiene como objetivo estimar la relación de entrada y salida de un amplificador, de modo que la predistorsión se obtenga “pre-invirtiéndose” las características del modelo del PA.

El aprendizaje indirecto, en cambio, consiste en obtener un postdistorsionador invirtiéndose el modelo no lineal y, entonces, este es usado como un predistorsionador. Sin embargo, la inversión de un sistema no lineal puede no ser posible. Esto se debe a que no todos los sistemas, ya sean lineales o no lineales, tienen inversa, y también existen sistemas que pueden tener inversa únicamente para un rango restringido de amplitud de entrada.

Además, el aprendizaje indirecto presenta algunos inconvenientes. Por una parte, la medida de la salida de un PA puede ser ruidosa, por lo que el algoritmo convergerá a valores con un cierto sesgo. En segundo lugar, los filtros no lineales no pueden conmutarse y en el caso de una estructura ILA, el modelo inverso adaptativo identificado es en realidad un modelo de postdistorsión. Por tanto, situar este modelo delante de un dispositivo no lineal no va a garantizar un buen modelo de predistorsión para el mismo. Estos dos inconvenientes, sin embargo, la arquitectura DLA no los presenta, lo que puede ser el motivo de que existan simulaciones que demuestran que se suelen obtener mejores resultados generalmente usando un predistorsionador basado en aprendizaje directo frente a un predistorsionador basado en aprendizaje indirecto [15]. Además, en el mismo documento se comenta que existen simulaciones que demuestran que es posible obtener un comportamiento aún mejor si se utilizan ambos esquemas a la vez, excepto para los predistorsionadores de correlación cruzada.

Tras esta introducción a este tipo de arquitecturas, se presentarán, en primer lugar, los principios en los que se basa un controlador de autoajuste, para luego continuar particularizando hacia las arquitecturas de aprendizaje indirecto y directo.

4.3.1 Controlador de autoajuste vs. arquitecturas de aprendizaje directo e indirecto

Aunque las arquitecturas de aprendizaje indirecto y directo se implementan de forma distinta, ambas son casos particulares del control de autoajuste. Como se ha comentado anteriormente, la técnica de autoajuste se llama así por su capacidad de ajustar sus propios parámetros. Este tipo de algoritmos, aplicados al ámbito de la predistorsión, pretenden obtener los parámetros del PA mediante un algoritmo de identificación recursiva.

La estructura general del controlador de autoajuste se muestra en la Figura 4.3 y presenta un bucle interno, más simple, y un bucle externo, que está formado por un bloque identificador y un bloque de diseño que ajusta los parámetros del controlador, el cual, a su vez, recibe tanto la señal de entrada como la señal de salida del amplificador. El bloque controlador de diseño se encargaría de recoger los parámetros del dispositivo,

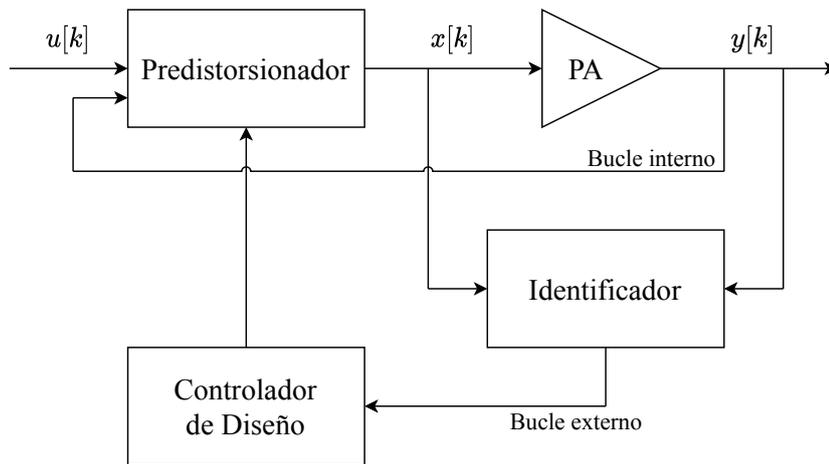


Figura 4.3 Controlador de autoajuste.

aplicar un algoritmo automático de diseño y, luego, ajustar los parámetros del bloque controlador, encargado de controlar directamente el dispositivo bajo prueba, el cual podría ser un amplificador.

Tomando como base el modelo del controlador de autoajuste, se pueden obtener los modelos de ILA y DLA. Si se elimina el bucle interior de la Figura 4.3, se obtendría un modelo similar al de aprendizaje indirecto, en el que el identificador se correspondería con un postdistorsionador y el bloque de diseño sería una copia de este para actuar como predistorsionador, como se muestra en la Figura 4.4.

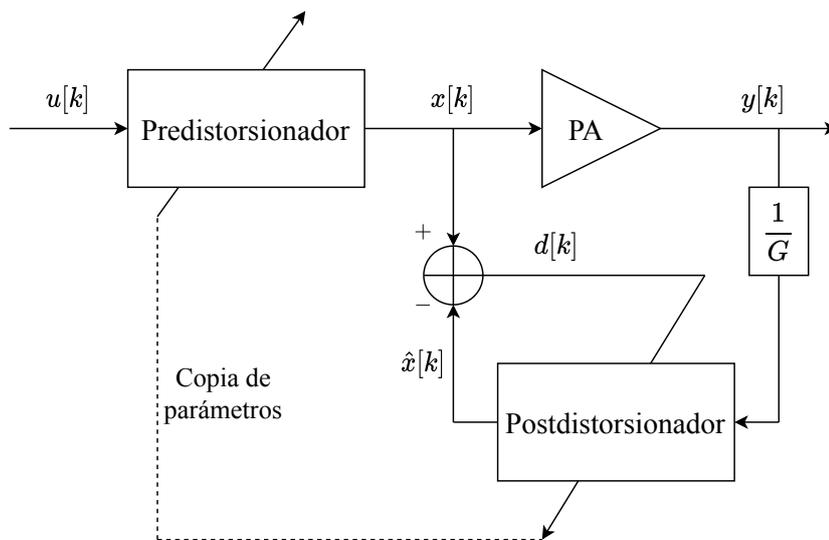


Figura 4.4 Arquitectura de aprendizaje indirecto.

Por otro lado, en el modelo de aprendizaje directo, el bloque identificador se encargaría de realizar una estimación directa del modelo y el bloque de diseño sería el encargado de calcular los parámetros para la predistorsión, quedando un esquema más simplificado en la Figura 4.5.

4.3.2 Arquitectura de aprendizaje indirecto

La arquitectura de aprendizaje indirecto es actualmente la técnica más utilizada para identificar los parámetros de los predistorsionadores digitales. En términos generales, esta arquitectura pretende estimar una “post-inversa” en lugar de una “pre-inversa” del PA. El modelo post-inverso se calcula usando la señal de salida del PA y su respectiva entrada, siguiendo la minimización de la señal diferencia entre $x[k]$ y $\hat{x}[k]$ y llegando a la solución LS anteriormente descrita en la Sección 3.3, gracias a la relación lineal que existe entre el modelo y los parámetros.

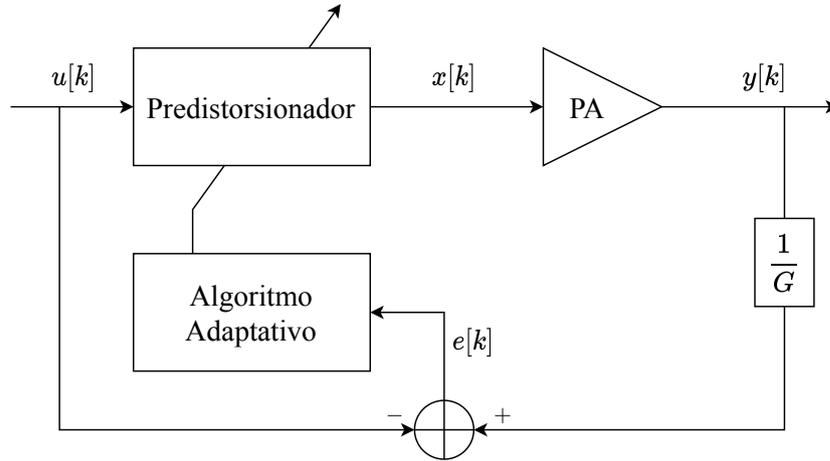


Figura 4.5 Arquitectura de aprendizaje directo.

Este modelo post-inverso se correspondería con los parámetros de un postdistorsionador que, una vez calculados, son copiados a un modelo idéntico que es usado como predistorsionador, situado previamente al amplificador.

La esencia de esta técnica se basa en el trabajo de Schetzen que establecía que la pre-inversa de orden p de un sistema es idéntica a su post-inversa. Esto se cumple asumiendo que el amplificador genera componentes hasta el orden p , lo cual es realista y válido para amplificadores ya que las componentes no lineales de orden alto tienen contribuciones insignificantes al comportamiento del PA [14].

La mayor limitación de una arquitectura de aprendizaje indirecto, junto con los inconvenientes comentados anteriormente, es que el hecho de obtener la función del predistorsionador intercambiando la salida y la entrada del amplificador lleva a una reducción del rango de potencias del predistorsionador comparado con el del amplificador [14].

4.3.3 Arquitectura de aprendizaje directo

Otra técnica popular utilizada para identificar los parámetros del predistorsionador es la arquitectura de aprendizaje directo. En general, la DLA se implementa en una serie de pasos. En primer lugar, se identifica un modelo del PA basado en series de Volterra. Luego, se utiliza un algoritmo no lineal para estimar, mediante un procesamiento iterativo, los parámetros del predistorsionador que minimizan cierta señal de error. En la Ecuación (4.3) y en la Ecuación (4.4) se han expresado matemáticamente dos posibilidades equivalentes que pueden usarse como señal de error.

$$e_1[k] = y_d[k] - y[k] = Gu[k] - y[k] \quad (4.3)$$

$$e_2[k] = \frac{1}{G}y[k] - u[k] \quad (4.4)$$

En la primera, se calcularía la diferencia entre la señal de salida deseada $y_d[k]$ y la señal de salida del amplificador real $y[k]$. En el segundo caso, el error vendría definido por la diferencia entre la señal de salida tras sustraer la ganancia introducida por el amplificador y la señal de entrada del DPD.

Tras obtener el error, se utilizan algoritmos de optimización para estimar los parámetros del predistorsionador. Generalmente, los algoritmos de optimización presentan estructuras complejas, así como una convergencia lenta, por lo que suelen requerir un alto coste computacional. Además, si el amplificador entra en saturación, se pueden producir divergencias en el algoritmo adaptativo, provocando que este intente incrementar, repetidamente y sin éxito, la potencia de salida del DPD con el objetivo de corregir unas distorsiones incorregibles inducidas por la saturación. Por este motivo, existen trabajos que buscan no solo conseguir buenos resultados, sino también algoritmos que reduzcan el coste computacional, algunos de los cuales serán presentados en el siguiente capítulo.

Una vez se aplica el algoritmo y se encuentra una solución, los parámetros estimados son empleados en generar una señal predistorsionada $x[k]$ que es aplicada como entrada al PA. Con la nueva salida del amplificador, se vuelve a ejecutar el mismo procedimiento, de modo que se completa un número determinado

de iteraciones hasta que el sistema compuesto por el DPD seguido del PA converge a la mejor solución posible.

En este trabajo, se realizarán una serie de experimentos de distintas características. Sin embargo, todos coincidirán en la aplicación de diferentes alternativas de aprendizaje directo por lo que, en este caso, se hará un mayor hincapié en este tipo de arquitectura. Tras haber introducido los conceptos más importantes, en el siguiente capítulo se hará una explicación de los casos más concretos que serán probados, así como los resultados tras aplicarlos en los experimentos correspondientes.

5 Técnicas y experimentos

An experiment is a question which science poses to Nature, and a measurement is the recording of Nature's answer.

MAX PLANCK

En capítulos anteriores se ha realizado una revisión de cómo se comportan los amplificadores de potencia, pasando por diferentes herramientas para caracterizarlos y modelarlos para continuar estudiando la técnica de la predistorsión digital. En este capítulo, se expondrán una serie de experimentos *hardware*, empleando diferentes algoritmos y técnicas de aprendizaje directo para conseguir la linealización de un amplificador real.

De este modo, el presente capítulo se estructurará de manera que, en primer lugar, se presentarán las características generales y algunas de las consideraciones más importantes que tendrán en común todos los experimentos. Posteriormente, se mostrarán los resultados obtenidos, precedidos por la revisión de las técnicas empleadas en la realización de cada experimento.

5.1 Configuración de los experimentos

Para realizar experimentos reales de predistorsión es necesario disponer de una configuración con un amplificador de potencia al que realizar el modelado y, posteriormente, calcular los coeficientes del predistorsionador, como se ha comentado en el Capítulo 4; además de los instrumentos necesarios, como un generador de señal (VSG) y un analizador de señal (VSA).

En esta sección se hará una introducción del laboratorio y las herramientas utilizadas, así como las características de las señales y otros parámetros comunes a todos los experimentos.

5.1.1 WebLab de Chalmers

El estudio de diferentes técnicas de predistorsión y, concretamente, de aprendizaje directo en este trabajo se realizó con vistas a la competición de estudiantes denominada *IMS Digital PreDistortion Student Design Competition*, que estaba programada para ser realizada durante la celebración del *International Microwave Symposium (IMS)* en 2020, que tuvo que ser cancelada. Por ello, los diferentes experimentos se han llevado a cabo haciendo uso de la plataforma WebLab [16], que se encuentra en el laboratorio de la Universidad Tecnológica de Chalmers (Suecia). Este laboratorio cuenta con un ordenador en el que se está ejecutando una versión del software MATLAB[®], que es capaz de comunicarse con un VST, modelo PXIe-5646R, que cuenta con un ancho de banda instantáneo de 200 MHz. Este dispositivo genera una señal que es aplicada a la entrada de un amplificador CGH40006P. A continuación, la salida del PA pasa a través de un atenuador de 30 dB para que esta señal pueda ser reconducida hacia el receptor del VST. Esta configuración ha sido representada en la Figura 5.1 y su principal ventaja es que el ordenador se encuentra conectado a Internet y, haciendo uso de MATLAB[®], permite realizar medidas a distancia, lo cual lo convierte en una herramienta muy útil, utilizada para los experimentos que se explicarán más adelante.

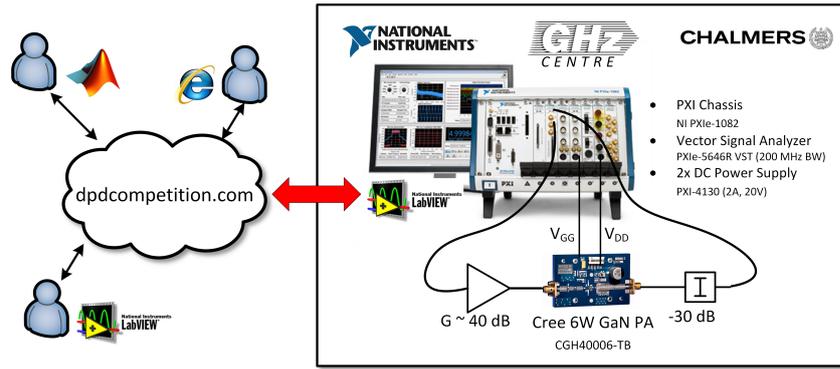


Figura 5.1 Configuración del laboratorio de Chalmers [17].

5.1.2 Características de la señal

Tras introducir el laboratorio empleado para la realización de los experimentos, es importante describir las principales características de las señales que serán enviadas al amplificador. En todos los experimentos, se generará una señal que siga el formato del enlace descendente del estándar de comunicaciones móviles de cuarta generación (LTE). Concretamente, esta se tratará de una señal OFDM de la que se pueden especificar los siguientes aspectos:

- 75 bloques de recursos activos, con 12 subportadoras cada uno, es decir, 900 subportadoras activas.
- La separación entre portadoras será de 15 kHz.
- Se utilizará una banda centrada con un ancho de 15 MHz.
- La frecuencia de muestreo de la señal será de 92,16 MHz, como resultado de la Ecuación (5.1), donde Δ_f es la separación entre portadoras, NFFT representa el tamaño de la FFT, y OVS se corresponde con el sobremuestreo o, en inglés, *oversampling*.

$$f_s = \Delta_f \cdot \text{NFFT} \cdot \text{OVS} = 15 \cdot 10^3 \cdot 1024 \cdot 6 = 92,16 \text{ MHz} \quad (5.1)$$

- La potencia de la señal será escalada teniendo en cuenta las restricciones del WebLab de Chalmers, por lo que se establecerá una potencia de entrada promedio de -22 dBm.
- Con las características establecidas, el espectro de la señal de entrada se muestra en la Figura 5.2.

Un aspecto que es importante aclarar es que el WebLab de Chalmers, utilizado para realizar los experimentos, presenta una frecuencia de muestreo fija de 200 MHz. Como la señal generada en este caso es de 92,16 MHz, será necesario realizar un paso adicional, volviéndola a muestrear teniendo en cuenta los 200 MHz de la frecuencia de muestreo del WebLab.

5.1.3 Modelado del amplificador

Como se ha comentado en anteriores capítulos, antes de realizar el cálculo de los coeficientes de un predistorcionador, es necesario obtener un modelo del amplificador bajo prueba. En este caso, el procedimiento de modelado será el mismo para cada uno de los experimentos, puesto que se prestará mayor atención en la evaluación de técnicas de aprendizaje directo para el cálculo de los coeficientes del predistorcionador.

Para obtener el modelo del amplificador, se ha optado por recurrir al modelo GMP. Como se explicó en la Sección 3.2.3, este tiene en cuenta algunos términos cruzados. Concretamente, para la configuración del modelo empleado en este caso, se han utilizado los parámetros recogidos en la Tabla 5.1. En esta se puede observar que se ha dado una mayor importancia a los términos correspondientes al caso de un modelo MP, es decir, aquellos términos no cruzados, mientras que se ha utilizado un menor número de términos tanto para los retrasados como para los adelantados.

El procedimiento seguido para obtener el modelo del amplificador en cuestión, consiste en generar una señal de entrada $u[k]$ con las características especificadas en la Sección 5.1.2 para, a continuación, enviarla al

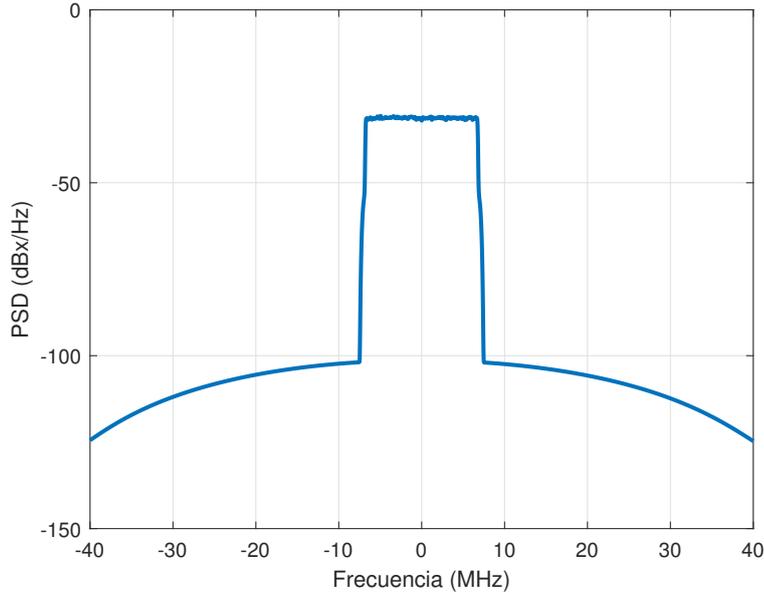


Figura 5.2 Espectro de la señal en banda base.

Tabla 5.1 Parámetros de configuración del modelo GMP utilizado.

<i>Kernel</i>	<i>Variable</i>	<i>Valor</i>
a_{nl}	n	[0 1 2 3 4 5 6 7 8 9 10 11 12]
	l	[15 10 5 5 5 5 5 1 1 1 1 1 1]
b_{nlm}	n	[2 4 6]
	l	[1 1 1]
	m	[1 1 1]
c_{nlm}	n	[2 4 6]
	l	[1 1 1]
	m	[1 1 1]

WebLab y aplicarla como entrada del amplificador. Seguidamente, la señal de salida del mismo es recogida y, utilizando la relación entre la señal de entrada $u[k]$ y la señal de salida $y[k]$, es posible obtener la matriz de los regresores que, para recordar que se ha calculado usando la señal de entrada $u[k]$, será denotada como \mathbf{U} . Esta matriz tendrá una gran importancia en las siguientes etapas de los experimentos, donde se realizará la predistorsión aplicando técnicas de aprendizaje directo que requerirán diferentes formas de procesar dicha matriz para calcular los coeficientes del predistorsionador.

5.1.4 Análisis de las medidas

A lo largo del aprendizaje, como en cada iteración se obtendrá la salida del amplificador, será posible seguir la evolución de las características de la señal. Además, una vez haya terminado el proceso de aprendizaje, se debe llevar a cabo una fase de validación para poder analizar los resultados que se puedan obtener en un futuro, generalizando las medidas para señales diferentes usadas en el entrenamiento. Por este motivo, será de gran utilidad analizar las medidas obtenidas con una serie de parámetros. Aunque existe una gran variedad de parámetros que se pueden medir, de los cuales algunos ya fueron presentados en la Sección 2.3.1, en el caso de los experimentos que se realizarán en este trabajo, se analizarán los siguientes parámetros:

- Se calcularán cuatro valores de ACPR para las señales de salida del PA. Se recogerán tanto para los

canales adyacentes superior e inferior, como para los canales superior e inferior de los mismos, es decir, los segundos canales adyacentes. El ACPR correspondiente con el canal inmediatamente inferior será denotado como $ACPR_a$, el inmediatamente superior será $ACPR_b$, el del segundo canal inferior será $ACPR_{2a}$ y el del segundo canal superior será $ACPR_{2b}$.

- El valor del error EVM será calculado como manera de analizar los resultados obtenidos en la constelación de la señal recibida.
- La diferencia entre la señal previa al DPD y la salida del PA será medida mediante el valor de NMSE.

Tras haber repasado los aspectos generales, comunes a todos los experimentos, en las próximas secciones se entrará en mayor detalle de cada una de las técnicas utilizadas, seguidas de los resultados obtenidos para cada una de ellas.

5.2 Aprendizaje directo con bucle cerrado

En la Sección 4.3.3 se adelantaron los fundamentos de cómo funciona la arquitectura de aprendizaje directo. Sin embargo, existen diferentes variantes a la hora de implementar este tipo de aprendizaje, existiendo muchas posibilidades en la ejecución del algoritmo adaptativo que se encarga de la obtención de los parámetros del DPD.

5.2.1 Funcionamiento del algoritmo

En esta sección, se especificará el algoritmo utilizado, cumpliendo con la arquitectura de aprendizaje directo, que tendrá como base el sistema de bucle cerrado expuesto en [18]. Cabe decir que el funcionamiento de esta arquitectura no será únicamente utilizado en los experimentos de esta sección, sino que también servirán como base para los siguientes, los cuales incluirán modificaciones o añadirán técnicas para luego comparar los resultados.

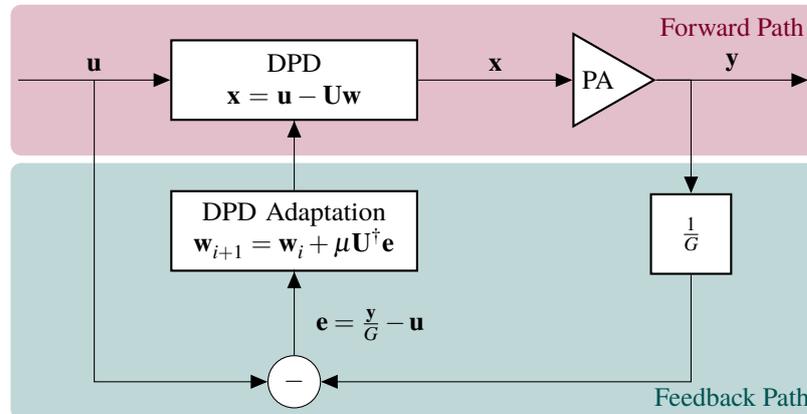


Figura 5.3 Diagrama de bloques de la DLA utilizada.

El diagrama de bloques de la arquitectura de aprendizaje directo utilizada se muestra en la Figura 5.3 y, como se puede comprobar, presenta un mayor detalle que la presentada en la Sección 4.3.3. En este diagrama se ha diferenciado con diferentes colores el camino hacia delante (*Forward Path*) y el camino hacia atrás (*Feedback Path*).

Sombreado en rojo, el *Forward Path* se inicia con la entrada de la señal que se desea transmitir u . Esta señal sería procesada por el DPD, que ofrecería a su salida la señal predistorsionada x . Concretamente, esta señal es generada a partir de la diferencia entre la señal original y una señal que podría denotarse como d , como se muestra en la Ecuación (5.2). La señal d es calculada como el producto de la matriz U , que contiene los regresores del modelo del amplificador calculado previamente, y el vector w , que almacena los valores de los pesos del DPD. Como la matriz U se obtiene antes de comenzar con el aprendizaje directo, su valor no cambia a lo largo de este proceso. Los pesos w , en cambio, son el principal objetivo del propio aprendizaje

por lo que los valores de este vector deberán ir variando cada iteración hasta que el algoritmo de adaptación converja.

$$\mathbf{x} = \mathbf{u} - \mathbf{d} = \mathbf{u} - \mathbf{U}\mathbf{w} \quad (5.2)$$

La señal predistorsionada \mathbf{x} es enviada al amplificador, haciendo uso del WebLab, con un valor de potencia de entrada promedio de -22 dBm. A continuación, la plataforma devuelve la señal de salida \mathbf{y} del amplificador del laboratorio, acabando en este punto el avance hacia delante. A continuación se iniciaría el proceso del *Feedback Path*, en el que se comenzaría con la iteración de aprendizaje de los pesos. En primer lugar, la señal de salida \mathbf{y} debe ser normalizada. Aunque en la Figura 5.3 se representa este paso como una división entre la ganancia del amplificador, lo cual también sería válido, en este caso se ha optado por normalizar la señal de salida dividiéndola entre su norma y , luego, multiplicándola por la norma de la señal de entrada, de modo que también se eliminaría la ganancia introducida por el amplificador.

Tras esto, es posible calcular la señal de error \mathbf{e} que, como se muestra en la Ecuación (5.3), se define como la diferencia entre la señal de salida del amplificador sin la ganancia y la señal de entrada original.

$$\mathbf{e} = \frac{\mathbf{y}}{G} - \mathbf{u} \quad (5.3)$$

Por último, el vector de error es utilizado para realizar la actualización de los pesos, siguiendo la solución LS como se indica en la Ecuación (5.4). En esta, se puede observar que, en primer lugar, se calcularía el producto entre la pseudoinversa de Moore-Penrose de la matriz de regresores y el vector error. A este término, se le incluiría un coeficiente μ , con valores en el intervalo $0 < \mu \leq 1$, que sería un factor de ponderación, con la función de ajustar el grado de aprendizaje o de actualización de los pesos. Finalmente, el vector resultante sería sumado al vector de pesos de la iteración anterior para finalizar con la actualización de los pesos y, si procede, se realizaría una nueva iteración del bucle.

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \mathbf{U}^\dagger \mathbf{e} \quad (5.4)$$

Cabe destacar que la inicialización de los pesos \mathbf{w} se puede llevar a cabo de diferentes formas. Una forma sencilla es inicializar el vector con los valores a cero. En este caso, en la primera iteración, la señal predistorsionada \mathbf{x} y la señal de entrada al DPD, \mathbf{u} , serían idénticas. Posteriormente, en la primera actualización de los pesos, pasarían a tomar valores distintos de cero.

Otra alternativa sería inicializar los pesos utilizando las señales de entrada y salida del amplificador en la medida inicial. Si se recuerda, antes del proceso de aprendizaje directo, se debe realizar una primera medida para generar el modelo \mathbf{U} . Si estas señales están disponibles en el proceso de aprendizaje, se podrían utilizar para obtener un vector de error inicial \mathbf{e} , según la Ecuación (5.3), e inicializar los pesos multiplicando la pseudoinversa de la matriz \mathbf{U} por dicho vector.

En el primer caso, las señales de entrada y de salida del amplificador de la primera medida no son requeridos, por lo que el proceso de aprendizaje directo no necesitaría entradas externas para comenzar. En el segundo caso, en cambio, se necesitaría contar con dichas señales, presentando la ventaja de ahorrar una iteración del algoritmo. No obstante, el resultado de inicializar los pesos de una forma u otra no deberá repercutir en el resultado, más allá de requerir una iteración más en la primera alternativa.

Un aspecto importante que se debe tener en cuenta en las técnicas de aprendizaje directo reside en la ecuación del cálculo de los coeficientes. Concretamente, en la Ecuación (5.4) se observa la presencia de la pseudoinversa de la matriz de regresores. Este término requiere un gran coste computacional, por lo que se debe evitar su cálculo un número excesivo de veces. Para tener esto en cuenta, en la implementación de este algoritmo se ha realizado este cálculo una única vez, fuera del bucle para luego, al aplicar la ecuación de la actualización de pesos, utilizar el mismo resultado, pues este no varía a lo largo del aprendizaje.

Por otro lado, el proceso de aprendizaje es iterativo, por lo que se deberá realizar una serie de veces hasta que el algoritmo converja, es decir, hasta que, a pesar de realizar más iteraciones, el valor de los pesos \mathbf{w} , por tanto, de las características de las señales obtenidas se queden estabilizados en torno a un determinado valor.

Una vez que el algoritmo haya convergido, es necesario llevar a cabo una fase de validación. En esta, se realizaría únicamente el camino hacia delante, puesto que el aprendizaje ya habría terminado. En esta última fase, el procedimiento empezaría por crear una señal, \mathbf{u}_2 , distinta a la utilizada para aprender y generar su correspondiente señal predistorsionada.

En este caso, la generación de la señal predistorsionada será como se expresa en la Ecuación (5.5). Como se puede apreciar, esta ecuación es muy similar a la Ecuación (5.2), sin embargo, cuenta con ciertos matices que es necesario aclarar. En primer lugar, el vector de pesos será el que se ha obtenido como resultado del

proceso de aprendizaje, siendo este vector común para cualquier señal de entrada al sistema. Sin embargo, es importante tener en cuenta que la matriz con los regresores debe ser la correspondiente a la señal de validación \mathbf{u}_2 , siendo denotada en dicha ecuación como \mathbf{U}_2 .

$$\mathbf{x}_2 = \mathbf{u}_2 - \mathbf{U}_2 \mathbf{w} \quad (5.5)$$

Finalmente, tras generar la señal predistorsionada \mathbf{x}_2 , esta es enviada al amplificador, para recoger su salida correspondiente y analizar sus características. En el siguiente apartado, se especificarán los parámetros de las señales que serán analizados con el objetivo de evaluar el funcionamiento del sistema completo.

5.2.2 Experimento 1: Variación del factor de ponderación

Con todas las consideraciones explicadas hasta el momento, es posible comenzar con la explicación del primer experimento. En este primer caso, el experimento consiste en llevar a cabo el entrenamiento siguiendo el algoritmo explicado en el apartado anterior, fijando un máximo de 15 iteraciones en el proceso de aprendizaje. A lo largo de cada iteración, se toman los parámetros descritos anteriormente y, para cada uno de ellos, se mostrará una gráfica con la evolución a lo largo del aprendizaje, teniendo un punto por cada iteración.

Además, esto es realizado fijando diferentes valores del parámetro μ para observar cómo afecta tanto en el proceso de aprendizaje como en los resultados finales. Concretamente, se fijarán 4 valores distintos: 1/4, 1/2, 3/4 y 1. El Código 5.1 incluye tanto los parámetros de configuración como el bucle de aprendizaje de este método.

Código 5.1 Bucle del experimento 1.

```

1 %Inicialización de parámetros
2 Niter_DL=15;
3 mu=1; %Factor de ponderación
4 [N,f]=size(U);
5
6 %Inicialización de los pesos a 0
7 w = zeros(f,1);
8 %Inicialización de los pesos usando la medida inicial
9 %w = w + pinv_U*(y*norm(u)/norm(y)-u);
10
11 %Cálculo de la pseudoinversa
12 pinv_U=pinv(U);
13
14 %Proceso de aprendizaje
15 for iter=1:Niter_DL
16     %Señal predistorsionada
17     x1 = u-U*w;
18
19     %Medida
20     [yPA, xPA, ~] = myPACHalmers(x1, RMSin, Nmeas);
21
22     %Normalización de y
23     yPAnorm = yPA*norm(xPA)/norm(yPA);
24
25     %Diferencia entre la señal de salida normalizada y la entrada
26     e1 = yPAnorm-u;
27
28     %Actualización de los pesos
29     dw1 = pinv_U*e1;
30     w = w + mu * dw1;
31 end

```

Por último, antes de comenzar a comentar la representación de los resultados, es importante realizar una aclaración. En todas ellas se podrán apreciar no solo 15 iteraciones, sino que también se muestra una iteración

0 y una iteración 16. Los valores asignados a la iteración 0 se corresponden con los obtenidos en la primera medida, que es utilizada para construir el modelo. En este caso, los parámetros del DPD se han inicializado a 0, por lo que los dos primeros valores en las gráficas deberán ser parecidos (aunque no necesariamente iguales). La última medida, siendo representada en las gráficas como la iteración número 16, se corresponde con los valores medidos para la señal de validación. Idealmente, los valores obtenidos en esta última iteración debería ser similares a los de la anterior iteración, es decir, a la última del entrenamiento. Sin embargo, es posible que existan ocasiones en los que el DPD no sea capaz de generalizar para otras señales y, por tanto, las características no solo no sean iguales, sino que sean bastante peores respecto a las obtenidas durante el entrenamiento.

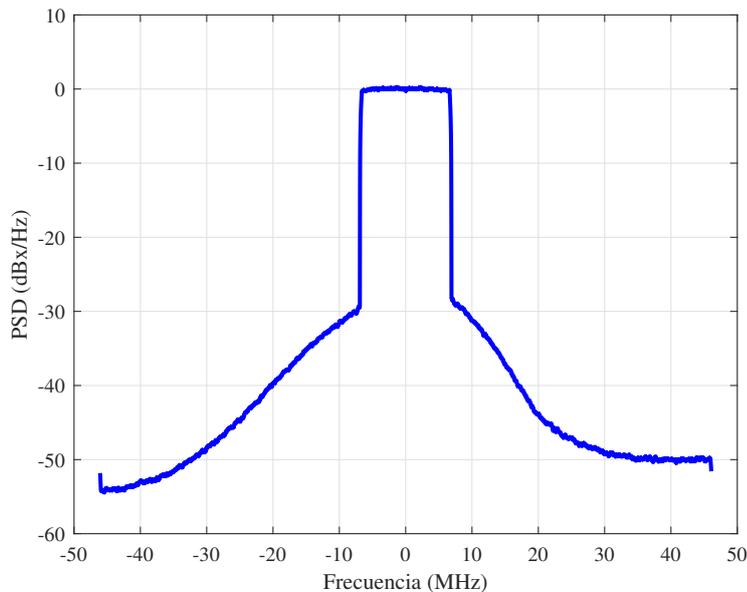


Figura 5.4 Espectro de la señal de salida sin DPD.

Dicho esto, será más fácil realizar el análisis de los resultados obtenidos. En la Figura 5.5 y en la Figura 5.6 se muestran los valores de ACPR para el canal inferior y el canal superior, respectivamente. En primer lugar, se puede apreciar un comportamiento similar para ambos canales. Sin embargo, se observa que el amplificador no presenta una respuesta perfectamente simétrica pues, en las primeras medidas, en las que no se ha aplicado DPD, se obtiene un $ACPR_a$ menor a -36 dBc, mientras que el $ACPR_b$ se encuentra por encima de dicho valor. Este efecto se puede apreciar también si se analiza el espectro de la señal de salida del amplificador, representado en la Figura 5.4, donde la diferencia entre las bandas adyacentes se muestra más claramente. De hecho, en los segundos canales la diferencia es aún mayor, por lo que este efecto tiene más repercusión en la Figura 5.7 y la Figura 5.8. En la primera, se aprecia un $ACPR_{2a}$ superior a -46 dBc, mientras que en la segunda, se tiene un $ACPR_{2b}$ inferior a los -44 dBc, existiendo una diferencia de 2 dB entre dichos valores. Este efecto de asimetría que se aprecia en el espectro de la señal de salida indica que el amplificador de potencia presenta efectos de memoria.

Pasando al estudio del proceso de entrenamiento, en ambos casos se observa un descenso similar del ACPR de la primera banda, mejorando desde los -36 dBc hasta encontrarse por debajo de los -43 dBc en las últimas iteraciones del aprendizaje, lo que sería una mejora de unos 7 dB. Haciendo un estudio de las gráficas de los valores de ACPR de los segundos canales adyacentes, también se aprecia un importante descenso. En este caso, como el amplificador es más asimétrico para estas bandas, se obtienen valores con algunos decibelios de diferencia. Concretamente, el valor de $ACPR_{2a}$ llega hasta los -52 dBc, mientras que el $ACPR_{2b}$ se queda en torno a los -49 dBc aunque, a pesar de ser distintos, se pueden considerar unos resultados aceptables.

Respecto a los resultados obtenidos de EVM en la Figura 5.9, antes de aplicar ningún tipo de DPD, se parte de un error de aproximadamente 4%. Sin embargo, con este método, en cada iteración va mejorando hasta conseguir un EVM en torno a 1,7%, lo que significa que los puntos recibidos en la constelación se encontrarán más concentrados en los puntos transmitidos.

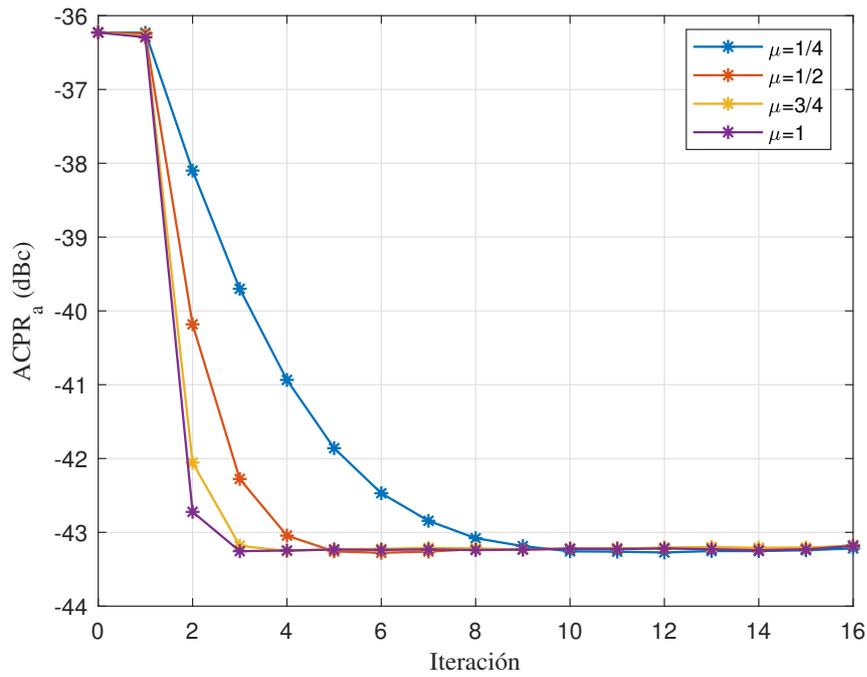


Figura 5.5 Evolución del ACPR del canal inferior en el experimento 1.

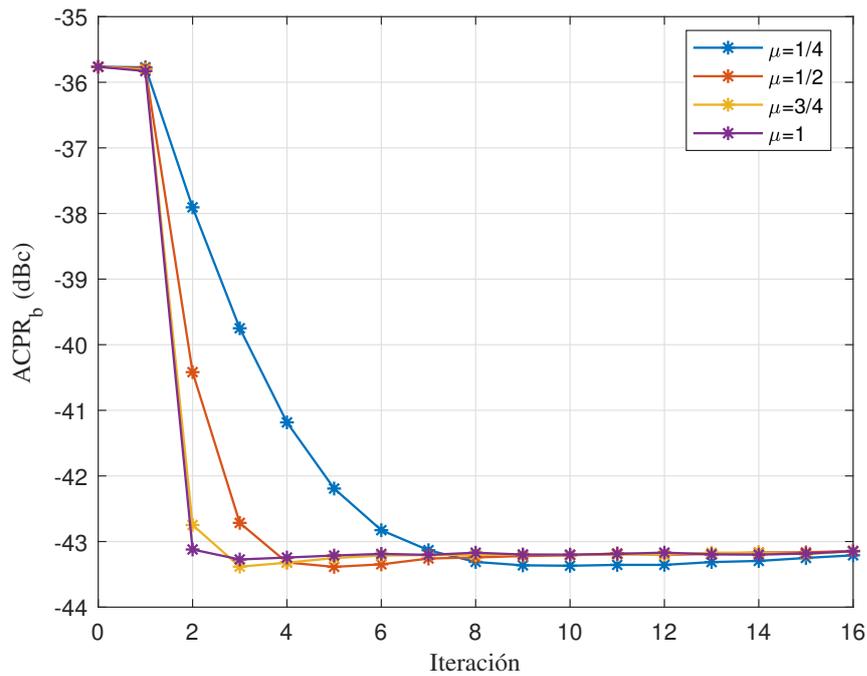


Figura 5.6 Evolución del ACPR del canal superior en el experimento 1.

Por último, un comportamiento similar se obtiene al analizar la evolución del NMSE mostrada en la Figura 5.16. Al enviar la señal directamente al amplificador, se obtiene un NMSE mayor a -28 dB mientras que, una vez se ha realizado el entrenamiento del DPD, se consiguen valores por debajo de los -35 dB, suponiendo una mejora de 7 dB.

Un comportamiento en el proceso de aprendizaje del DPD que se puede observar en todas las gráficas, no solo en las medidas de este experimento, sino también en los que se comentarán posteriormente, es el efecto del valor del factor de ponderación μ . Si este es pequeño, el aprendizaje avanza más lentamente, mientras que en el caso de ser mayor, por ejemplo $\mu = 1$, el entrenamiento converge mucho antes. Con estos resultados se puede llegar a la conclusión de que, para que el algoritmo converja en un número menor de iteraciones,

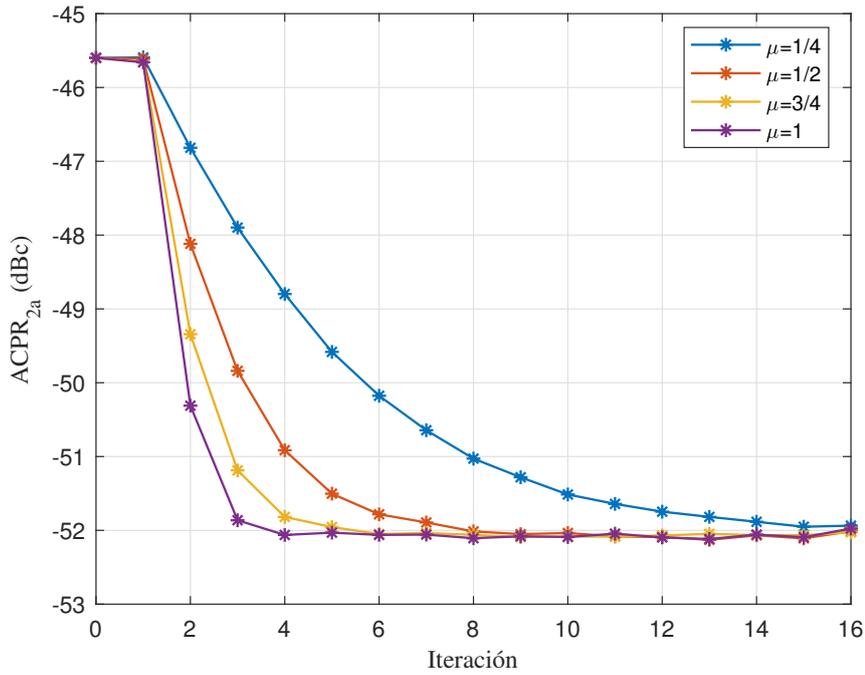


Figura 5.7 Evolución del $ACPR_{2a}$ en el experimento 1.

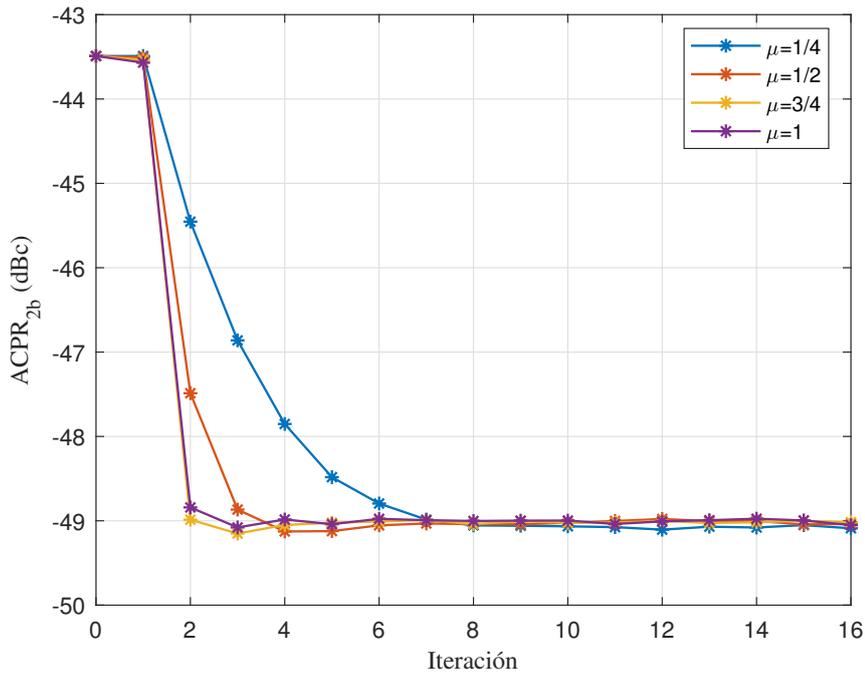


Figura 5.8 Evolución del $ACPR_{2b}$ en el experimento 1.

se debe configurar un valor cercano a 1 de dicho parámetro. Sin embargo, el hecho de utilizar un factor de ponderación menor y, por tanto, invertir un mayor número de iteraciones en el entrenamiento, puede llegar a ofrecer resultados ligeramente mejores. Un ejemplo de este efecto se puede observar en la Figura 5.6, donde la curva azul, correspondiente al entrenamiento con menor μ , presenta valores de ACPR ligeramente menores que los obtenidos en el resto de casos.

Como último comentario de este experimento, es importante comparar los resultados obtenidos en las iteraciones 15 y 16. Como se ha comentado anteriormente, la primera se corresponde con la última iteración del aprendizaje, mientras que la segunda se corresponde con la medida de validación. En todas las figuras, se obtiene que los valores de ambas medidas son muy similares. Esto es muy positivo, puesto que significa que

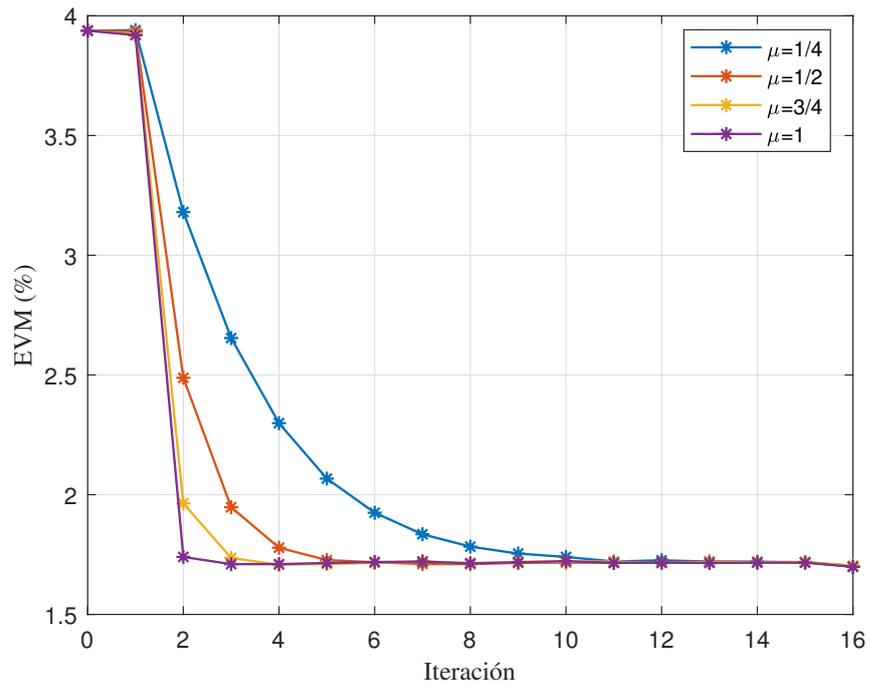


Figura 5.9 Evolución del EVM en el experimento 1.

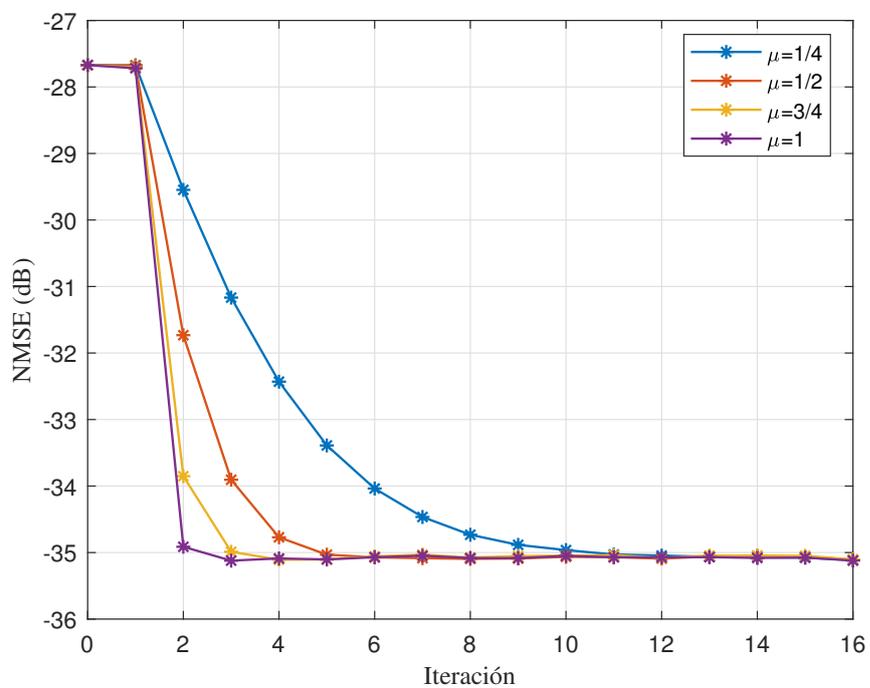


Figura 5.10 Evolución del NMSE en el experimento 1.

el DPD resultante de este algoritmo es capaz de generalizar de modo que, si se utiliza una señal diferente a la del entrenamiento, las prestaciones conseguidas serán similares a las del caso de utilizar la misma señal.

5.3 Aprendizaje directo de base adaptable

5.3.1 Funcionamiento del algoritmo

En esta sección se hablará de un método presentado a principios de este mismo año 2020. En [19], se introduce un método en el que, durante el entrenamiento, no varía únicamente el vector de pesos, sino que la base también cambia con ellos.

Concretamente, se parte de la misma ecuación de actualización de pesos de la mayoría de esquemas DLA, descrita en la Ecuación (5.4). Esta se muestra desglosada en la Ecuación (5.6), donde la pseudoinversa de Moore-Penrose se ha sustituido por su definición y la señal de error \mathbf{e} por la diferencia entre la señal de salida del amplificador $\tilde{\mathbf{y}}$ tras eliminarle la ganancia y la señal de entrada \mathbf{u} .

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \mathbf{U}^\dagger \mathbf{e} = \mathbf{w}_i + \mu (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{U}^H (\tilde{\mathbf{y}} - \mathbf{u}) \quad (5.6)$$

Por otro lado, se tendría la Ecuación (5.2), que muestra cómo se generaría una señal predistorsionada. Al fin y al cabo, el vector de pesos del DPD \mathbf{w} será utilizado para predistorsionar una señal, multiplicando por su izquierda la matriz \mathbf{U} y, posteriormente, restando este resultado a dicha señal. Por tanto, la propuesta de [19] consiste en multiplicar esa matriz no solo en la predistorsión, sino a la hora de entrenar los pesos. Así, se efectuaría la multiplicación de esa matriz en la ecuación de actualización de pesos, como se representa en la Ecuación (5.7) y, a continuación, se asume que el producto de la matriz por su pseudoinversa es aproximadamente la matriz identidad $\mathbf{U}\mathbf{U}^\dagger \approx \mathbf{I}$, llegando finalmente a la Ecuación (5.8).

$$\mathbf{U}\mathbf{w}_{i+1} = \mathbf{U}\mathbf{w}_i + \mu \mathbf{U} (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{U}^H (\tilde{\mathbf{y}} - \mathbf{u}) \quad (5.7)$$

$$\mathbf{U}\mathbf{w}_{i+1} \approx \mathbf{U}\mathbf{w}_i + \mu (\tilde{\mathbf{y}} - \mathbf{u}) \quad (5.8)$$

De esta forma, durante el entrenamiento se evita realizar el cálculo de la pseudoinversa de la matriz, por lo que el coste computacional de este método resulta ser menor. Únicamente al terminar el proceso de aprendizaje, se debería calcular una pseudoinversa para obtener, finalmente, el vector de pesos del DPD para, luego, poder generar la señal predistorsionada en la fase de validación.

Aunque en [19] se tienen en cuenta algunos otros aspectos, como la inicialización de la matriz mediante ILA o la aplicación de OMP, en este trabajo se han seguido las indicaciones anteriormente explicadas para la realización de los experimentos, los cuales se aportan en el siguiente apartado.

5.3.2 Experimento 2: Variación del factor de ponderación con base adaptable

Las condiciones de este experimento son similares a las del experimento 1. En este caso, también se representará la evolución de los parámetros a lo largo de las iteraciones del aprendizaje, coincidiendo la última muestra, representada como la iteración 16, con la medida de la señal validación. El Código 5.2 se corresponde con la implementación de este método en MATLAB[®] utilizada para este experimento.

Código 5.2 Bucle del experimento 2.

```

1 %Inicialización de parámetros
2 N_iter=15;
3 mu=1; %Factor de ponderación
4 [N,f]=size(U);
5
6 %Inicialización de los pesos a 0
7 w = zeros(f,1);
8 U_w=U*w; %Matriz de aprendizaje
9
10 %Proceso de aprendizaje
11 for iter=1:N_iter
12     %Señal predistorsionada
13     x1 = u-U_w;
14

```

```

15 %Medida
16 [yPA, xPA, ~] = myPACHalmers(x1, RMSin, Nmeas);
17
18 %Normalización de y
19 yPAnorm = yPA*norm(xPA)/norm(yPA);
20
21 %Diferencia entre la señal de salida normalizada y la entrada
22 e1 = yPAnorm-u;
23
24 %Actualización de la matriz de aprendizaje
25 U_w = U_w + mu * e1;
26 end
27 w=pinv(U)*U_w; %Pesos finales

```

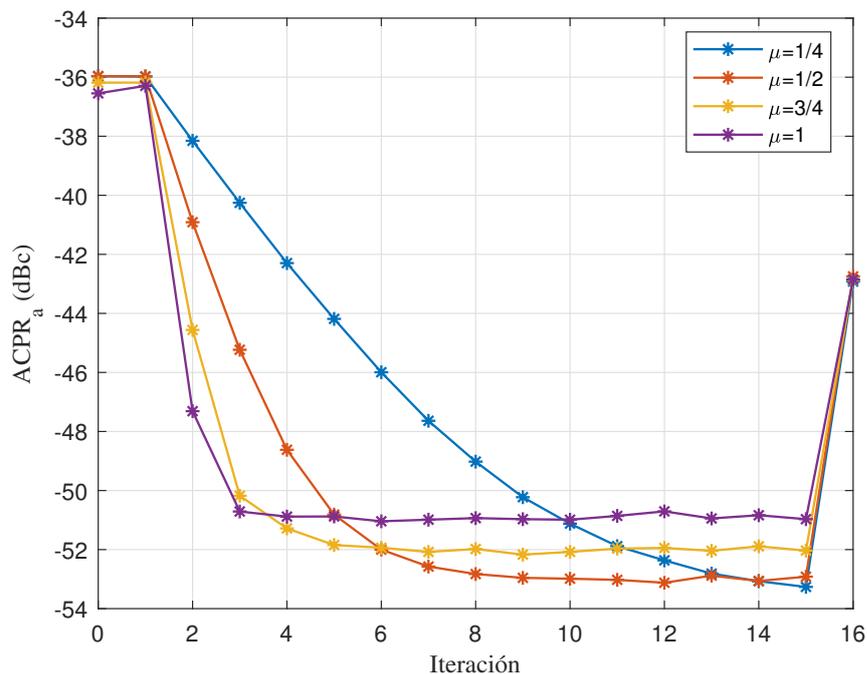


Figura 5.11 Evolución del ACPR del canal inferior en el experimento 2.

Aunque la configuración de los experimentos solo varía en el método utilizado respecto al experimento anterior, los resultados muestran características muy interesantes. Comenzando con las gráficas de ACPR de la banda adyacente inferior y superior, correspondiéndose con la Figura 5.11 y la Figura 5.12, respectivamente, ambas presentan un comportamiento similar. Además, ambas coinciden en un fenómeno que llama la atención al analizar las gráficas. En primer lugar, en las dos gráficas se aprecia un considerable descenso del ACPR a lo largo del aprendizaje, pasando desde los -36 dBc obtenidos sin aplicar predistorsión para llegar a ofrecer valores por debajo de los -50 dBc, dependiendo del valor del parámetro μ . De hecho, se comentó en el anterior experimento que en algunas ocasiones se apreciaba el efecto de este, además del ritmo de aprendizaje, en los resultados obtenidos. Como se comprueba, no únicamente en estas dos gráficas, sino en todas las medidas de este experimento, se obtienen diferentes resultados en el entrenamiento dependiendo del valor μ . Concretamente, si se presta atención, por ejemplo, en la Figura 5.11, se observa que cuando se realiza el entrenamiento con una $\mu = 1$, el $ACPR_a$ se estabiliza en torno a los -51 dBc. Sin embargo, cuanto menor es dicho parámetro, menor $ACPR_a$ se obtiene en las últimas iteraciones, llegando a estar en unos -53 dBc cuando $\mu = 1/4$. A pesar de que las demás gráficas de ACPR presentan este mismo comportamiento, los valores no son los mismos debido a que, como se comprobó analizando el espectro de la señal, la respuesta obtenida en distintas bandas es diferente debido al comportamiento asimétrico del amplificador en cuestión.

Otro aspecto muy importante que destaca en todas las figuras obtenidas para este experimento son los valores medidos en la fase de validación. Si se comparan los resultados obtenidos con el experimento anterior,

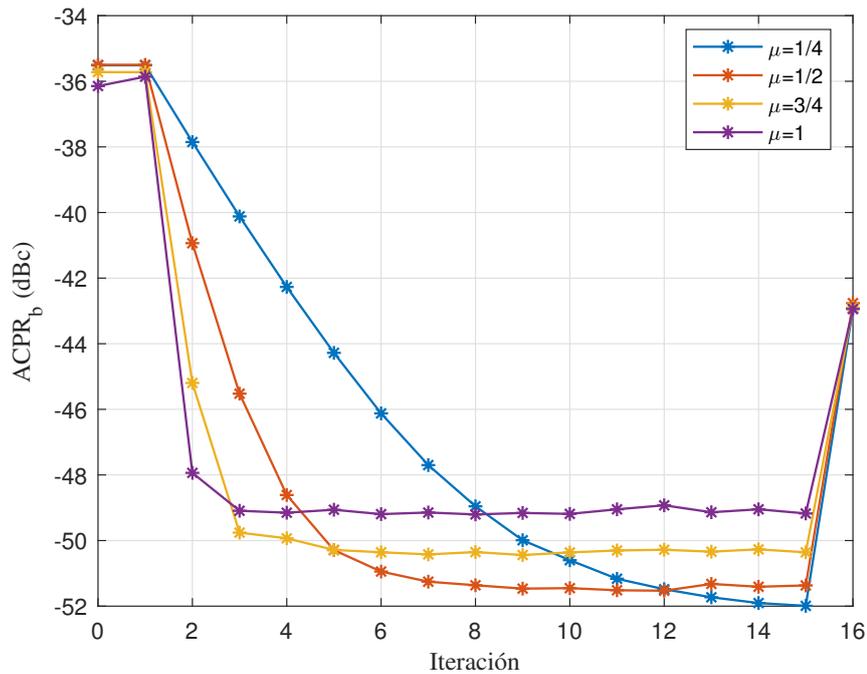


Figura 5.12 Evolución del ACPR del canal superior en el experimento 2.

se obtiene una clara diferencia en la fase de entrenamiento. Por ejemplo, comparando los valores de $ACPR_a$, mientras que en el experimento 1 se llega a -43 dB, en este se alcanzan valores de -53 dB. Como este experimento ofrece un ACPR de 10 dB menor respecto al anterior, se podría llegar a pensar que es mucho más conveniente utilizar este. Sin embargo, fijando la atención en los datos obtenidos en la validación, se comprueba que esto no es así, puesto que al cambiar la señal, como se hace en esta fase, el valor del ACPR (y de las demás medidas en las otras figuras) empeora hasta llegar a valores muy similares a los obtenidos en el experimento 1. Este fenómeno puede deberse a que este método produce un sobreajuste de los pesos, también denominado en inglés como *overfitting*. Este término se utiliza en el ámbito de la inteligencia artificial y *Machine Learning* cuando un modelo se ajusta demasiado a los datos de entrenamiento de forma que, al aplicar datos de validación, no se obtienen los resultados adecuados. Esto puede coincidir con lo que se observa en las medidas de este experimento donde, a pesar de que en el entrenamiento se obtienen valores muy buenos, al cambiar la señal, los resultados empeoran, llegando a ser similares a los del primer experimento.

Por último, en la Figura 5.14, llaman la atención los resultados respecto al resto de medidas ya que, mientras las demás ofrecen peores prestaciones cuando se valida con respecto al entrenamiento, cuando el valor de μ es 1, se obtiene una mejora, llegando a ofrecer los mismos resultados en validación que en los demás casos. Este efecto también ocurre ligeramente al fijar un valor de μ de $3/4$. Es bastante probable que este extraño comportamiento se produzca debido a la asimetría que se ha comprobado en las salidas del amplificador, puesto que este comportamiento solo ocurre para el $ACPR_{2b}$, en el que influyen las componentes frecuenciales más altas.

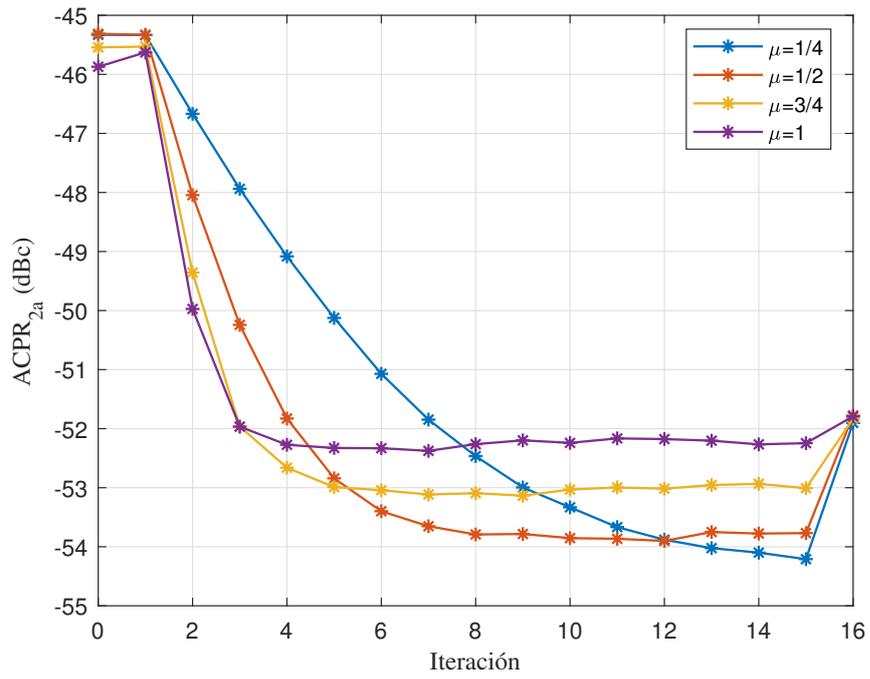


Figura 5.13 Evolución del ACPR_{2a} en el experimento 2.

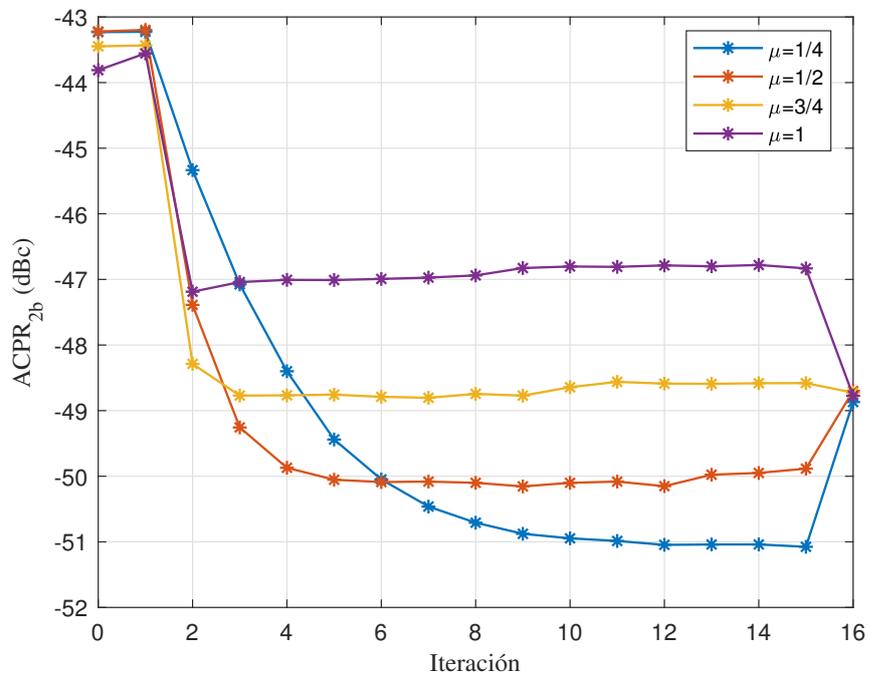


Figura 5.14 Evolución del ACPR_{2b} en el experimento 2.

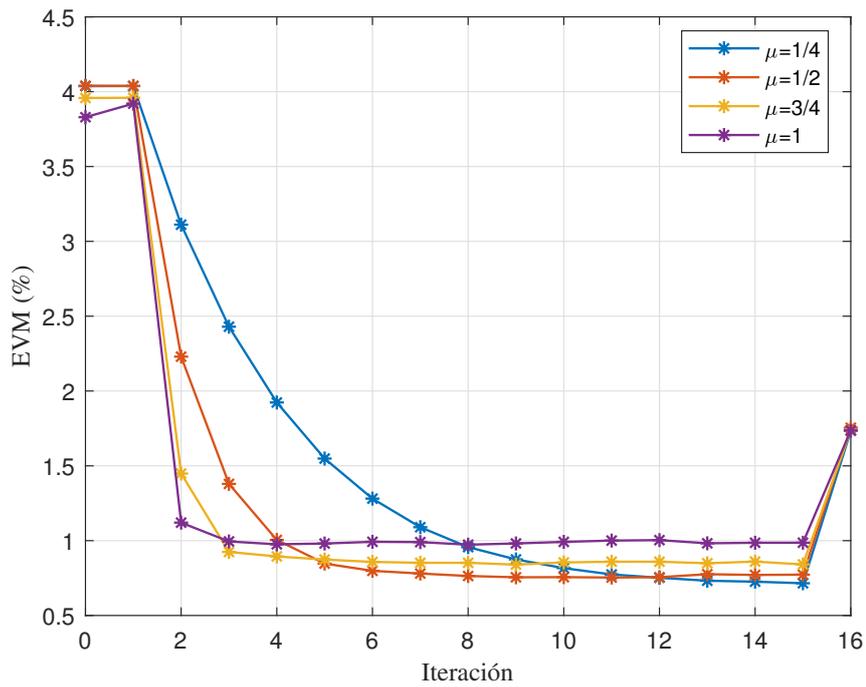


Figura 5.15 Evolución del EVM en el experimento 2.

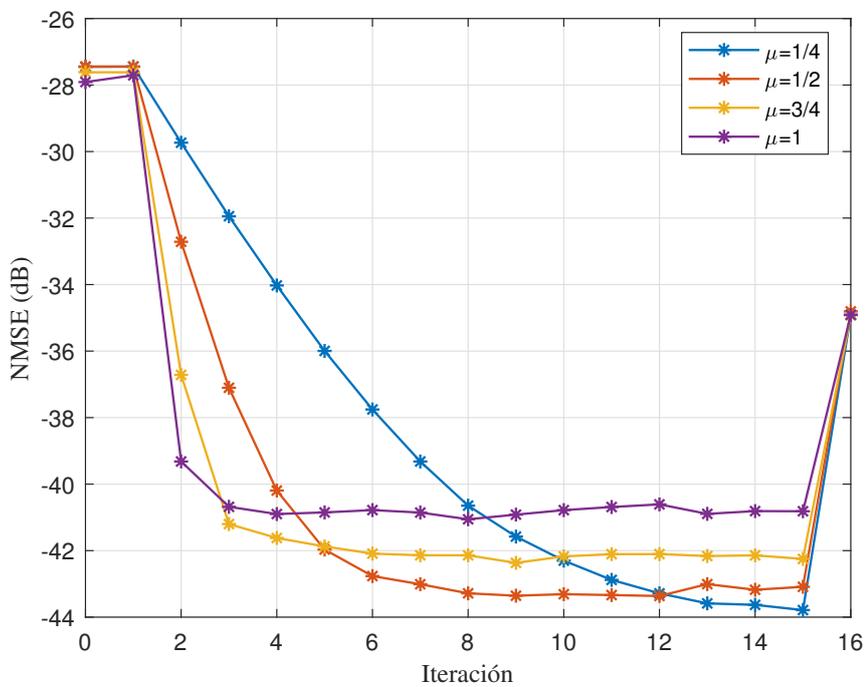


Figura 5.16 Evolución del NMSE en el experimento 2.

5.4 Aprendizaje directo con matriz de covarianza

Anteriormente se ha visto que existen diferentes modelos de comportamiento de amplificadores. Concretamente, muchos de estos modelos tienen como objetivo reducir la complejidad computacional a la hora de trabajar con ellos, por ejemplo, en algoritmos de aprendizaje directo. Del mismo modo, existen algoritmos de aprendizaje directo que buscan obtener buenos resultados de predistorsión, pero requiriendo un coste computacional menor a los usados como base. Un caso concreto de este tipo de algoritmos es el presentado en [20].

A continuación, se realizará una revisión de este método de aprendizaje directo para, luego, mostrar los resultados obtenidos en este trabajo tras implementarlo.

5.4.1 Funcionamiento del algoritmo

El método introducido en [20] se puede analizar partiendo de la base de arquitectura de aprendizaje explicada en el primer algoritmo. Concretamente, introduce una ecuación similar a la Ecuación (5.6), ya presentada en la sección anterior para realizar el cálculo de los coeficientes del DPD. El término $(\mathbf{U}^H \mathbf{U})^{-1}$ de esta ecuación requiere de un gran coste computacional. Por este motivo, en [20] se plantea como objetivo evitar dicho cálculo y, para ello, se parte del razonamiento de que la señal de entrada al amplificador es estacionaria y ergódica de modo que, cuando el número de muestras M de dicha señal usadas para el cálculo de la matriz \mathbf{U} tiende a infinito, la matriz $\mathbf{U}^H \mathbf{U} / M$ tiende a su matriz de covarianza. Esto se muestra más claramente en la Ecuación (5.9), donde la matriz \mathbf{C}_U denota la matriz de covarianza de \mathbf{U} y que, además, es una matriz de valores constantes.

$$\mathbf{C}_U = \lim_{M \rightarrow \infty} \mathbf{U}^H \mathbf{U} / M \quad (5.9)$$

El hecho de que esta matriz de covarianza sea constante permite que su inversa sea calculada una única vez. Así, en [20] se define la inversa de esta matriz como

$$\mathbf{V} = \mathbf{C}_U^{-1} = \lim_{M \rightarrow \infty} (\mathbf{U}^H \mathbf{U} / M)^{-1} \quad (5.10)$$

y como, a diferencia de esta matriz, el otro término de la ecuación, $\mathbf{U}^H (\tilde{\mathbf{y}} - \mathbf{u})$, sí que varía para distintas iteraciones, se define finalmente la ecuación de actualización propuesta:

$$\mathbf{w}_{i+1} = \mathbf{w}_i + \mu \frac{\mathbf{V} \mathbf{U}^H (\tilde{\mathbf{y}} - \mathbf{u})}{M} \quad (5.11)$$

La solución que se plantea en este método está pensada cuando el número de muestras M es prácticamente infinito. Evidentemente, trabajar con infinitas muestras es inviable y, por este motivo, es necesario conocer un número de muestras finito para el cual se sigue cumpliendo esta hipótesis. En [20], se realiza una demostración para llegar a la condición de convergencia que se corresponde con la Ecuación (5.12), donde σ^2 representa la varianza y \mathbb{E} la función valor esperado. Además N representaría el orden no lineal de la matriz \mathbf{U} y H el número de funciones base del modelo.

$$M \geq \frac{H \times \sigma_{|x|^{2N}}^2}{\mathbb{E}^2(|x|^{2N})} \quad (5.12)$$

En la Figura 3 de [20], también se aporta una gráfica para obtener una estimación del término $\frac{\sigma_{|x|^{2N}}^2}{\mathbb{E}^2(|x|^{2N})}$ en función de la PAPR y el orden del modelado. Particularizando esta ecuación para el caso de este trabajo, como se tiene una PAPR de 10,5 dB, aproximadamente, y el modelo es de orden 13, se podría aproximar el término $\frac{\sigma_{|x|^{2N}}^2}{\mathbb{E}^2(|x|^{2N})}$ al valor de 10^4 . No obstante, esta ecuación puede ser calculada también analíticamente mediante el estimador de la varianza y la media al contar con un gran número de muestras de la señal. De este modo, siguiendo la Ecuación (5.12) y utilizando la propia señal se llega a la conclusión de que, para esta señal, se debe cumplir $M \geq 8,156 \cdot 10^3 \times H$. Asimismo, el modelo usado en este caso corresponde a un valor de $H = 81$. Por tanto, se obtiene que la condición de convergencia en este caso será $M \geq 660.530$, por

lo que se debería usar un número mayor de muestras con este método para evitar problemas. En todos los experimentos de este trabajo se han empleado un total de 800.000 muestras, lo cual implica el número de muestras es 1,21 veces mayor que el mínimo, por lo que la condición de convergencia no será un problema para obtener buenos resultados.

5.4.2 Experimento 3: Análisis de las prestaciones usando la matriz de covarianza

Este experimento se ha realizado para comprobar las prestaciones que ofrece este método. Como el efecto de los diferentes valores de μ se aprecian de forma adecuada en el experimento 2, en este caso se aportan los resultados obtenidos para $\mu = 1$ y para $\mu = 1/2$. En el primer caso, se ha utilizado un total de 5 iteraciones en el aprendizaje puesto que, como se comprueba también en los anteriores experimentos, con dicho valor, no se necesita un número elevado de iteraciones para que el algoritmo converja. Para el caso de $\mu = 1/2$, el aprendizaje puede ser un ligeramente más lento que para $\mu = 1$, por lo que se ha fijado un total de 7 iteraciones de entrenamiento para asegurar la convergencia del algoritmo. El bucle correspondiente a este experimento se incluye en el Código 5.3.

Código 5.3 Bucle del experimento 3.

```

1  %Inicialización de parámetros
2  N_iter=5;
3  mu=1;
4  [N,f]=size(U);
5
6  %Inicialización de los pesos a 0
7  w = zeros(f,1);
8
9  %Matriz de covarianza
10 cov_U=cov(U);
11 %Inversa de la covarianza
12 V=inv(cov_U);
13
14 %Proceso de aprendizaje
15 for iter=1:N_iter
16   %Señal predistorsionada
17   x1 = u-U*w;
18
19   %Medida
20   [yPA, xPA, ~] = myPACHalmers(x1, RMSin, Nmeas);
21
22   %Normalización de la y
23   yPAnorm = yPA*norm(xPA)/norm(yPA);
24
25   %Diferencia entre la señal de salida normalizada y la entrada
26   e1 = yPAnorm-u;
27
28   %Actualización de los pesos
29   w = w + mu * V * U' * e1/N;
30 end

```

En este caso, los resultados han sido recogidos en la Tabla 5.2. En la primera columna se encuentran los parámetros medidos, que son los mismos que los de los experimentos anteriores. En la segunda columna se aportan datos obtenidos en la primera medida en la que no se aplica predistorsión y en las dos últimas se han recogido los datos obtenidos para $\mu = 1/2$ y $\mu = 1$ en la fase de validación, puesto que en este caso no ocurre el problema de sobreajuste visto en el experimento 2 y, por tanto, se obtienen resultados similares para la validación con respecto a la última iteración del entrenamiento.

Como se puede comprobar, los resultados obtenidos son muy similares a los de la fase de validación de los experimentos anteriores. Los valores de ACPR de los primeros canales adyacentes mejoran en unos 7 dB, a

Tabla 5.2 Resultados del experimento 3.

Parámetro	Sin DPD	$\mu = 1/2$	$\mu = 1$
ACPR _a (dBc)	-36	-43,2	-43,2
ACPR _b (dBc)	-35,5	-43,5	-43,5
ACPR _{2a} (dBc)	-45,2	-52,2	-52,1
ACPR _{2b} (dBc)	-43,1	-49,3	-49,1
EVM (%)	4,0	1,6	1,6
NMSE (dB)	-27,4	-35,5	-35,5

excepción del ACPR_{2b}, que presenta una mejora de 6 dB. Es importante recordar que el comportamiento del amplificador es asimétrico y que, en todos los experimentos, los resultados en esta banda varían respecto a las otras bandas. Aún así, sigue siendo una buena mejora respecto al caso sin predistorsión. Respecto al EVM, se ha conseguido una mejora desde el 4% hasta unos 1,6%. Finalmente, el NMSE ha obtenido una mejora de unos 8 dB respecto al caso sin DPD.

Es importante tener en cuenta que en este método no se ha utilizado la matriz pseudoinversa de \mathbf{U} en ningún momento, puesto que se ha trabajado con la matriz de covarianza de la misma. La propuesta que se hacía en este método implicaba algunas aproximaciones y usar la estadística de la señal con el objetivo de reducir el coste computacional y, al obtener medidas similares a los del caso original, en el que sí se utiliza la pseudoinversa, se puede afirmar que se han conseguido resultados positivos.

5.4.3 Covarianza de Tyler

A lo largo del desarrollo de este trabajo, también se han realizado numerosas pruebas, muy similares a este experimento, pero calculando la matriz de covarianza mediante el estimador de Tyler en vez de la matriz de covarianza común, usada en el experimento anterior. Este estimador es empleado en una amplia variedad de campos actualmente [21], [22], por lo que se decidió probar su funcionamiento en este tipo de experimentos.

La definición de la matriz de covarianza de un vector o una matriz es comúnmente descrita como el producto de sí mismo por su transpuesto o hermítico en caso de ser complejo, dividido entre el número de muestras M . Este producto puede expresarse también en forma de sumatorio, como se muestra en la siguiente ecuación.

$$\mathbf{C}_U = \mathbf{U} \cdot \mathbf{U}^H / M = \frac{1}{M} \sum_{i=1}^M \mathbf{U}_i^H \mathbf{U}_i \quad (5.13)$$

El estimador de Tyler consiste en una forma de calcular la matriz de covarianza. Esta consiste en aplicar un número determinado de iteraciones de la Ecuación (5.14) hasta llegar a un estado de convergencia.

$$\mathbf{C}_U^{(k)} = \frac{1}{M} \sum_{i=1}^M \frac{\mathbf{U}_i^H \mathbf{U}_i}{\mathbf{U}_i^H \mathbf{C}_U^{(k-1)} \mathbf{U}_i} \quad (5.14)$$

Este método iterativo permite que si existe una muestra en el vector, en este caso la matriz \mathbf{U} , muy poco correlacionada con el resto de muestras, su efecto quedará anulado tras algunas iteraciones.

Aunque se ha considerado apropiado incluir este breve apartado para indicar que se han realizado numerosas pruebas, lamentablemente no es posible adjuntar buenos resultados ya que, al cambiar la matriz de covarianza común por la obtenida mediante el estimador de Tyler, no ha sido posible obtener un comportamiento adecuado puesto que, al aplicar la predistorsión, no solo no mejoraba la señal, sino que la señal predistorsionada llegaba a valores de PAPR exageradamente elevados. Esto podría haber sucedido debido a problemas numéricos que no llegaron a ser detectados. Por este motivo, a pesar de recurrir a diferentes técnicas como aplicar CFR o la normalización de la matriz, que sí ofrecen resultados coherentes, como se verá en experimentos posteriores, no fue posible obtener una buena respuesta usando dicho estimador.

5.5 Aprendizaje directo con selección de componentes

5.5.1 Introducción al experimento

En la Sección 3.4 se adelantaba que cuando se trabaja con modelos basados en series de Volterra, estos suelen contar con un número de términos suficientes como para que, al tratar con este tipo de datos, se requiera un coste computacional elevado. En la misma sección, se presentaron dos técnicas que permiten reducir el número de componentes presentes en un modelo para facilitar el procesamiento del mismo. Concretamente, se introdujo la técnica OMP para, seguidamente, explicar con mayor detalle el algoritmo de la técnica denominada DOMP.

En este caso, se va a realizar un experimento en el que se aplica la técnica DOMP a la arquitectura de aprendizaje directo para, posteriormente, comparar los resultados frente a los obtenidos sin selección de componentes. Como la propia técnica consiste en reducir el número de componentes del modelo que se utilizará en el experimento, en principio, se espera que los resultados no sean mejores que sin DOMP. Sin embargo, será de gran interés comprobar si los resultados son demasiado peores o, por el contrario, sigue ofreciendo un buen comportamiento, a la vez que se mejora el coste computacional gracias a dicha técnica.

5.5.2 Experimento 4: Análisis del efecto de la selección de componentes

Con el objetivo de comparar los resultados obtenidos realizando la selección de componentes mediante el uso del algoritmo DOMP con los recogidos en el experimento 1, sin aplicar este tipo de técnica, se aportan las gráficas correspondientes a las diferentes medidas, tanto en entrenamiento como en validación, para los mismos valores de μ que los de dicho experimento. Es importante aclarar que, para llevar a cabo la fase de validación, la selección de componentes también debe ser aplicada a la matriz U_2 de la señal de validación. Sin embargo, el algoritmo DOMP es aplicado únicamente una vez con la señal de entrenamiento, de modo que los componentes de la segunda matriz serán seleccionados usando los índices de la primera matriz. En este caso, el bucle de entrenamiento es idéntico al del Código 5.1, cambiando únicamente la matriz U , la cual deberá ser resultado de aplicar DOMP y, por tanto, de dimensiones menores que la original. Concretamente, el algoritmo DOMP reduce el número de columnas de la matriz U , es decir, el número de regresores. Aunque para diferentes ejecuciones el número exacto puede variar, se ha comprobado que, partiendo de la matriz U original, se llega a obtener una reducción de casi la mitad, pasando de las 81 columnas originales a entre 42 y 46, lo que significa que se consigue reducir hasta la mitad, aproximadamente.

Las gráficas correspondientes a los resultados de este experimento se incluyen a continuación. Realizando una comparación con las obtenidas en el primer experimento, se puede decir, en primer lugar, que la aplicación del algoritmo DOMP no tiene influencia en el desempeño del entrenamiento en términos de prestaciones, puesto que el número de iteraciones requeridas para que el algoritmo converja es similar.

Por otro lado, se puede comprobar que los resultados obtenidos, tanto en entrenamiento como en validación son también muy similares a los recogidos sin usar DOMP. Sería posible mencionar alguna mínima reducción en el ACPR pero, teniendo en cuenta que la diferencia es de pocas décimas y que las unidades son decibelios, se puede considerar como una diferencia despreciable. No obstante, estos resultados son positivos puesto que el algoritmo DOMP se encarga de reducir el número de componentes que se utilizan en el modelo del amplificador lo que podría significar una pérdida en la precisión del mismo. Sin embargo, a pesar de haber aplicado esta técnica, las medidas demuestran que no afectan negativamente en el entrenamiento de un DPD con DLA sino que, por el contrario, el comportamiento es similar, además de producirse una reducción del coste computacional del proceso.

Aunque los resultados de las medidas pueden ser similares y, por tanto, parecer que no habría ningún tipo de ventaja en reducir el número de componentes, en el experimento 6 se realizará un análisis sobre la estabilidad y los tiempos de ejecución al aplicar esta técnica, frente al resto de casos.

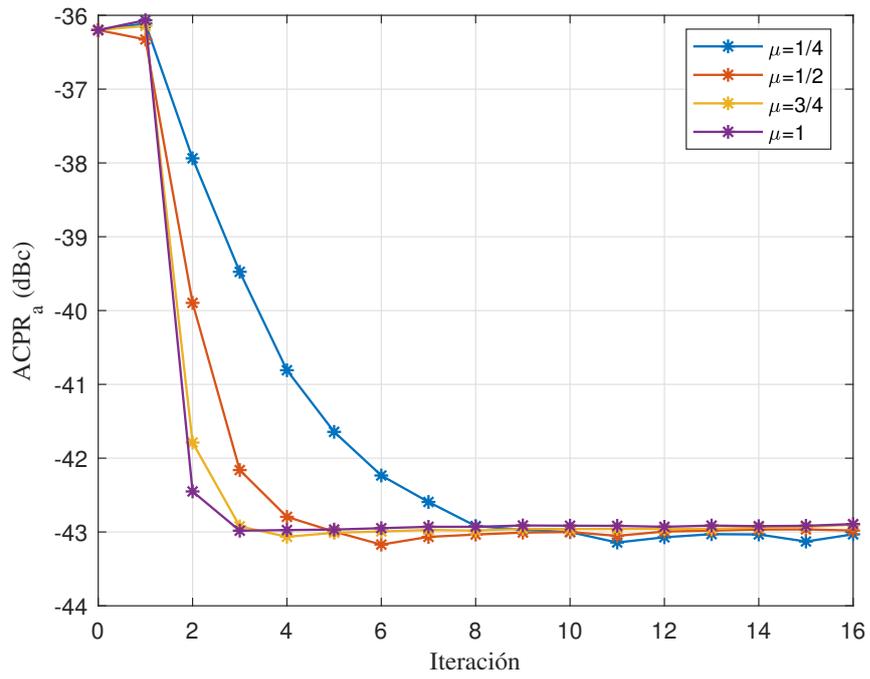


Figura 5.17 Evolución del ACPR del canal inferior en el experimento 4.

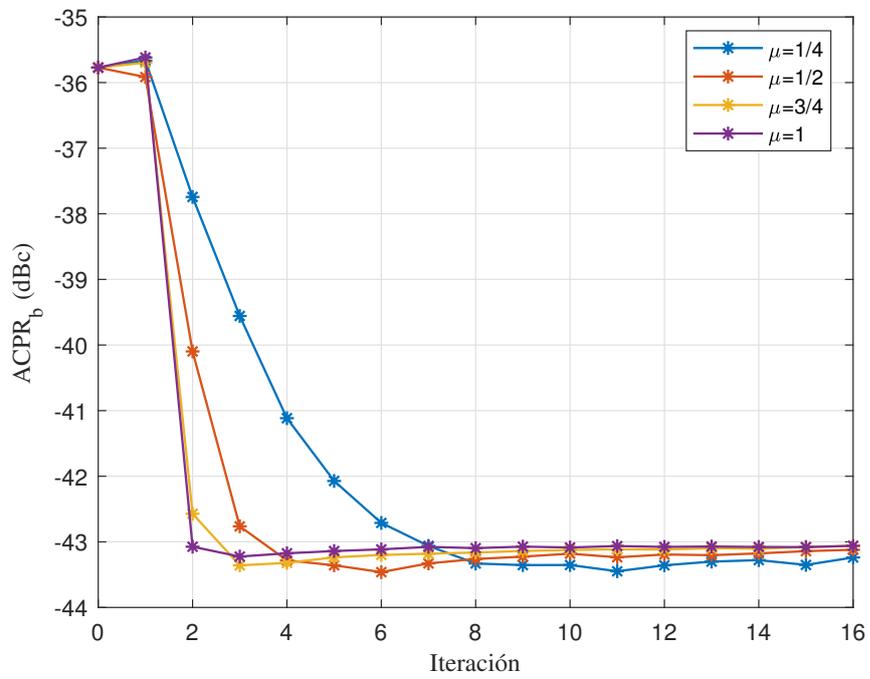


Figura 5.18 Evolución del ACPR del canal superior en el experimento 4.

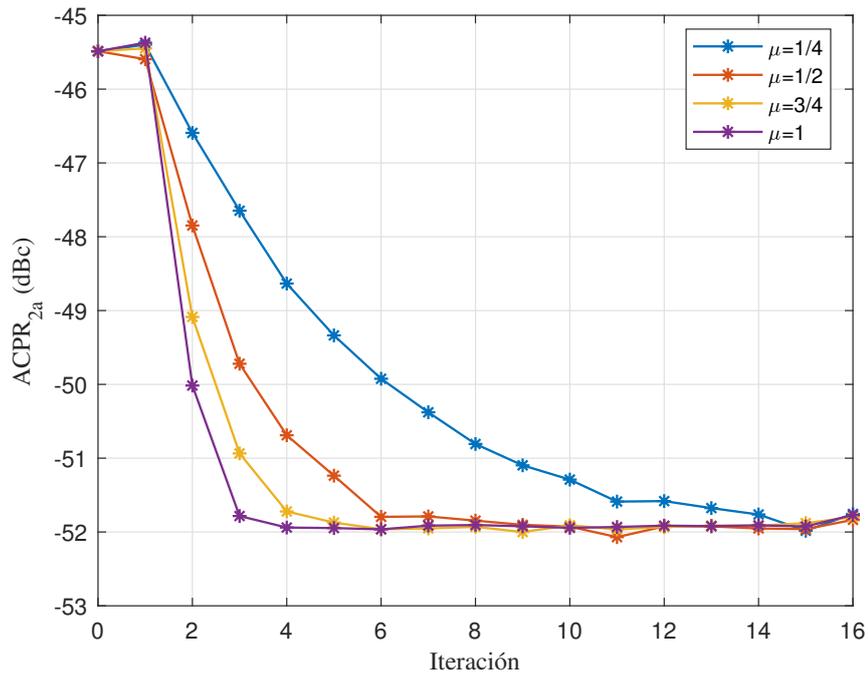


Figura 5.19 Evolución del $ACPR_{2a}$ en el experimento 4.

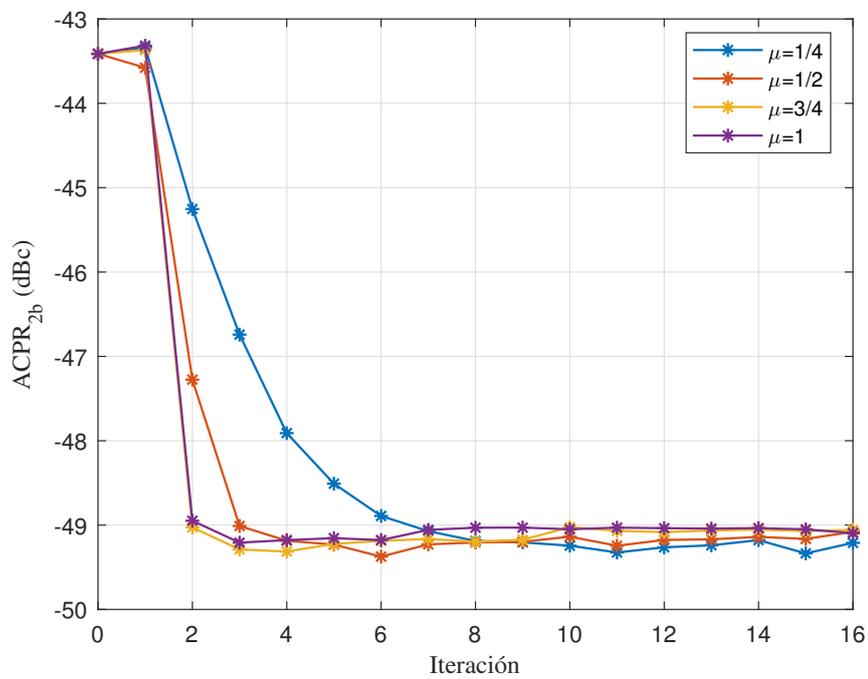


Figura 5.20 Evolución del $ACPR_{2b}$ en el experimento 4.

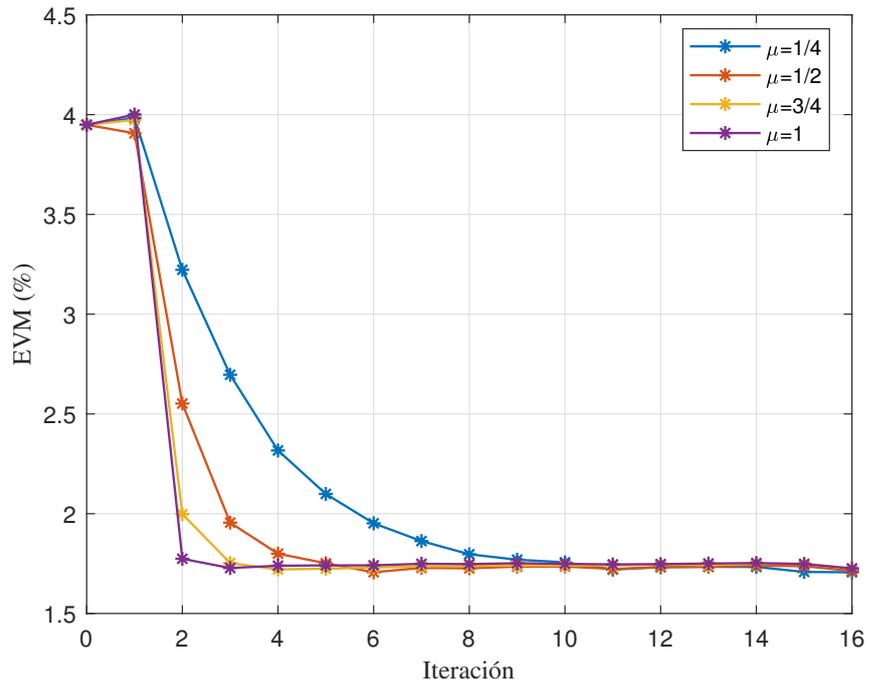


Figura 5.21 Evolución del EVM en el experimento 4.

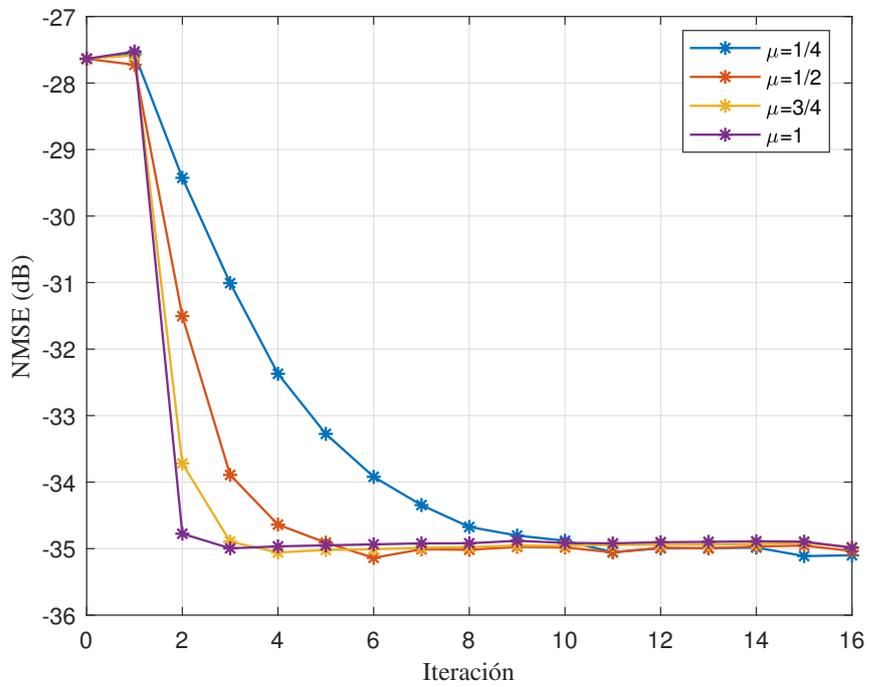


Figura 5.22 Evolución del NMSE en el experimento 4.

5.6 Aplicación de CFR y normalización

5.6.1 Técnicas utilizadas

En la Sección 4.2 se argumentaba que en un sistema en cascada compuesto por un DPD y un amplificador podría presentar problemas con determinados valores de PAPR de la señal de entrada, provocando la saturación del PA. Por este motivo, se comentó que una de las formas de reducir la PAPR de una señal es aplicando técnicas de reducción del factor de cresta o CFR. De este modo, se ha decidido realizar un experimento en el que se aplique CFR a la señal que se envía al amplificador, teniendo como objetivo comprobar la repercusión del mismo al emplear aprendizaje directo para el diseño de un DPD. Existe una gran variedad de técnicas de CFR, yendo desde un corte directo en ciertos instantes donde la señal supere determinados valores hasta técnicas más complejas [13]. En este caso, el CFR aplicado tendrá una respuesta suave, ya que las técnicas conocidas como *hard clipping* suelen presentar un peor desempeño que las de *soft clipping*.

Por otro lado, en el experimento correspondiente con esta sección, también se pondrá en cuestión la normalización de las señales. En muchas ocasiones, cuando se trabaja con algoritmos de tratamiento de señal de diferentes formas, suele ser recomendable realizar los procesos con las señales previamente normalizadas. Esto es debido a que en algunas ocasiones se pueden dar problemas numéricos, producidos por la manipulación de señales de potencia muy elevada o muy reducida. Por ello, también se comprobará en este experimento el efecto de normalizar la matriz \mathbf{U} antes de calcular su pseudoinversa y, cuando haya que usar esta, se desnormaliza.

Finalmente, este experimento tendrá en cuenta una última comprobación. En la implementación del método original en MATLAB®, el cálculo de la pseudoinversa de la matriz \mathbf{U} ha sido realizado haciendo uso de la función `pinv` que se aporta en las librerías de MATLAB®. Sin embargo, esta función utiliza un modo “económico” y fija ciertos valores de tolerancia a partir de los cuales, se desprecian algunos datos. En determinadas situaciones, esto puede ser perjudicial, provocando la pérdida de precisión en algunos algoritmos. Por ello, en este experimento, también se tratará el caso de no recurrir a dicha función y utilizar la definición de pseudoinversa de Moore-Penrose, es decir, calculando $(\mathbf{U}^H\mathbf{U})^{-1}\mathbf{U}^H$ directamente.

5.6.2 Experimento 5: Aplicando CFR y normalizando

En este experimento se ha fijado el factor de ponderación $\mu = 1$ ya que en este caso no será necesario observar su influencia. Por otro lado, se ha establecido un total de 10 iteraciones de entrenamiento, por lo que en las gráficas que se incluirán a continuación se contará con una iteración inicial, indicada como 0, sin DPD, 10 iteraciones de entrenamiento y una undécima con los resultados de la validación.

En este caso, también se aportará una gráfica por cada una de las figuras de mérito utilizadas para caracterizar las medidas. Sin embargo, como en este caso el valor de μ siempre es el mismo, en cada figura se diferenciarán varias curvas pero esta vez se corresponderán con:

- En azul, se mostrará la curva de entrenamiento obtenida empleando el método original, es decir, el del experimento 1 sin aplicar técnicas de CFR, normalización ni ninguna otra modificación.
- En naranja, se mostrará la curva de entrenamiento en el caso de aplicar CFR a la señal. Tanto en este como en los demás experimentos en los que se ha aplicado CFR, se ha fijado una *Peak to Average Power Ratio* (PAPR) deseada de 9 dB.
- En amarillo irá representada la curva correspondiente al caso de aplicar tanto CFR como la normalización.
- Los resultados obtenidos aplicando la expresión completa de la pseudoinversa y la técnica CFR irán en un tono morado.
- Finalmente, para el caso de aplicar CFR, normalización y usar la expresión completa de la pseudoinversa, se representará con color verde.

El Código 5.4 se corresponde con las líneas utilizadas para implementar en MATLAB® el presente experimento, donde se han incluido las diferentes formas probadas en el mismo como, por ejemplo, en el cálculo de la pseudoinversa. La aplicación de la técnica de CFR se realiza inmediatamente tras la generación de las señales utilizando funciones auxiliares. Como el detalle de dichas funciones no es objetivo de este trabajo, no se incluye en el presente código.

Código 5.4 Bucle del experimento 5.

```

1 %Inicialización de parámetros
2 Niter_DL=10;
3 mu=1; %Factor de ponderación
4 [N,f]=size(U);
5
6 %Inicialización de los pesos a 0
7 w = zeros(f,1);
8
9 %Cálculo de la pseudoinversa
10 pinv_U=pinv(U);
11 %pinv_U=inv(U'*U)*U';
12 %pinv_U=(U'*U)\U';
13
14 %Normalización de la pseudoinversa
15 normU = vecnorm(U);
16 Un = U*diag(normU.^-1);
17 pinv_Un=pinv(Un);
18 %pinv_Un=(inv(Un'*Un)*Un');
19
20 %Proceso de aprendizaje
21 for iter=1:Niter_DL
22     %Señal predistorsionada
23     x1 = u-U*w;
24
25     %Medida
26     [yPA, xPA, ~] = myPACHalmers(x1, RMSin, Nmeas);
27
28     %Normalización de y
29     yPAnorm = yPA*norm(xPA)/norm(yPA);
30
31     %Diferencia entre la señal de salida normalizada y la entrada
32     e1 = yPAnorm-u;
33
34     %Actualización de los pesos
35     dw1 = pinv_U*e1;
36     w = w + mu * dw1;
37 end

```

Como los resultados son ligeramente distintos entre las diferentes características medidas, se analizará cada uno de estos parámetros individualmente. En la Figura 5.23 se ha representado el $ACPR_a$ y en la Figura 5.24 el $ACPR_b$. En ambas medidas se obtiene un resultado similar. En primer lugar, se observa que al aplicar CFR, el ACPR ha empeorado ligeramente respecto al caso original aunque, teniendo en cuenta que se trata de unidades logarítmicas, la diferencia es muy pequeña. Por otro lado, al añadir también la normalización de la matriz U , se aprecia una mejora de unos 0,3 dB. Por último, se debe comentar el resultado de estos dos últimos casos al utilizar la expresión de la pseudoinversa en lugar de la función `pinv` de MATLAB®. En estos casos, aunque la mejora es muy pequeña, se trata de los mejores resultados obtenidos en este experimento, llegando a los $-43,3$ dBc en el $ACPR_a$ y $-43,6$ dBc en el $ACPR_b$, al emplear CFR y la normalización.

Los resultados de los valores de ACPR para los segundos canales adyacentes se muestran en la Figura 5.25 y en la Figura 5.26. En ambos casos, la peor medida es la original, mientras que con CFR se obtiene una ligera mejora. Si además del CFR también se aplica la normalización, se obtiene una ligera mejora del $ACPR_{2a}$ de 0,2 dB y de 0,3 dB de $ACPR_{2b}$ respecto al caso sin normalizar. Finalmente, se podría afirmar que el hecho utilizar la expresión completa de la pseudoinversa no ofrece una mejora del ACPR del segundo canal, puesto que las medidas de validación obtenidas presentan diferencias despreciables con las que usan la función `pinv`.

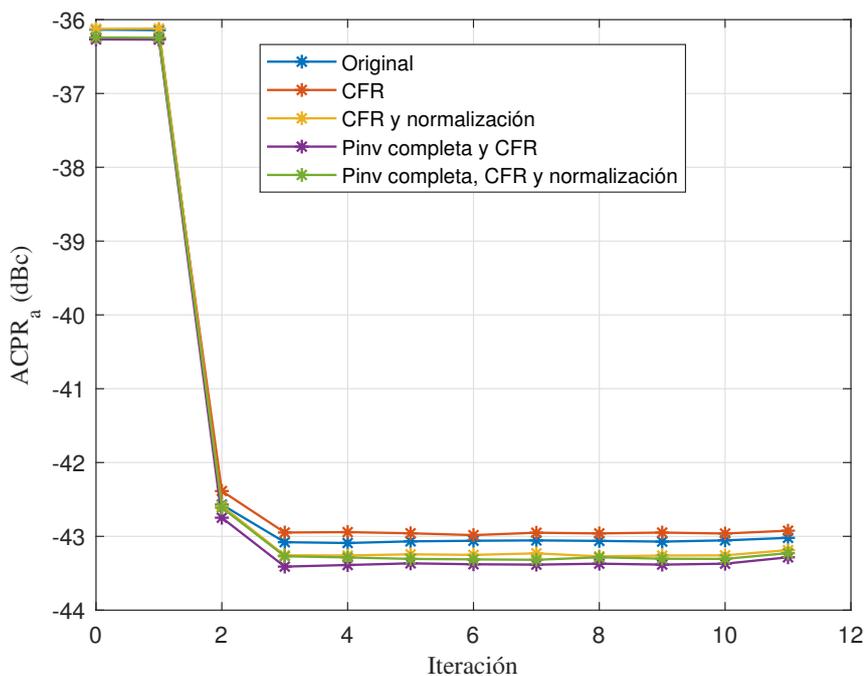


Figura 5.23 Evolución del ACPR del canal inferior en el experimento 5.

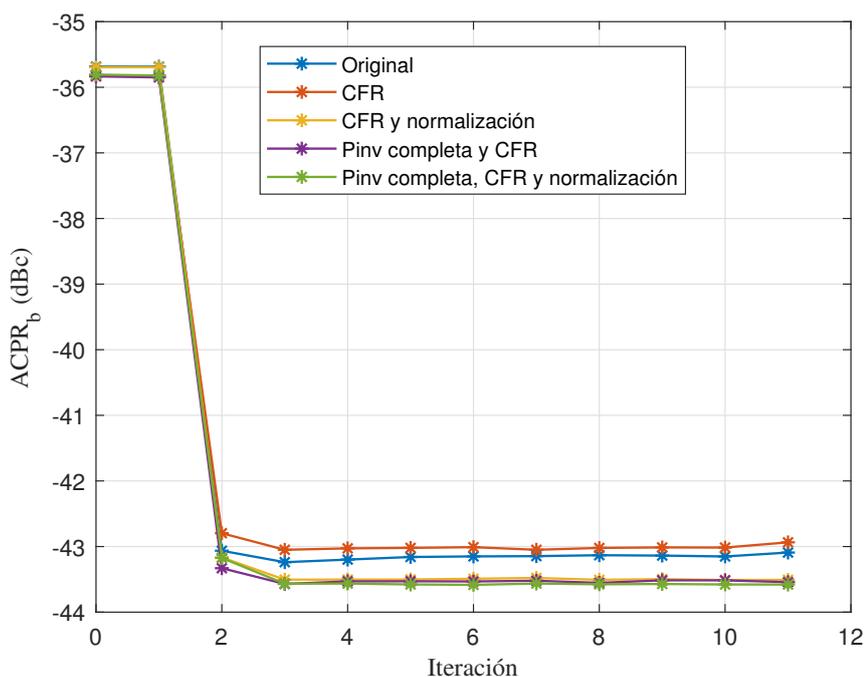


Figura 5.24 Evolución del ACPR del canal superior en el experimento 5.

En la Figura 5.27 se ha representado el EVM. En esta medida, los resultados de aplicar CFR son prácticamente idénticos a los del caso original, por lo que se podría decir que aplicar esta técnica no tiene efectos en el EVM. Sin embargo, hay una mejora de 0,1% respecto al caso original cuando se aplica también la normalización. Al usar la expresión completa de la pseudoinversa, los resultados son muy similares a los de CFR y normalización.

La última gráfica de este experimento se corresponde con la Figura 5.28, que muestra los resultados del NMSE. Aunque es una figura de mérito distinta el comportamiento es muy parecido a las anteriores. En caso de usar el método original, se consigue pasar de valores por encima de los -28 dB hasta los -35 dB. Al añadir CFR, se obtiene un NMSE de $-35,3$ dB y al añadir la normalización se consigue llegar casi a los

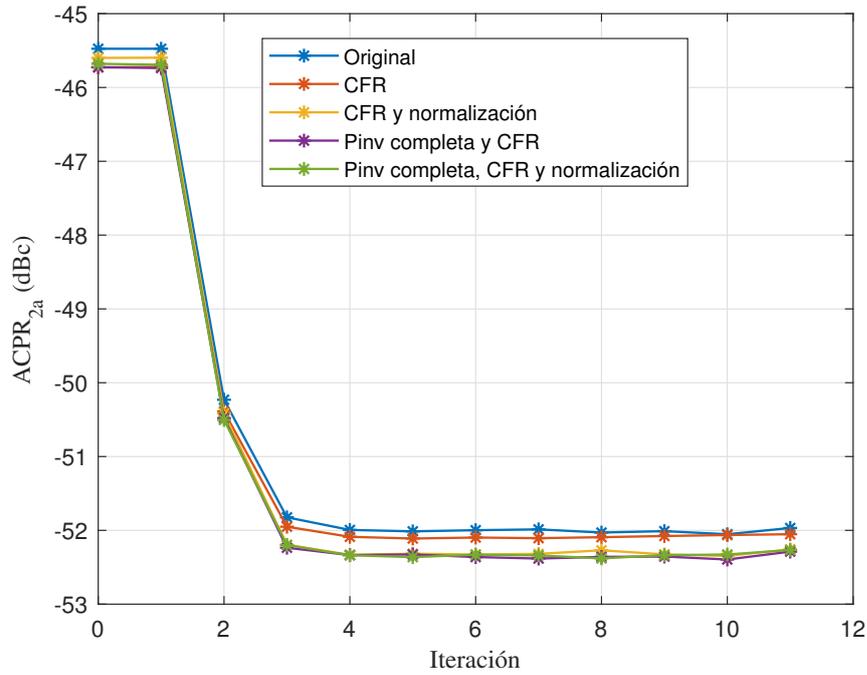


Figura 5.25 Evolución del $ACPR_{2a}$ en el experimento 5.

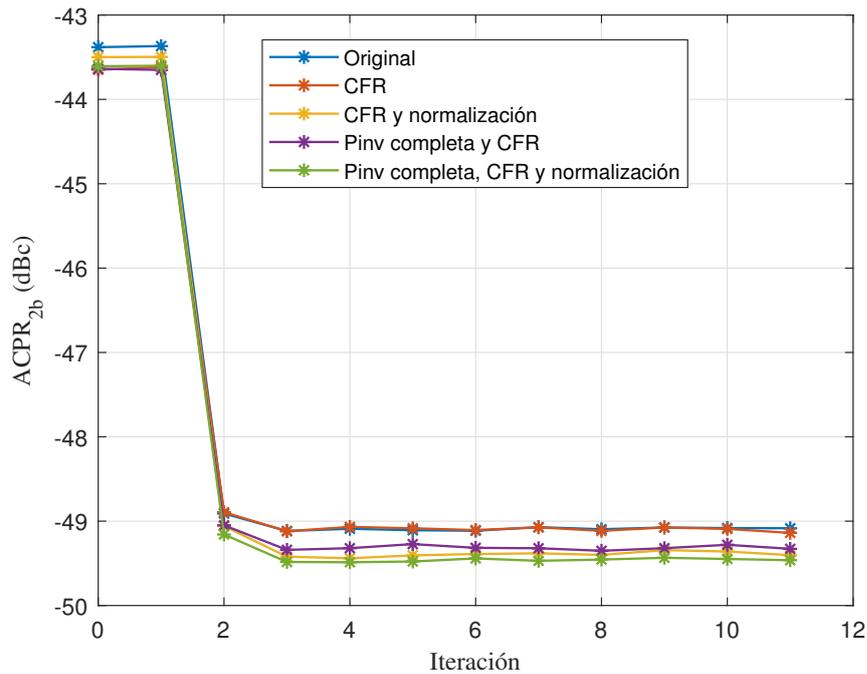


Figura 5.26 Evolución del $ACPR_{2b}$ en el experimento 5.

-35,9 dB. Por último, en los casos de usar CFR con la expresión completa de la pseudoinversa y, luego, añadiendo la normalización, se obtienen valores que superan los -35,9 dB.

De todos los resultados, se puede comprobar que añadiendo únicamente la técnica de CFR al método original no se obtienen mejoras reseñables. Sin embargo, si se trabaja con la U normalizada, sí que se aprecia una ligera mejora respecto al caso de trabajar sin normalización.

Finalmente, también se puede llegar a la conclusión de que suele ser conveniente aplicar algunas de estas modificaciones ya que se suelen obtener mejores prestaciones que en el caso de no aplicarlas. Aunque la diferencia no siempre es muy grande, en determinados casos, medio decibelio puede suponer una gran diferencia. Además, si se mejorase el propio algoritmo de aprendizaje, se podrían obtener incluso mejores

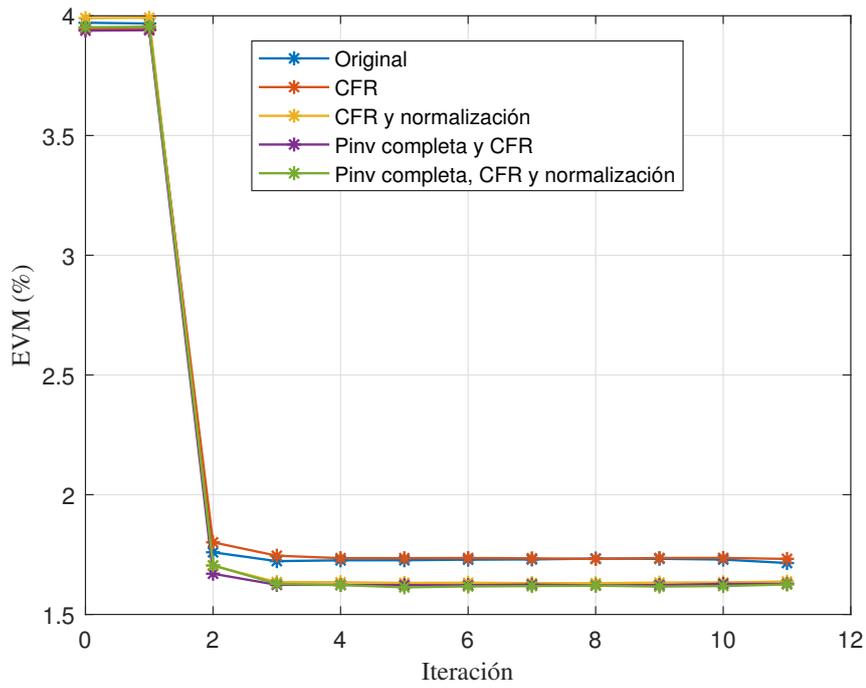


Figura 5.27 Evolución del EVM en el experimento 5.

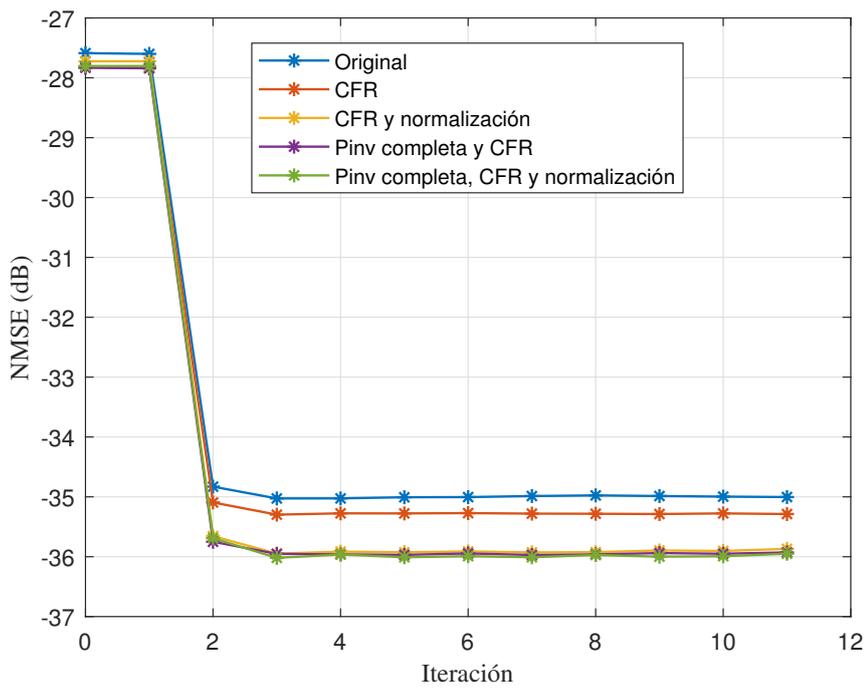


Figura 5.28 Evolución del NMSE en el experimento 5.

resultados incluyendo estas modificaciones.

5.7 Pruebas de estabilidad y coste computacional

5.7.1 Introducción al experimento

En esta última sección se tratará de analizar cómo se comportan algunos de los métodos utilizados anteriormente en términos de coste computacional y de estabilidad. Concretamente, se recurrirá al cálculo de dos parámetros distintos. Por un lado, se calculará el número de condicionamiento y, por otra parte, se obtendrá tiempo requerido por el método para realizar el aprendizaje, lo cual tiene una estrecha relación con el coste computacional.

En primer lugar, es necesario definir el número de condicionamiento. Este se trata de la relación entre el mayor y el menor autovalor de una matriz y sirve para estimar si un sistema es inestable o no. Concretamente, el número de condicionamiento permite medir cómo de sensible es una matriz a una inversión, siendo más importante su orden de magnitud que su propio valor, puesto que servirá como aproximación para conocer el grado de variación de la inversa respecto al de la propia matriz. Cuanto más cercano esté el número de condicionamiento de una matriz a 1, mejor precisión se tendrá a la hora de calcular su inversa. En este caso, la matriz sería *well-conditioned*. Por otro lado, cuanto mayor sea dicho número, el cómputo de la inversa será más propenso a grandes errores numéricos, siendo la matriz *ill-conditioned*. De hecho, una matriz no invertible tendrá un número de condicionamiento infinito.

Por otro lado, como se ha comentado, se medirá el tiempo de ejecución para cada algoritmo. Esto se realizará haciendo uso de las instrucciones `tic` y `toc` de MATLAB®. En este estudio, se eliminará el tiempo empleado en enviar y recibir las medidas mediante el WebLab, ya que este proceso ocupa gran parte del tiempo y puede ser variable en diferentes ocasiones, provocando la alteración de los resultados. En este experimento no se aportarán fragmentos de código para facilitar la lectura de esta sección, puesto que simplemente se requiere añadir las instrucciones para medir el tiempo a códigos mostrados anteriormente.

5.7.2 Experimento 6: Número de condicionamiento y tiempo de aprendizaje

Este último experimento, se puede dividir en dos pruebas distintas. En primer lugar, se ha calculado el número de condicionamiento para los diferentes métodos utilizados en este trabajo. Esto es realizado utilizando la instrucción `cond` de MATLAB®, pasando como parámetro de entrada la matriz que se desea invertir.

En el primer método, al igual que en el segundo, la matriz cuya pseudoinversa se calcula es directamente la matriz \mathbf{U} . Además, la definición de pseudoinversa se puede desglosar, de modo que se puede calcular el número de condicionamiento únicamente de la matriz cuya inversa debe obtener, que es el producto $\mathbf{U}^H \mathbf{U}$.

En el tercer método, el objetivo es evitar calcular la pseudoinversa de la matriz \mathbf{U} recurriendo a su matriz de covarianza en el entrenamiento del DPD. Por tanto, el número de condicionamiento será medido en este caso con la matriz de covarianza de dicha matriz.

En el caso de aplicar la técnica de selección de componentes DOMP, también se calcula la pseudoinversa de la matriz \mathbf{U} . Sin embargo, el número de condicionamiento debe ser calculado una vez que se ha reducido el número de componentes gracias a esta técnica, obteniendo una matriz \mathbf{U} reducida, pasando de 81 a 42 columnas. Por tanto, será esta matriz la utilizada en el entrenamiento y a la que se le calcula el número de condicionamiento.

Por último, si se recurre a la normalización de la matriz \mathbf{U} previamente al cómputo de su pseudoinversa, el número de condicionamiento deberá calcularse con la versión normalizada de la matriz.

Tabla 5.3 Valores de número de condicionamiento obtenidos en el experimento 6.

Método	Número de condicionamiento
Bucle cerrado original	$4,024 \cdot 10^{43}$
Base adaptable	$4,024 \cdot 10^{43}$
Matriz de covarianza	$3,880 \cdot 10^{43}$
DOMP	$2,080 \cdot 10^{37}$
Normalización	$9,540 \cdot 10^{16}$
Normalización y DOMP	$1,223 \cdot 10^9$

En la Tabla 5.3 se ha recogido el número de condicionamiento obtenido para cada uno de los casos tenidos en cuenta. En primer lugar, en caso de tener que invertir directamente la matriz U implica un número de condicionamiento de $4,024 \cdot 10^{43}$, lo cual es bastante elevado. Si se utiliza la matriz de covarianza, el número de condicionamiento es de $3,880 \cdot 10^{43}$, siendo muy similar al caso anterior, por lo que este método no mejora a los anteriores en términos de estabilidad. Si se aplica el algoritmo DOMP al método de bucle cerrado original, se consigue un número de condicionamiento de $2,080 \cdot 10^{37}$, lo que significa que la aplicación de dicho algoritmo mejora el condicionamiento en 6 órdenes de magnitud. El mejor resultado se consigue cuando se normaliza la matriz U , reduciendo el número de condicionamiento hasta $9,540 \cdot 10^{16}$, es decir, 27 órdenes de magnitud menor que sin normalizar.

Con este resultado, se puede concluir que, aunque la normalización no ofrezca grandes mejoras en las características de la señal, sí que permite trabajar con un número de condicionamiento mucho menor, por lo que sería muy recomendable utilizarla.

Por último, a pesar de que en este trabajo no se han podido realizar medidas en las que se aplique el algoritmo DOMP y, luego, se normalice la matriz correspondiente, ambos podrían ser compatibles, por lo que se ha realizado una última prueba para calcular el número de condicionamiento en este caso. El resultado ha sido recogido en la última fila de la Tabla 5.3, llegando a los $1,223 \cdot 10^9$ y consiguiendo así el menor de los valores computados.

La segunda parte de la prueba consiste en medir el tiempo empleado por cada método para ajustar los pesos del DPD. Para medir el tiempo que requiere cada método, se ha fijado un total de 5 iteraciones y un valor de $\mu = 1$. Aunque con estos valores se deberían obtener buenos resultados de predistorsión, no será objeto de este experimento medir una señal de validación, sino únicamente comparar los tiempos invertidos por cada método.

Los resultados de este experimento han sido recogidos en la Tabla 5.4, que cuenta con cuatro columnas. La primera indica los métodos que han sido cronometrados, los cuales coinciden con los de la primera parte de este experimento. La segunda indica el tiempo total invertido desde que empieza el entrenamiento hasta que se obtiene el valor definitivo de los pesos del DPD. La tercera recoge el tiempo invertido en enviar la medida y recoger los resultados, lo cual es útil para restarlo al tiempo total y obtener el tiempo invertido en aprendizaje, que es recogido en la última columna.

Tabla 5.4 Tiempo de entrenamiento medido en el experimento 6.

Método	Tiempo total (s)	Tiempos de medidas (s)	Tiempo de aprendizaje (s)
Bucle cerrado original	211,959	198,036	13,923
Base adaptable	219,413	205,731	13,682
Matriz de covarianza	191,223	184,487	6,736
DOMP	210,162	205,186	4,976
Normalización	227,924	212,644	15,280

De esta tabla se puede obtener que el tiempo empleado en el aprendizaje para el primer método es de casi 14 segundos. Muy cerca de este, se sitúa el método de base adaptable, cuya diferencia es de décimas de segundo, por lo que podría decirse que requieren del mismo tiempo, lo cual tendría sentido porque realmente, en ambos se realizan las mismas operaciones, con la diferencia de que el segundo computa la pseudoinversa al final.

Pasando al método que utiliza la matriz de covarianza, se comprueba que el tiempo empleado ha sido menor a los 7 segundos y, si se realiza un cálculo comparándolo con el primer método, se obtiene que ha sido un 51,6% más rápido que este. Esto quiere decir que utilizar este algoritmo requiere un menor coste computacional, necesitando la mitad de tiempo que el primer método.

En la siguiente fila se encuentran los resultados obtenidos para el primer método, añadiendo la selección de componentes mediante DOMP. Este caso ha sido el que menor tiempo de entrenamiento ha requerido. Esto se debe a que cuando se aplica el algoritmo DOMP la matriz U se ve reducida, lo que puede facilitar su procesamiento. Un inconveniente de aplicar este algoritmo es que se requiere un tiempo previo para ejecutarlo, el cual ha sido en este experimento de 10,24 segundos. Sin embargo, aplicar esta técnica supone el aumento de un 64,3% de la velocidad de entrenamiento respecto a no utilizarlo, por lo que en determinados

casos en los que se realicen varias iteraciones o se requiera un menor tiempo de entrenamiento, será muy recomendable utilizar el algoritmo DOMP.

Finalmente, se tiene el tiempo requerido para entrenar los pesos del DPD usando las matrices normalizadas. Este caso es el que más tiempo ha necesitado invertir, lo cual tiene sentido porque realiza las mismas operaciones que el método original, pero habría que añadir los procesos de normalización y desnormalización de la matriz U . Aunque pueda parecer que la normalización sea un inconveniente debido a que no mejora el tiempo de aprendizaje, se ha comprobado que el número de condicionamiento mejora si se normalizan las matrices, por lo que normalizar ofrecerá una mejor estabilidad a cambio de requerir un tiempo algo mayor para el entrenamiento. No obstante, la aplicación del método DOMP es compatible con la posterior normalización de la matriz, por lo que se podría utilizar este algoritmo para compensar, e incluso reducir, el tiempo de computación.

6 Conclusiones

We've always defined ourselves by the ability to overcome the impossible. And we count these moments. These moments when we dared to aim higher, to break barriers, to reach for the stars, to make the unknown known. We count these moments as our proudest achievements. But we lost all that. Or perhaps we've just forgotten that we're still pioneers, that we've barely begun and our greatest accomplishments cannot be behind us, that our destiny lies above us.

JOSEPH A. COOPER EN INTERSTELLAR

En este último capítulo de la presente memoria, se realizará una breve revisión general de los métodos utilizados, obteniendo algunas conclusiones finales de los resultados obtenidos y, finalmente, exponiendo algunas de las posibilidades de continuación de este trabajo.

6.1 Conclusión

A lo largo de este trabajo, se ha ido realizando un estudio de algunos de los métodos de predistorsión digital empleados en la actualidad. Para ello, se comenzó partiendo de la definición y caracterización de los amplificadores de potencia. Posteriormente, se comentó el modelado de su comportamiento mediante los modelos basados en series de Volterra, para dar paso a una explicación para conocer en qué consiste la predistorsión digital. Además, se particularizó dicha explicación realizando una distinción entre las arquitecturas de aprendizaje indirecto y directo para, a continuación, introducir algunos métodos de esta última con el objetivo de la realización de diferentes experimentos.

Concretamente, se han realizado seis experimentos distintos, cuyos resultados fueron expuestos y, de los cuales, es posible llegar a algunas conclusiones. En primer lugar, se debe comentar el desempeño del primer método ya que, al aplicar este tipo de predistorsión digital, se obtiene una mejora significativa de todos los parámetros medidos en los experimentos.

Con los resultados del segundo experimento, se obtuvieron resultados mucho mejores respecto a los del primer método. Sin embargo, todas las mejoras se producían en la fase de entrenamiento, cuando se utilizaba la misma señal de entrenamiento, mientras que al cambiar la señal por otra distinta, las medidas empeoraban hasta llegar a unos niveles similares a los del anterior experimento. Con estos resultados, se puede afirmar que se aprecia un posible caso de *overfitting*, es decir, que la solución se ajusta demasiado a la señal de entrenamiento, evitando una buena generalización con señales diferentes.

En el tercer experimento se empleaba una técnica distinta que tenía como objetivo la reducción del coste computacional haciendo uso de la matriz de covarianza. Gracias a los resultados obtenidos en este experimento y en el último, se puede afirmar que este método no solo consiguió unos resultados similares a los de los otros métodos, sino que lo hace reduciendo el tiempo requerido, lo cual se convierte en argumentos muy favorables de cara a la utilización de este algoritmo en casos de recursos de computación limitados.

Por otro lado, no solo se han probado diferentes métodos de aprendizaje directo, sino que también se han realizado experimentos para observar su comportamiento al aplicar otro tipo de técnicas compatibles. Específicamente, se analizaron la selección de componentes y la reducción del factor de cresta. En el primer

caso, se utilizó la técnica DOMP para reducir el número de componentes y se llegó a la conclusión de que la aplicación de esta técnica no provocaba un deterioro de las prestaciones obtenidas. Además de eso, se comprobó que aplicar el algoritmo DOMP es muy recomendable para reducir el tiempo y coste de computación. En el caso de aplicar CFR, se obtuvieron resultados similares a los recogidos al no aplicarlo. Esto puede ser positivo ya que, en determinadas ocasiones puede requerirse la reducción del nivel de PAPR además de la predistorsión, por lo que es necesario saber que las prestaciones no vayan a empeorar en caso de aplicar CFR.

Finalmente, se realizaron pruebas para analizar el desempeño del aprendizaje directo al trabajar con la matriz U normalizada y se ha podido llegar a la conclusión de que ello implica un ligero aumento del tiempo de aprendizaje y el coste computacional, puesto que será necesario realizar las operaciones de normalización y desnormalización oportunos. Sin embargo, trabajar de esta forma implica que el número de condicionamiento sea mucho menor y, por tanto, que el cálculo de las matrices inversas involucradas en los algoritmos sea más estable.

Así, se puede concluir con que la aplicación de diferentes técnicas y algoritmos pueden tener distintos comportamientos. Además, será importante seleccionar las técnicas que se utilizarán adecuadamente puesto que, mientras que algunas de ellas son compatibles, otras implican requisitos distintos.

6.2 Líneas futuras

Mediante los experimentos realizados a lo largo de este proyecto se han analizado y comparado distintas técnicas y formas de implementar distintas arquitecturas de aprendizaje directo. Sin embargo, el ámbito de la predistorsión digital se trata de un campo de investigación con un gran interés actualmente. Por tanto, al existir numerosas investigaciones en curso a lo largo de todo el mundo acerca de la predistorsión, es muy probable que se publiquen nuevos métodos, que sean capaces de mejorar las prestaciones de señal obtenidas o que requieran un menor coste computacional para su ejecución.

Teniendo esto en cuenta, las líneas futuras de este trabajo podrían enfocarse de diferentes formas. En primer lugar, una vía de investigación podría consistir en el diseño de nuevos métodos o algoritmos de aprendizaje directo, que consigan mejores resultados o que sean más ligeros en tiempo de ejecución. Por otro lado, a pesar de que se han realizado diferentes experimentos en este trabajo, existen otros algoritmos que se podrían añadir a la comparativa. Además, este proyecto se ha centrado en la comparación de técnicas de aprendizaje directo, aunque también existe la posibilidad de realizar un análisis del comportamiento ofrecido por el aprendizaje directo frente al del aprendizaje indirecto. De hecho, estas dos arquitecturas son compatibles, por lo que es posible aplicar aprendizaje indirecto y, a continuación, aprendizaje directo para obtener los pesos del predistorsionador, lo cual significaría otra interesante línea futura de trabajo.

Finalmente, es importante recordar que la predistorsión es una técnica necesaria para conseguir que los amplificadores de potencia que pueden situarse, por ejemplo, en estaciones base, operen a potencias elevadas, con una eficiencia deseable y un comportamiento lineal. Por este motivo, otra de las líneas futuras de este proyecto consistiría en la implementación de estas técnicas en una placa de evaluación con el objetivo de analizar las distintas formas de aprendizaje directo utilizando un *hardware* que pudiera ser instalado en una estación base.

Bibliografía

- [1] S. Lambert, W. Heddeghem, W. Vereecken, B. Lannoo, D. Colle, and M. Pickavet, “Worldwide electricity consumption of communication networks,” *Optics express*, vol. 20, pp. B513–24, 12 2012.
- [2] P. K. Singya, N. Kumar, and V. Bhatia, “Mitigating NLD for wireless networks: Effect of nonlinear power amplifiers on future wireless communication networks,” *IEEE Microwave Magazine*, vol. 18, no. 5, pp. 73–90, 2017.
- [3] A. Oppenheim, *Señales y sistemas*. México: Prentice Hall, 1998.
- [4] J. Pedro, *Intermodulation distortion in microwave and wireless circuits*. Boston: Artech House, 2003.
- [5] K. Gharaibeh, *Nonlinear distortion in wireless systems : modelling and simulation with MATLAB*. Piscataway, New JerseyChichester, West Sussex, United Kingdom: IEEE Press John Wiley & Sons Ltd, 2012.
- [6] J. C. Wood, *Behavioral Modeling and Linearization of RF Power Amplifiers*. Artech House Publishers, 2014.
- [7] R. Raich, H. Qian, and G. Zhou, “Orthogonal polynomials for power amplifier modeling and predistorter design,” *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1468–1479, sep 2004.
- [8] A. Pérez Hernández, “Comparación de técnicas de selección de coeficientes en modelos de comportamiento basados en series de Volterra para sistemas de comunicaciones,” Trabajo Fin de Grado, Universidad de Sevilla, 2018.
- [9] A. Pérez-Hernández, J. Becerra, M. Madero-Ayora, and C. Crespo-Cadenas, “A Comprehensive Performance Benchmark of Component Selection Techniques for Volterra Behavioral Models,” *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, 2018.
- [10] D. Morgan, Z. Ma, J. Kim, M. Zierdt, and J. Pastalan, “A generalized memory polynomial model for digital predistortion of RF power amplifiers,” *IEEE Transactions on Signal Processing*, vol. 54, no. 10, pp. 3852–3860, oct 2006.
- [11] J. A. Becerra, M. J. Madero-Ayora, J. Reina-Tosina, C. Crespo-Cadenas, J. Garcia-Frias, and G. Arce, “A doubly orthogonal matching pursuit algorithm for sparse predistortion of power amplifiers,” *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 8, pp. 726–728, aug 2018.
- [12] S. Cripps, *Advanced techniques in RF power amplifier design*. Boston: Artech House, 2002.
- [13] D. López-Morilla, “Técnicas de reducción del factor de cresta aplicadas a señales 5G New Radio,” Trabajo Fin de Máster, Universidad de Sevilla, 2020.
- [14] F. Ghannouchi, *Behavioral modelling and predistortion of wideband wireless transmitters*. Hoboken: John Wiley & Sons Inc, 2015.
- [15] H. Paaso and A. Mämmelä, “Comparison of direct learning and indirect learning predistortion architectures,” in *2008 IEEE International Symposium on Wireless Communication Systems*, 2008, pp. 309–313.

- [16] P. N. Landin, S. Gustafsson, C. Fager, and T. Eriksson, "Weblab: A web-based setup for pa digital predistortion and characterization [application notes]," *IEEE Microwave Magazine*, vol. 16, no. 1, pp. 138–140, 2015.
- [17] "RF WebLab measurement setup," <http://dpdcompetition.com/rfweblab/measurement-setup/>, 2015.
- [18] D. Lopez-Bueno, Q. A. Pham, G. Montoro, and P. L. Gilabert, "Independent digital predistortion parameters estimation using adaptive principal component analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5771–5779, dec 2018.
- [19] C. Yu, K. Tang, and Y. Liu, "Adaptive Basis Direct Learning Method for Predistortion of RF Power Amplifier," *IEEE Microwave and Wireless Components Letters*, vol. 30, no. 1, pp. 98–101, 2020.
- [20] Z. Wang, W. Chen, G. Su, F. M. Ghannouchi, Z. Feng, and Y. Liu, "Low computational complexity digital predistortion based on direct learning with covariance matrix," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 11, pp. 4274–4284, 2017.
- [21] G. Drašković and F. Pascal, "New properties for Tyler's covariance matrix estimator," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, 2016, pp. 820–824.
- [22] I. Soloveychik and A. Wiesel, "Tyler's covariance matrix estimator in elliptical models with convex structure," *IEEE Transactions on Signal Processing*, vol. 62, no. 20, pp. 5251–5259, 2014.
- [23] S. C. Cripps, *RF Power Amplifiers for Wireless Communications*. Artech House Publishers, 2006.
- [24] K. Rawat, P. Roblin, and S. K. Koul, *Bandwidth and Efficiency Enhancement in Radio Frequency Power Amplifiers for Wireless Transmitters*. Springer International Publishing, 2020.
- [25] S. P. Yadav and S. C. Bera, "Nonlinearity effect of high power amplifiers in communication systems," in *2014 International Conference on Advances in Communication and Computing Technologies (ICACACT 2014)*, jun 2014.
- [26] T. Jiang and Y. Wu, "An overview: Peak-to-average power ratio reduction techniques for OFDM signals," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 257–268, jun 2008.
- [27] S. Maas, *Nonlinear microwave and RF circuits*. Boston, MA: Artech House, 2003.

Índice de Figuras

2.1	Punto de compresión de 1 dB	5
2.2	Definición de ACPR	6
2.3	Definición de EVM	7
2.4	Ejemplo de característica AM-AM	8
2.5	Ejemplo de característica AM-PM	9
4.1	Característica entrada-salida de un amplificador y de un predistorsionador	20
4.2	Diagrama de bloques genérico de un sistema de predistorsión digital	21
4.3	Controlador de autoajuste	23
4.4	Arquitectura de aprendizaje indirecto	23
4.5	Arquitectura de aprendizaje directo	24
5.1	Configuración del laboratorio de Chalmers [17]	28
5.2	Espectro de la señal en banda base	29
5.3	Diagrama de bloques de la DLA utilizada	30
5.4	Espectro de la señal de salida sin DPD	33
5.5	Evolución del ACPR del canal inferior en el experimento 1	34
5.6	Evolución del ACPR del canal superior en el experimento 1	34
5.7	Evolución del ACPR _{2a} en el experimento 1	35
5.8	Evolución del ACPR _{2b} en el experimento 1	35
5.9	Evolución del EVM en el experimento 1	36
5.10	Evolución del NMSE en el experimento 1	36
5.11	Evolución del ACPR del canal inferior en el experimento 2	38
5.12	Evolución del ACPR del canal superior en el experimento 2	39
5.13	Evolución del ACPR _{2a} en el experimento 2	40
5.14	Evolución del ACPR _{2b} en el experimento 2	40
5.15	Evolución del EVM en el experimento 2	41
5.16	Evolución del NMSE en el experimento 2	41
5.17	Evolución del ACPR del canal inferior en el experimento 4	46
5.18	Evolución del ACPR del canal superior en el experimento 4	46
5.19	Evolución del ACPR _{2a} en el experimento 4	47
5.20	Evolución del ACPR _{2b} en el experimento 4	47
5.21	Evolución del EVM en el experimento 4	48
5.22	Evolución del NMSE en el experimento 4	48
5.23	Evolución del ACPR del canal inferior en el experimento 5	51
5.24	Evolución del ACPR del canal superior en el experimento 5	51
5.25	Evolución del ACPR _{2a} en el experimento 5	52
5.26	Evolución del ACPR _{2b} en el experimento 5	52
5.27	Evolución del EVM en el experimento 5	53
5.28	Evolución del NMSE en el experimento 5	53

Índice de Tablas

5.1	Parámetros de configuración del modelo GMP utilizado	29
5.2	Resultados del experimento 3	44
5.3	Valores de número de condicionamiento obtenidos en el experimento 6	54
5.4	Tiempo de entrenamiento medido en el experimento 6	55

Glosario

3GPP *3rd Generation Partnership Project*. 19

ACI *Adjacent-Channel Interference*. 6

ACPR *Adjacent-Channel Power Ratio*. 6, 21, 29, 33, 34, 38, 43, 45, 49

AM-AM *Amplitude Modulation - Amplitude Modulation*. VII, 8, 10, 11, 21

AM-PM *Amplitude Modulation - Phase Modulation*. VII, 8, 10, 11, 20, 21

CFR *Crest Factor Reduction*. V, VIII, 21, 44, 47–49, 51, 58

dB *Decibelios*. VII, 5, 8, 61

DC *Direct Current*. 4, 9

DLA *Direct Learning Architecture*. 22–24, 30, 37, 46, 61

DOMP *Doubly Orthogonal Matching Pursuit*. V, VII, 16, 17, 44–46, 53–55, 58

DPD *Digital Predistortion*. I, III, VII, 1, 19–22, 24, 25, 30–34, 37, 41, 43, 46–48, 53–55, 61

DSP *Digital Signal Processor*. 20, 21

DUT *Device Under Test*. 21, 22

EVM *Error Vector Magnitude*. VII, 7, 21, 29, 33, 36, 40, 43, 47, 49, 52, 61

FFT *Fast Fourier Transform*. 28

FV *Full-Volterra*. 14, 15

GMP *Generalized Memory Polynomial*. VII, 15, 28, 29, 63

ILA *Indirect Learning Architecture*. 22, 23, 37

IMD *Intermodulation Distortion*. 6

IMR *Intermodulation Ratio*. 6

IMS *International Microwave Symposium*. 27

ITU-T *International Telecommunications Union Standardization Sector*. 19

LS *Least Squares*. 15–17, 23, 31

LTE *Long-Term Evolution*. 28

MIT *Massachusetts Institute of Technology*. 12

MP *Memory Polynomial*. VII, 14, 15

NMSE *Normalized Mean Squared Error*. VII, 7, 30, 33, 36, 41, 43, 47, 49, 52, 61

OFDM *Orthogonal Frequency Division Multiplexing*. 6, 19, 28

OMP *Orthogonal Matching Pursuit*. 16, 37, 44

PA *Power Amplifier*. I, III, 1, 3–6, 8, 10, 12, 20–25, 27, 29, 30, 47

PAPR *Peak to Average Power Ratio*. VII, 5, 6, 19–21, 42, 44, 47, 48, 58

PCA *Principal Component Analysis*. 16

PSD *Power Spectral Density*. 7

RF *Radiofrecuencia*. 3, 9, 11, 12, 19

RMS *Root Mean Square*. 28, 30

RSS *Residual Sum of Squared error*. 16

VSA *Vector Signal Analyzer*. 27

VSG *Vector Signal Generator*. 27

VST *Vector Signal Transceiver*. 27