

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Comparación de técnicas de selección de coeficientes en modelos de comportamiento basados en series de Volterra para sistemas de comunicaciones

Autor: Abraham Pérez Hernández

Tutores: María José Madero Ayora y Juan Antonio Becerra González

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

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Comparación de técnicas de selección de coeficientes en modelos de comportamiento basados en series de Volterra para sistemas de comunicaciones

Autor:

Abraham Pérez Hernández

Tutores:

María José Madero Ayora

Profesora Titular

Juan Antonio Becerra González

Profesor Sustituto Interino

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

Sevilla, 2018

Trabajo Fin de Grado: Comparación de técnicas de selección de coeficientes en modelos de comportamiento basados en series de Volterra para sistemas de comunicaciones

Autor: Abraham Pérez Hernández

Tutores: María José Madero Ayora y Juan Antonio Becerra González

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 dar las gracias a todos los educadores que durante esta carrera de fondo han sabido ofrecernos su sabiduría para que crezcamos y podamos desenvolvernos en un ambiente de trabajo futuro. En especial, a Carlos Crespo y a Javier Reina por la tutoría y la ayuda que me ofrecieron cuando decidí pedir la beca de pregraduados del MTT-S, punto de inflexión de este viaje.

A la tutora de este trabajo, María José Madero, porque además de ser una excelente profesora, ha sabido guiarme por el mundo de la predistorsión digital y de la investigación, y ha estado siempre disponible para una tutoría motivadora. Y sobre todo, a Juan Antonio Becerra, cotutor del proyecto: por los juegos de Twitter, por la práctica de Comunicaciones Digitales, por las prácticas en Sigrow, por la feria... En definitiva, por confiar en mí y por ese "eagerness and bright" que has traído a mi vida. Thank you, Master.

También querría agradecer a todos los compañeros que han hecho que este grado sea más fácil. A Guillermo, Javi, Dani, Diego, Ilde, a todos, ha sido un placer y es un honor poder consideraros hoy mis amigos. No quiero olvidar a los apoyos palermos que he tenido, que saben lo especial que son para mí: Alejandro, Castilla, Lucía, Kake, Barrous, Sonia, y todos los demás, gracias.

A ella, mi mejor amiga y mi compañera, por aguantarme, por hacerme reír, por reírte,...; en definitiva, por acompañarme y dejarte acompañar durante estos cuatro años de universidad y estos cinco años de vida. A ti, Isabel Ortega. Te quiero.

Empecé estos agradecimientos dando las "gracias a todos los educadores que he tenido desde que nací", pero no he nombrado a conciencia a los más importantes: a mis padres, Vicente y Pilar, por apostar por la fe en aquel año de 1996. A todos mis hermanos: Pilar, Vicente, Ana, Noemí, Gabriel y Ester, porque hacen que la vida no sea tan aburrida. Gracias a toda esta gran familia numerosa por todos los enfados, todas las reconciliaciones, por todas las enseñanzas. Me gustaría tomarle prestado a una de estas personas un texto que escribió hace pocos meses: *"Pues sí, somos siete. Somos siete hermanos, cada cual más distinto, algunos con alguna que otra faceta (y cara) muy similares. Médicos, Arquitecto, Ingeniero y alguno que otro en proceso de formación. Siete, cada uno por su lado, pero todos a una a la hora de la verdad. Siete "bichos raros" que se han educado en la constancia y el amor. Siete, que cada día agradecen más ser parte de ese número. Siete, que agradecen a dos el que un día decidieran ser uno, y darnos la vida"*. Gracias OHANA.

Abraham Pérez Hernández
Sevilla, 2018

Resumen

Este trabajo presenta una comparativa de técnicas de selección de componentes para modelos basados en series de Volterra destinado a la predistorsión digital de amplificadores de potencia. En primer lugar se ha introducido teóricamente la distorsión no lineal presente en los dispositivos activos, concretando en el amplificador de potencia, y la serie de Volterra, herramienta matemática en la que se basan los modelos de comportamiento de amplificadores de potencia contemplados en este documento. Seguidamente, las técnicas *Orthogonal Matching Pursuit* (OMP), *Iterated Ridge Regression* (IRR) y Análisis de Componentes Principales (PCA) han sido analizados y comparados desde un punto de vista teórico. La parte experimental consiste en el modelado de la salida esperada de un amplificador de potencia GaN bajo prueba cuando se le excita con una señal de entrada según el estándar *long term evolution* (LTE) a 15 MHz. Las señales medidas han sido adquiridas usando un analizador vectorial de señal y postprocesadas para obtener alineamiento temporal entre la señal medida y la de referencia. Los resultados experimentales enriquecen esta comparativa y resaltan los beneficios de cada uno de los algoritmos.

Abstract

This work presents a comparative of component selection techniques for Volterra series models to digital predistortion of RF power amplifiers. At first, the theory of non-linear distortion, present in active devices as the power amplifier, has been introduced, as well as the Volterra series, mathematical tool on which the behavioral models treated in this document are based. Next, the Orthogonal Matching Pursuit (OMP), Iterated Ridge Regression (IRR) and Principal Component Analysis (PCA) techniques are compared from a theoretical point of view. The experimental part consists on the modeling of the expected output of the GaN power amplifier under test driven by a 15-MHz Long Term Evolution (LTE) input signal. The measured signals are acquired by using a vector signal analyzer and post-processed to get time alignment between the measured and the reference signal. Experimental results enrich the desired benchmark and highlight the benefits of each one of the algorithms.

Índice Abreviado

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1 Introducción	1
2 Distorsión no lineal, efectos de memoria y cómo detectarlos	3
2.1 Distorsión no lineal	3
2.2 Técnicas de caracterización experimental de efectos no lineales	6
2.3 Efectos de memoria	9
3 Series de Volterra y Amplificadores de Potencia	11
3.1 Introducción y desarrollo de las series de Volterra	12
3.2 Modelos de comportamiento de amplificadores de potencia basados en series de Volterra	14
4 Técnicas de selección de componentes	17
4.1 Obtención de coeficientes de los modelos de comportamiento	17
4.2 Estrategias de reducción de componentes activos	18
5 Resultados	23
6 Conclusiones y líneas futuras de trabajo	31
Apéndice A Códigos de MATLAB®	33
A.1 Función generatriz de parámetros del modelo FV	33
A.2 Función de identificación y validación del modelo FV	35
A.3 Algoritmos y funciones auxiliares de las técnicas OMP, IRR y PCA.	37
<i>Índice de Figuras</i>	43
<i>Índice de Tablas</i>	45
<i>Índice de Códigos</i>	47
<i>Bibliografía</i>	49

Índice

<i>Resumen</i>	III
<i>Abstract</i>	V
<i>Índice Abreviado</i>	VII
1 Introducción	1
2 Distorsión no lineal, efectos de memoria y cómo detectarlos	3
2.1 Distorsión no lineal	3
2.1.1 Linealidad y no linealidad	3
2.1.2 Efectos de las no linealidades	4
2.2 Técnicas de caracterización experimental de efectos no lineales	6
2.2.1 Test de caracterización de un tono	6
2.2.2 Test de caracterización de dos tonos	7
2.2.3 Test de caracterización de un espectro continuo limitado en banda	8
2.3 Efectos de memoria	9
3 Series de Volterra y Amplificadores de Potencia	11
3.1 Introducción y desarrollo de las series de Volterra	12
3.1.1 Serie de Volterra como desarrollo de una serie de Taylor	12
3.1.2 Serie de Volterra como desarrollo de un sistema lineal	13
3.2 Modelos de comportamiento de amplificadores de potencia basados en series de Volterra	14
3.2.1 Modelo completo de Volterra	14
3.2.2 Modelo Polinomial con Memoria Generalizado	14
3.2.3 Modelo de series de Volterra de valores complejos	15
4 Técnicas de selección de componentes	17
4.1 Obtención de coeficientes de los modelos de comportamiento	17
4.2 Estrategias de reducción de componentes activos	18
4.2.1 <i>Orthogonal Matching Pursuit</i>	19
4.2.2 <i>Iterated Ridge Regression</i>	19
4.2.3 Análisis de Componentes Principales	20
5 Resultados	23
6 Conclusiones y líneas futuras de trabajo	31
Apéndice A Códigos de MATLAB®	33
A.1 Función generatriz de parámetros del modelo FV	33
A.2 Función de identificación y validación del modelo FV	35
A.3 Algoritmos y funciones auxiliares de las técnicas OMP, IRR y PCA.	37

<i>Índice de Figuras</i>	43
<i>Índice de Tablas</i>	45
<i>Índice de Códigos</i>	47
<i>Bibliografía</i>	49

1 Introducción

Sometimes it is the people no one imagines anything of who do the things no one can imagine.

ALAN TURING EN *The Imitation Game*, 2014

La tecnología ha cambiado a la industria de las telecomunicaciones. Todas las comunicaciones en la actualidad usan señales moduladas digitalmente y con un ancho de banda eficiente, transmitiendo además una alta cantidad de información a velocidades muy altas. Un ejemplo son las comunicaciones móviles. Su objetivo es llegar al máximo número de clientes posibles en el área más extensa posible y lo más barato posible.

Sin embargo, realizar esto implica un gasto de energía estratosférico. En el mundo actual, reducir el consumo de energía eléctrica es uno de los focos de atención principales, no solo por el ahorro económico sino también por la reducción de la contaminación. Se estima que las telecomunicaciones representan sobre un 10% del consumo energético total [1]. Además, el crecimiento anual de la energía consumida por las redes de comunicación es dos veces más rápido que el crecimiento de otros sectores dentro de las telecomunicaciones. Esto se debe a que los operadores continuamente instalan nuevas estaciones para perseguir su objetivo: más clientes y más área cubierta. Y estas estaciones usan una cantidad de energía enorme para transmitir la información. En particular, el dispositivo que consume la mayor parte de esta energía es el amplificador de potencia. Los amplificadores de potencia tienen una característica muy especial: su eficiencia es mayor cuando se trabaja a potencias altas, pero a estas potencias aparece el fenómeno de la distorsión no lineal. Esta distorsión provoca que la calidad de la señal sea menor y se produzcan más errores en la recepción de las señales. Por tanto, si se busca la eficiencia energética, habría que aumentar la potencia. Además, las señales actuales poseen una relación de potencia pico a potencia media muy alta, por lo que para estas señales la linealidad de los amplificadores es muy importante.

Por estos motivos, el interés en la linealización de los amplificadores ha crecido últimamente, pero es importante destacar que no es nuevo. En la década de 1920 durante la expansión de las telecomunicaciones ya se concibieron los tres tipos de linealización que se conciben hoy en día: la retroalimentación (*feedback*), la pre-alimentación (*feedforward*) y la predistorsión. Según la literatura [2], la primera forma de linealización en radiofrecuencia (RF) fue la pre-alimentación, inventada por Howard Black in 1923 en Bell Telephone Laboratories. Se basaba en dos bucles donde se usa un amplificador auxiliar además del primero, y este debía operar en una zona relativamente lineal para que no introdujera su propia distorsión, por lo que la cancelación de la distorsión se producía unos 6 dB por debajo de saturación, lo que es ineficiente. En consecuencia, la linealización por pre-alimentación no es viable para operar cerca de saturación.

De esta forma, los ingenieros de las estaciones base comenzaron a buscar otras formas de linealización. Fue en 1927 cuando de nuevo Howard Black propuso la linealización por retroalimentación negativa: se dió cuenta que realimentando la salida a la entrada con un desfase de 180°, la distorsión podría ser altamente reducida a expensas de la ganancia. Este concepto supuso la fundación del amplificador industrial por retroalimentación, que conseguía unos muy buenos resultados de rechazo de distorsión y fue usado ampliamente en los amplificadores de potencia de RF. Aún así, seguía habiendo limitaciones cuando se trabajaba con altas frecuencias y anchos de banda extensos.

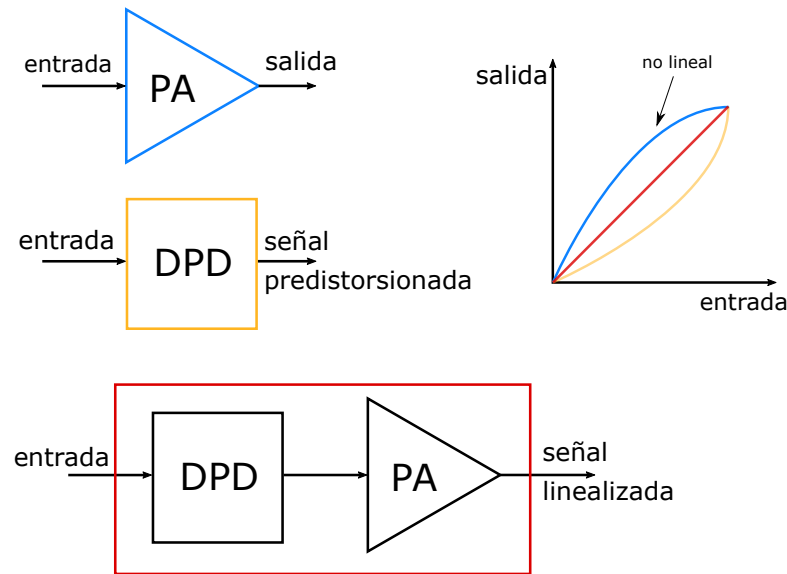


Figura 1.1 Esquema gráfico del concepto de la predistorsión para linealizar amplificadores de potencia.

Más adelante, en torno a la década de 1970 se empezaron a desarrollar los predistorsionadores. La idea es muy sencilla: diseñar un dispositivo no lineal cuya función de transferencia sea la inversa de la del amplificador de potencia. Así, como se ve en la Figura 1.1, la cascada de ambos sería un amplificador lineal. Aún así, esto no es totalmente posible, pero se puede llegar a resultados muy satisfactorios. Al principio se trabajaba con predistorsionadores analógicos, pero a lo largo de los años 90, las comunicaciones móviles pasaron de ser de modulación analógica a digital, por lo que la predistorsión intentó seguir este movimiento y lograr obtener amplificadores de potencia altamente eficientes para reemplazar aquellos que usaban pre-alimentación y retroalimentación. Esto marcó el inicio de la predistorsión digital (DPD, del inglés *Digital PreDistortion*).

Para desarrollar predistorsionadores digitales por tanto, es clave poder modelar en primer lugar el comportamiento de los amplificadores de potencia: la ganancia que ofrece y la distorsión no lineal que provoca. Así, se empezaron a realizar experimentos con los amplificadores y se desarrolló la predistorsión LUT (del inglés *LookUp Table*), donde la respuesta de amplitud y fase del amplificador se grababa en función de la potencia de entrada y la respuestas se invertían para otorgar linealización. Sin embargo, se descubrió que los efectos de memoria eran bastante notorios en los amplificadores, por lo que había que modelarlos también. La implementación LUT no tenía en cuenta la memoria, se trataba de un modelo ML (del inglés *MemoryLess*), así que había que encontrar otra forma. La respuesta fueron las series de Volterra.

Con las series de Volterra se puede modelar cualquier comportamiento no lineal y con todos los efectos de memoria que se requiera, simplemente designando la importancia de cada componente mediante coeficientes o "kernels". El problema que presentan los modelos basados en series de Volterra es que constan de multitud de coeficientes, por lo que para diseñar el predistorsionador digital se toman dos aproximaciones distintas: elegir los coeficientes del modelo que van a estar activos *a priori*, o desarrollar algoritmos de selección de componentes.

En el Capítulo 2 de este trabajo se explicará porqué los amplificadores de potencia son intrínsecamente no lineales, qué efectos tiene esta no linealidad y cómo se pueden detectar y medir, resaltando las figuras de mérito dentro de este ámbito. En el Capítulo 3 se introducirá matemáticamente la serie de Volterra y los modelos de comportamiento de amplificadores de potencia basados en ella, comentando los modelos más usuales. Posteriormente, en el Capítulo 4 se pone el foco en la obtención de coeficientes de los modelos y principalmente en las estrategias de reducción de estos, destacando tres algoritmos: *Orthogonal Matching Pursuit* (OMP), *Iterated Ridge Regression* (IRR) y el Análisis de Componentes Principales (PCA, del inglés *Principal Components Analysis*). A continuación, en el Capítulo 5 se explican los experimentos y las simulaciones realizadas en este proyecto, y para finalizar, en el Capítulo 6 se exponen unas conclusiones finales y se proponen líneas de trabajo futuras.

2 Distorsión no lineal, efectos de memoria y cómo detectarlos

Dudar de todo o creerlo todo son dos opciones igualmente cómodas, pues tanto una como otra nos eximen de reflexionar.

JULES HENRI POINCARÉ

El propósito de este capítulo es introducir teóricamente el concepto de *distorsión no lineal*, explicar las técnicas experimentales para medir los efectos de las no linealidades y también comentar las posibles consecuencias que la presencia de memoria en los dispositivos de nuestro sistema puede tener en nuestra señal. De esta forma se podrá entender la importancia que tiene el foco final de esta línea de trabajo: la predistorsión digital.

2.1 Distorsión no lineal

Según [3], la distorsión no lineal es "*una forma de perturbación de la señal originada por las no linealidades del sistema*" (p. 16). Por tanto, es interesante hablar primero de las posibles perturbaciones que puede tener nuestra señal. Se pueden agrupar en ruido, interferencia y distorsión. El ruido se define como una señal perturbadora en un circuito producida por sus componentes. A su vez, se entiende por interferencia a aquella señal procedente de otro sistema o comunicación que se introduce, de forma intencionada o no, en nuestro sistema o comunicación. En conclusión, tanto el ruido como la interferencia son perturbaciones aditivas a nuestra señal. Sin embargo, la distorsión tiene otro enfoque totalmente distinto. Es una modificación de la señal, por lo que no puede separarse de esta. En otras palabras, la distorsión no puede captarse si la señal perturbada no está siendo transmitida, al contrario que el ruido y la interferencia. Por otro lado, también se debe hablar de la linealidad y no linealidad de un sistema. Para ello, se definirán ambos conceptos y se caracterizarán, llegando así a un mayor entendimiento de la distorsión no lineal.

2.1.1 Linealidad y no linealidad

Un sistema lineal es aquel en el que se cumple el principio de superposición, el cual establece que la respuesta de un sistema a una suma de señales de entrada es la suma de las respuestas del sistema a cada señal de entrada por separado. Matemáticamente se puede expresar como: siendo $y_i(t)$ la salida del sistema cuando la señal $x_i(t)$ está a la entrada en el instante t , siendo i un número entero, un sistema es lineal si la salida obtenida es igual a

$$y(t) = \alpha_1 y_1(t) + \alpha_2 y_2(t) \quad (2.1)$$

cuando a la entrada se ha introducido la señal

$$x(t) = \alpha_1 x_1(t) + \alpha_2 x_2(t), \quad (2.2)$$

donde α_i son los coeficientes de cada señal $x_i(t)$. En este principio radica la importancia de la linealidad en el desarrollo de la teoría de sistemas, ya que en un sistema lineal, una entrada arbitraria muy compleja se puede

descomponer en señales básicas más simples y estudiar la respuesta de cada una de estas por separado: facilita en gran medida el estudio de los sistemas. También es importante mencionar que el término *lineal* proviene del hecho de que la ganancia en potencia de un sistema lineal es constante, por lo que si se representa la potencia de la señal de salida de un sistema lineal frente a la potencia de la señal de entrada se debe observar una recta, una *línea*.

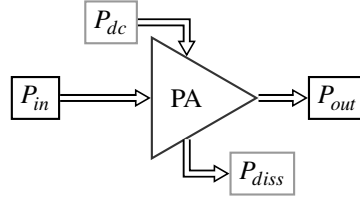


Figura 2.1 Balance de potencias de un amplificador de potencia, sirviendo como ejemplo de cualquier dispositivo activo.

Una vez definida la linealidad de un sistema, todo sistema que no cumpla dichos principios se trata de un sistema no lineal. Aunque parezca que es la excepción, lo cierto es que los sistemas no lineales son mucho más comunes que los sistemas lineales. Sin embargo, a menudo los sistemas no lineales se comportan como sistemas lineales cuando se trabaja en cierto rango de potencia o en variaciones de potencia muy pequeña, como es el caso del modelo electrónico de pequeña señal de los transistores, o los amplificadores de potencia cuando se trabaja en potencias bajas. Para ilustrar este comportamiento, se va a hallar la ganancia de un amplificador real, siguiendo el esquema de la Figura 2.1, en el cual P_{in} es la potencia de la señal de entrada, P_{out} la potencia de la señal de salida, y P_{diss} y P_{dc} representan las potencias disipada y de alimentación, respectivamente. De esta forma, la ganancia del sistema se define como

$$G = \frac{P_{out}}{P_{in}}, \quad (2.3)$$

y aplicando la ley de la conservación de energía que impone que $P_{in} + P_{dc} = P_{out} + P_{diss}$, se deriva que

$$G = \frac{P_{out}}{P_{in}} = \frac{P_{in} + P_{dc} - P_{diss}}{P_{in}} = 1 + \frac{P_{dc} - P_{diss}}{P_{in}}. \quad (2.4)$$

Así pues, suponiendo que P_{dc} está limitada a un valor finito por parte de la alimentación y que P_{diss} tiene un valor teórico mínimo nulo, observando la Ec. (2.4) se concluye que es imposible para el amplificador presentar una ganancia constante cuando P_{in} va aumentando. Este resultado se ilustra en la Figura 2.2, donde se observa cómo la ganancia no es una recta constante, sino que decrece cuando la potencia de entrada aumenta. Esta deducción se aplica a cualquier dispositivo que esté alimentado, por lo que todos los dispositivos activos son inherentemente no lineales.

2.1.2 Efectos de las no linealidades

Al contrario de lo que pueda parecer, los sistemas no lineales son básicos y necesarios en los sistemas de telecomunicación. Muchas veces los efectos que estos tienen son beneficiosos para el desarrollo de la comunicación, como es el caso de los mezcladores; pero en otros casos, empeoran y *ensucian* el espectro actuando de forma negativa en la canalización de los sistemas. Para poder comprender de dónde provienen dichos efectos, se van a desarrollar matemáticamente a continuación las expresiones que originan estos resultados.

En el documento se tomará la notación más usada en las referencias de análisis de sistemas de comunicaciones, en las cuales la señal de entrada del sistema será $x(t)$ y la señal de salida $y(t)$. Los sistemas son operadores $S[\cdot]$ de la entrada, siendo el ejemplo típico de sistema lineal $y(t) = S_L[x(t)] = \alpha x(t - \tau)$, en el cual α es el coeficiente por el que se multiplica la señal, pudiendo resultar en una ganancia o en una atenuación, y τ es el retraso de la transmisión de la señal por el sistema. En primer lugar, hay que establecer que la señal típica en los sistemas de comunicaciones es una señal sinusoidal que se puede generalizar como

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

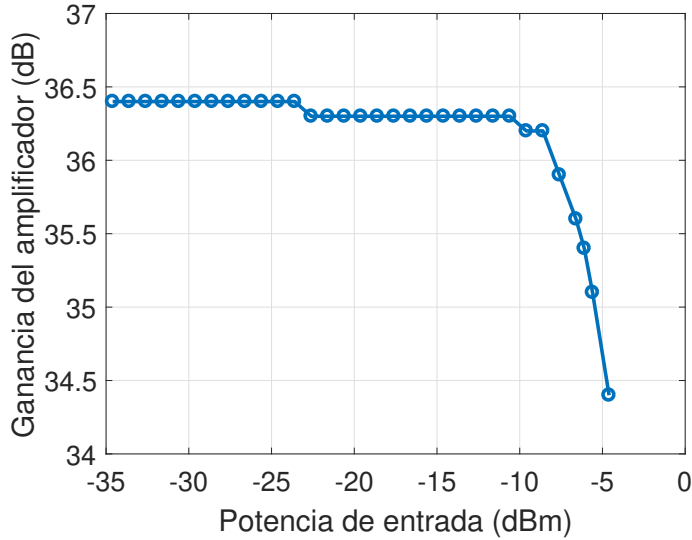


Figura 2.2 Ganancia estática real presentada por el amplificador ZHL42W de Minicircuits frente a una entrada de un tono a 850 MHz, obtenida por medidas experimentales en el laboratorio.

donde la variación de la amplitud $A(t)$ y/o la variación de la fase $\theta(t)$ portan la información de la señal. De esta forma, la señal de salida típica de un sistema lineal, habiéndose sustituido $\phi = f_c \tau$, sigue la expresión

$$y_L(t) = S_L[x(t)] = \alpha A(t - \tau) \cos[2\pi f_c t + \theta(t - \tau) - \phi]. \quad (2.6)$$

Tal y como se ha narrado en la sección anterior, los sistemas no lineales son mucho más genéricos que los lineales, ya que todo sistema que no cumpla el principio de superposición, en la Ec. (2.1), es no lineal. Sin embargo, para realizar el desarrollo deseado, se van a simplificar los sistemas no lineales como un polinomio, truncado a tercer grado, siguiendo la expresión

$$y_{NL}(t) = S_{NL}[x(t)] = \alpha_1 x(t - \tau_1) + \alpha_2 x^2(t - \tau_2) + \alpha_3 x^3(t - \tau_3). \quad (2.7)$$

De esta forma, si la señal $x(t)$ es la expresada en la Ec. (2.5), la señal que se obtiene por este sistema es

$$\begin{aligned} y_{NL}(t) = & \alpha_1 A(t - \tau_1) \cos[2\pi f_c t + \theta(t - \tau_1) - \phi_1] \\ & + \alpha_2 A(t - \tau_2) \cos[2\pi f_c t + \theta(t - \tau_2) - \phi_2]^2 \\ & + \alpha_3 A(t - \tau_3) \cos[2\pi f_c t + \theta(t - \tau_3) - \phi_3]^3, \end{aligned} \quad (2.8)$$

habiéndose sustituido $2\pi f_c \tau_i$ por ϕ_i para $i = 1, 2, 3$. Esta expresión se puede desarrollar matemáticamente, usando la propiedad trigonométrica del producto de cosenos, de forma que se pueda analizar la frecuencia de cada sinusoides de forma más clara. Además, se puede suponer que la variación de la portadora de radiofrecuencia es mucho más rápida que la variación de la amplitud $A(t)$ y la fase $\theta(t)$, por lo que el retraso del sistema en estas señales se puede despreciar, llevando en el desfase ϕ los efectos de este retraso. En consecuencia, si se aplica y desarrolla lo anteriormente dicho, es sencillo obtener como expresión final la siguiente:

$$\begin{aligned} y(t) = & \frac{1}{2} \alpha_2 A(t)^2 \\ & + \alpha_1 A(t) \cos[2\pi f_c t + \theta(t) - \phi_1] + \frac{3}{4} \alpha_3 A(t)^3 \cos[2\pi f_c t + \theta(t) - \phi_3] \\ & + \frac{1}{2} \alpha_2 A(t)^2 \cos[2\pi(2f_c)t + 2\theta(t) - 2\phi_2] \\ & + \frac{1}{4} \alpha_3 A(t)^3 \cos[2\pi(3f_c)t + 3\theta(t) - 3\phi_3] \end{aligned} \quad (2.9)$$

La primera conclusión que se puede obtener a simple vista comparando la expresión de la respuesta del sistema lineal en la Ec. (2.6) con la Ec. (2.9) es que la segunda tiene muchos más términos, lo que implica

que ésta es mucho más compleja que la primera. Observando la frecuencia de cada término, se puede ver cómo, al contrario que la respuesta lineal, donde la frecuencia de la entrada y la salida es la misma, en la respuesta no lineal aparecen nuevos términos con frecuencias distintas a la entrada: es lo que se conoce como *armónicos*. Es uno de los efectos más notables de los sistemas no lineales. Los lineales solo pueden afectar de forma cuantitativa al espectro de la señal de entrada, sin embargo los no lineales modifican la forma del espectro en sí, lo que los hace, además de más complejos, más atractivos desde el punto de vista de las telecomunicaciones. Ahora se puede entender cómo un filtro puede eliminar ciertas componentes espectrales y enfatizar otras, o cómo un mezclador puede elevar en frecuencia una señal de entrada en función de un oscilador local. También cabe destacar que las componentes generadas, tanto la que se encuentra en banda base, en torno a $f = 0$, como las componentes armónicas de segundo y tercer orden, que es el nombre que se les da a las sinusoides generadas en $f = 2f_c$ y $f = 3f_c$ respectivamente, todas están moduladas por distintas versiones distorsionadas de $A(t)$, por lo que se las denomina distorsión armónica no lineal.

Por último, es necesario analizar las componentes de la señal de salida del sistema no lineal en torno a la frecuencia inicial, $f = f_c$, llamada componente en banda, o fundamental. En primer lugar, se obtiene una senoide muy parecida a la señal de entrada, simplemente con un pequeño desfase y una atenuación (o ganancia). El problema surge con la siguiente componente, ya que está a la misma frecuencia que la primera y tiene un desfase del mismo orden, pero está modulada por una versión muy distorsionada de $A(t)$. Si se supone que los desfases son iguales $\phi_1 = \phi_3$, la amplitud de la componente en banda sería $\alpha_1 A(t) + \frac{3}{4} \alpha_3 A(t)^3$. Este tipo de distorsión, conocida como distorsión no lineal en banda, resulta de las más perjudiciales en un sistema de comunicaciones, ya que no se puede eliminar filtrando la señal de salida, como la distorsión armónica.

En conclusión, todos estos efectos se pueden observar fácilmente con un sistema no lineal muy básico como el descrito en la Ec. (2.7), pero para medirlos de forma experimental en un dispositivo concreto, existen algunas técnicas con las que se pueden obtener datos como los valores de los distintos α_i , τ_i o las componentes armónicas que se obtienen a la salida. Entre estas técnicas destacan los test de caracterización de un tono y de dos tonos.

2.2 Técnicas de caracterización experimental de efectos no lineales

Una vez entendida la importancia que tiene conocer los efectos que pueden provocar los dispositivos no lineales, es clave intentar caracterizar dichos efectos en los dispositivos del sistema que se quiera montar, para posteriormente encontrar herramientas con las que combatirlos. Con esta idea, se van a comentar las técnicas de caracterización de dispositivos no lineales básicas más usuales y a su vez se detallarán las figuras de mérito de estos análisis, resultados clave de estas técnicas. En adelante, se usará las siglas DUT para referirse al dispositivo que se somete a las pruebas, del inglés *Device Under Test*.

2.2.1 Test de caracterización de un tono

Esta técnica suele ser la primera que se realiza cuando comienza el proceso de determinación de las singularidades de un sistema. Consiste en el uso de una senoide pura como excitación al DUT, por lo que la entrada al sistema sería igual a la Ec. (2.5) donde $A(t) = A_i$ y $\theta(t) = 0$. En un sistema lineal, la respuesta sería del tipo

$$y_L(t) = A_o \cos[2\pi f t + \phi_o], \quad (2.10)$$

por lo que es trivial pensar que si la respuesta obtenida no sigue esta expresión, el DUT es un dispositivo no lineal. Esto ocurre si la amplitud A_o medida no es una réplica escalada de A_i y si ésta y la diferencia de fase entre la entrada y la salida, ϕ_o , varían de forma no lineal con el nivel de estímulo [4]. En dicho caso, se obtiene una señal de salida que sigue la expresión [3]

$$y_{LN}(t) = \sum_{r=0}^{\infty} A_{or}(f, A_i) \cos[r2\pi f t + \phi_{or}(f, A_i)], \quad (2.11)$$

ya que aparecerían armónicos, como se ha visto en la sección anterior. Así pues, un dispositivo no lineal puede convertir variaciones de la amplitud de entrada A_i en cambios en la amplitud o en la fase de salida, A_{or} o ϕ_{or} respectivamente [4]. Debido a esto, las figuras de mérito más importantes asociadas a las pruebas de un tono son [5]:

- **Característica AM-AM:** Gráfica que enfrenta la amplitud de la señal de salida en la frecuencia fundamental con la de la señal de entrada, como si transformara una posible modulación de amplitud (AM) en la entrada en otra a la salida. Una característica AM-AM típica muestra un comienzo lineal pero sufriendo una compresión al final, de la que deriva el concepto de "punto de compresión". Un ejemplo se puede observar en la Figura 2.4.
- **Característica AM-PM:** De la misma forma que la AM-AM, ésta relaciona la fase de la señal de salida con la amplitud de entrada, debido a la posibilidad de que el DUT no lineal transforme una modulación de amplitud en la entrada en una modulación de fase (PM) a la salida. Es importante mencionar que la característica AM-AM es visible en cualquier sistema no lineal, mientras que la AM-PM es exclusiva de sistemas dinámicos o con efectos de memoria. Un ejemplo se puede observar en la Figura 2.4.
- **Punto de compresión de 1 dB:** Es definido como el nivel de potencia al cual la señal de salida está 1 dB por debajo de la salida que se obtendría si se extrapolara la característica de pequeña señal lineal del sistema. Puede expresarse referido a la entrada o a la salida, y se observa de forma muy simple en una gráfica de la ganancia en función de la potencia de entrada, como la Figura 2.2.
- **Distorsión Armónica Total (THD):** Caracteriza la capacidad del DUT de generar nuevas componentes armónicas. Relaciona la raíz cuadrada de la potencia armónica total de salida y la raíz cuadrada de la potencia en la frecuencia fundamental:

$$THD = \frac{\sqrt{\frac{1}{T} \int_0^T \left[\sum_{r=2}^{\infty} A_{or}(f, A_i) \cos[r2\pi ft + \phi_{or}(f, A_i)] \right]^2 dt}}{\sqrt{\frac{1}{T} \int_0^T [A_{o1}(f, A_i) \cos[2\pi ft + \phi_{o1}(f, A_i)]]^2 dt}} \quad (2.12)$$

2.2.2 Test de caracterización de dos tonos

La prueba de dos tonos es más adecuada que la de un tono, porque además de poder visualizarse los armónicos generados, también permite identificar las componentes debidas a la intermodulación de ambos tonos. Como las señales de comunicaciones suelen tener cierto ancho de banda, la intermodulación debe ser tenida en cuenta si se caracteriza un dispositivo con este propósito [3]. Por tanto, si se aplica una excitación de dos tonos, del estilo $x(t) = A_{i1} \cos(2\pi f_1 t) + A_{i2} \cos(2\pi f_2 t)$, la salida estaría formada por una gran cantidad de términos sinusoidales con todas las posibles combinaciones de $\pm f_1$ y $\pm f_2$, que se puede agrupar en:

$$y_{NL}(t) = \sum_{r=1}^{\infty} A_{or} \cos(2\pi f_r t + \phi_{or}), \text{ donde } f_r = m f_1 + n f_2 \text{ y } m, n \in \mathbb{Z}. \quad (2.13)$$

Analizando esta expresión desde el punto de vista de una señal de banda estrecha, que se define como aquella señal de paso banda cuya frecuencia central es mucho mayor que el ancho de banda ocupado, se puede dividir la distorsión generada en la distorsión en banda y la fuera de banda.

- **Distorsión en banda:** Se considera distorsión en banda aquella que, siguiendo la Ec. (2.13), satisfaga que $m + n = 1$. De esta forma, en la banda de transmisión se podrían obtener medidas en las frecuencias fundamentales, f_1 y f_2 , y componentes de tercer orden ($|m| + |n| = 3$) a $2f_1 - f_2$ y $2f_2 - f_1$, componentes de quinto orden ($|m| + |n| = 5$) a $3f_1 - 2f_2$ y $3f_2 - 2f_1$, y demás componentes de orden impar. Estos productos de distorsión se denominan productos de intermodulación de orden n (IM n). El más importante es el IM3, por el que se define el *punto de intercepto de tercer orden*, IP_3 . Se trata de un punto ficticio obtenido como la intersección entre la extrapolación de la potencia de salida del tono fundamental, que tiene una pendiente de 1 dB/dB, y la extrapolación de la potencia de IM3, con pendiente de 3 dB/dB. En la Figura 2.8 de [3] puede encontrarse una definición gráfica del IP_3 .
- **Distorsión fuera de banda:** Las componentes fuera de banda son los productos de la Ec. (2.13) que cumplen $m + n \neq 1$. Son por tanto los armónicos de cada tono, y también los productos con frecuencia $m f_1 + n f_2$ que caen tanto cerca de la componente DC ($n + m = 0$) como los que rodean los armónicos ($m + n = 2, 3, 4, \dots$). Como son componentes que se sitúan en zonas alejadas de las frecuencias fundamentales, se pueden filtrar de forma simple en sistemas de banda estrecha.

2.2.3 Test de caracterización de un espectro continuo limitado en banda

En los estándares de la industria siempre se han usado las técnicas de uno y dos tonos para caracterizar los dispositivos no lineales. Sin embargo, hoy en día se buscan procedimientos alternativos que ofrezcan unos resultados más parecidos al régimen de operación final del sistema, que no será la transmisión de un tono únicamente, ni de dos, sino que serán señales de comunicación con un espectro cuasi-continuo limitado en una banda. Esto se observa simplemente analizando los estándares actuales de comunicaciones móviles por ejemplo, que siguen modulaciones OFDM. Por este motivo son cada vez más usuales los análisis de comportamiento mediante señales de espectro continuo.

Como se ha explicado en la Subsección 2.1.2 y en la Subsección 2.2.2, el espectro de una señal distorsionada por no linealidades consiste en agrupaciones de componentes frecuenciales centrados en los diferentes armónicos de la portadora. Centrándonos en la distorsión en banda, se observa que la salida contiene muchas más componentes frecuenciales, lo que genera recrecimiento espectral, que afecta en gran medida al canal adyacente. Además, también se obtiene distorsión cocanal de aquellas componentes de intermodulación que caen dentro del canal de la señal. Es muy importante entonces medir de alguna forma cuánto se afecta al canal adyacente, cuánto el canal propio de la señal, y en definitiva, lo distinta que es la señal de salida obtenida con la señal de salida deseada. Las figuras de mérito que nos otorgan dicha información son:

- **Relación de potencia del canal adyacente (ACPR):** Del inglés, *Adjacent Channel Power Ratio*. La distorsión del canal adyacente se refiere a las componentes del recrecimiento espectral que están situadas en los canales adyacentes de la señal y suponen una posible interferencia a éstos. Para caracterizar esta forma de distorsión se utiliza el $ACPR_U$ y el $ACPR_L$, que relaciona la potencia total introducida en el canal adyacente, superior o inferior respectivamente, con la potencia medida en la zona fundamental. Siguiendo la notación de la Figura 2.3 y tomando $Y_o(f)$ como la densidad espectral de potencia de la señal de salida, la expresión de los distintos $ACPR$ sería:

$$ACPR_U = \frac{\int_{f_{U_2}}^{f_{U_1}} Y_o(f) df}{\int_{f_{L_1}}^{f_{L_2}} Y_o(f) df} \quad (2.14)$$

$$ACPR_L = \frac{\int_{f_{L_2}}^{f_{L_1}} Y_o(f) df}{\int_{f_{U_1}}^{f_{U_2}} Y_o(f) df}$$

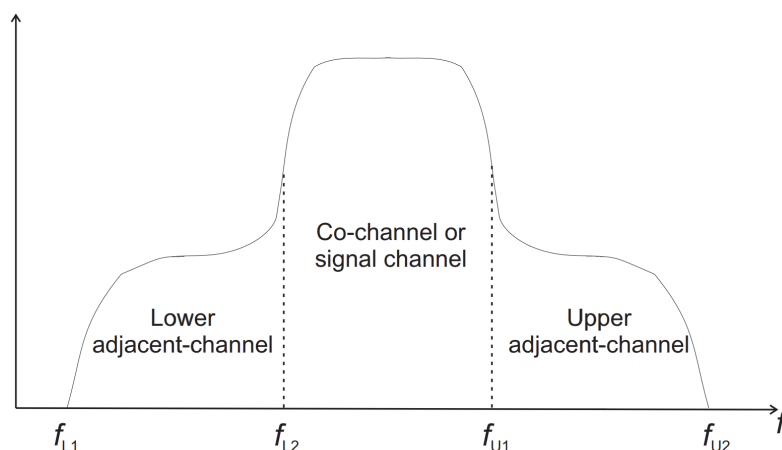


Figura 2.3 Recrecimiento espectral observado en un sistema no lineal excitado por un estímulo modulado de banda estrecha.

- **Magnitud del vector error (EVM):** Del inglés, *Error Vector Magnitude*. Cuando se trabaja con señales moduladas digitalmente la representación de los símbolos transmitidos y recibidos se realiza en una *constelación*: tras pasar la señal recibida por un filtro paso-banda centrado en la banda fundamental y

posteriormente demodular los símbolos obtenidos, éstos son representados como vectores en el espacio señal, que es la representación mediante unas funciones base ortogonales para las que, proyectando la señal, se obtienen las componentes en fase y cuadratura [6]. Realizando este proceso sin presencia de ruido, la dispersión de la constelación se deberá principalmente a la distorsión no lineal en banda. Por tanto, hallando la magnitud del vector error (diferencia entre símbolo demodulado \hat{y} y símbolo transmitido \hat{x}), se obtiene una medida que caracteriza la distorsión cocanal. El EVM se define:

$$\text{EVM} [\%] = \sqrt{\frac{\sum_{l=1}^L |\hat{y}_l - \hat{x}_l|^2}{\sum_{l=1}^L |\hat{x}_l|^2}} \times 100, \quad (2.15)$$

donde l es el índice del símbolo y L el número total de símbolos empleados.

- **Error cuadrático medio normalizado (NMSE):** Del inglés, *Normalized Mean Squared Error*. Las dos figuras de mérito anteriores se centran cada una por separado en distintos aspectos de la distorsión no lineal. Sin embargo, puede ser útil realizar una medida rápida que nos dé información completa de lo distinta que es la señal obtenida $y_{med}[k]$ con la señal de salida si el sistema fuera lineal $y_{ideal}[k]$. Por este razonamiento se emplea el NMSE, que no es más que una normalización del error cuadrático medio entre estas dos señales, cuya expresión es

$$\text{NMSE} [\text{dB}] = 10 \log \left\{ \frac{\sum_{k=1}^N (y_{ideal}[k] - y_{med}[k])^2}{\sum_{k=1}^N |y_{med}[k]|^2} \right\}. \quad (2.16)$$

2.3 Efectos de memoria

Según el diccionario de la Real Academia de la Lengua Española, "*la memoria es la facultad psíquica por medio de la cual se retiene y recuerda el pasado*". En el ámbito de los dispositivos electrónicos, en concreto de los amplificadores de potencia, los efectos de memoria es el término usado para describir la influencia de la historia de la señal en su valor actual. Esto es, los valores de una señal en el pasado contribuyen al valor en el presente. Un sistema con memoria es aquel en el que el valor de la señal de salida depende de los valores anteriores de la señal de entrada, $y[n] = f(x[n], x[n-1], x[n-2], \dots, x[n-Q])$. Es importante mencionar que el peso de los valores decrece con el tiempo que ha pasado, por lo que un cierto valor, $x[n-Q-1]$, tendrá una influencia despreciable. Ese efecto se llama *fading memory*, "memoria desvaneciente", y el número de muestras que contribuyen en el presente, Q , se denomina *memory depth*, "profundidad de la memoria". En general, un sistema no lineal sin memoria solo puede causar distorsión de amplitud, nunca de fase, por lo que si un sistema presenta distorsión de fase, posee obligatoriamente efectos de memoria [5].

Los efectos de memoria no son más que el comportamiento dinámico del sistema. No es ninguna sorpresa que dichos efectos existan, ya que los elementos electrónicos básicos, como el condensador o la bobina, también cuentan con estos efectos: la corriente que fluye en un circuito depende de cuánto se ha cargado la capacidad anteriormente. En general, son comportamientos debidos al almacenamiento de energía en el sistema, y pueden tener características temporales del orden del periodo de la señal, *memoria a corto plazo*, o mucho más lentos, *memoria a largo plazo*.

- **Efectos de memoria a corto plazo:** resumidamente, son producidos por las reactancias asociadas al transistor, descritas en el modelo circuital equivalente, y por las redes de adaptación empleadas, ya que suelen estar formados por componentes reactivos o líneas de transmisión.
- **Efectos de memoria a largo plazo:** se deben a tres causas principales, que son los efectos térmicos que se suelen dar para frecuencias de modulación por debajo de 100 kHz, los estados trampa producidos por imperfecciones en los semiconductores que dan lugar a variaciones en la escala de los kilohertzios a los megahertzios, y las redes de polarización, que proporcionan caminos de baja impedancia para la componente DC a la vez que una alta impedancia para la señal de RF.

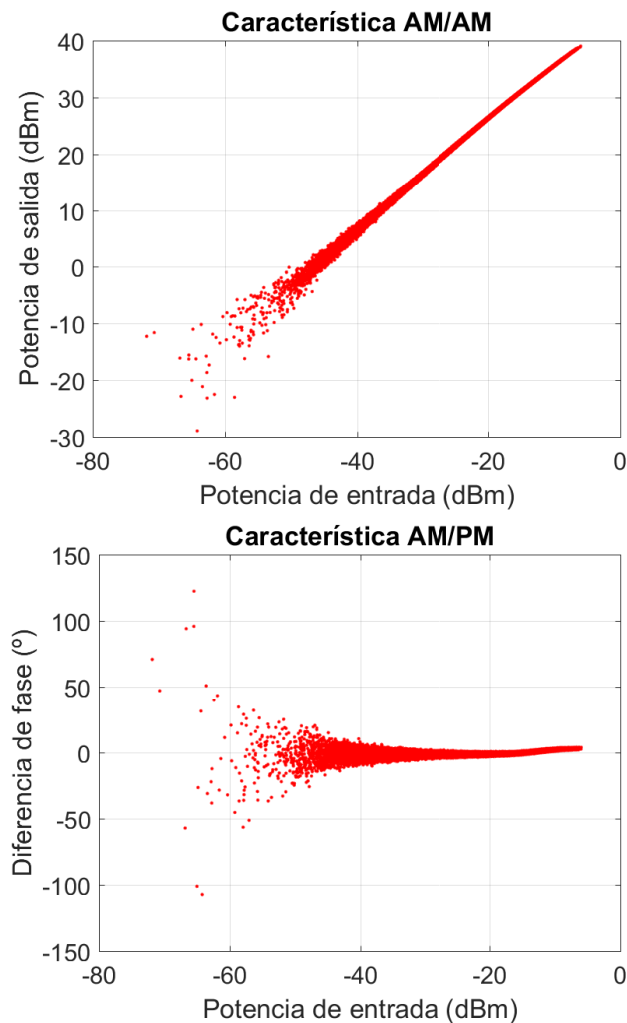


Figura 2.4 Características AM/AM y AM/PM de un amplificador de potencia, donde se aprecia la dispersión de los puntos debida a los efectos de memoria.

Una vez analizados los efectos de memoria, qué son y cuáles son las causas de sus apariciones, es interesante comentar que la forma más sencilla de "visualizar" dichos efectos es a través de características AM/AM y AM/PM del dispositivo. La dispersión que sufre estas gráficas como se puede observar en la Figura 2.4 que las hacen parecer una nube de puntos se debe esencialmente a los efectos de memoria. A mayor dispersión, los efectos de memoria son más notorios. Otra forma de observar este comportamiento de los dispositivos no lineales dinámicos es mediante las asimetrías del espectro. Por lo general, el espectro de la señal de salida de un dispositivo no lineal es simétrico con respecto a la frecuencia fundamental, ya que los armónicos y los productos de intermodulación se generan a la misma distancia y con la misma potencia a cada lado. Sin embargo, cuando el dispositivo presenta efectos de memoria, esto no es así, y normalmente se puede distinguir a simple vista que el espectro no es simétrico. En algunos casos, hay que calcular el ACPR para darse cuenta de que hay una diferencia apreciable entre el recrecimiento espectral a la izquierda y a la derecha.

3 Series de Volterra y Amplificadores de Potencia

¿Quién es más loco: el loco o el loco que sigue al loco?

OBI-WAN KENOBI EN *Star Wars: Episodio IV - Una nueva esperanza*, 1977

El desafío de modelar los efectos que provocan los dispositivos no lineales, como el amplificador de potencia, implica desarrollar o emplear herramientas matemáticas que puedan predecir estos comportamientos. La herramienta más usada en el análisis de circuitos no lineales es la serie de Volterra. Desarrollada por Norbert Wiener en 1942 a partir de la teoría matemática desarrollada por Vito Volterra (1860-1940), ha sido "redescubierta" recientemente ya que las distorsiones en los circuitos de comunicaciones han vuelto a ser más significativos en el diseño de circuitos. Gracias a ella, los distintos problemas de trabajar con dispositivos no lineales pueden ser tratados de forma eficiente y ordenada. Por estos motivos, las series de Volterra han sido clave en el desarrollo de modelos de comportamientos de amplificadores de potencia en radiofrecuencia, y en la linealización de estos usando técnicas de predistorsión digital.

A continuación se introducirá esta herramienta de forma matemática, y posteriormente se expondrán distintos modelos de comportamiento basados en series de Volterra.



Figura 3.1 Retrato de Vito Volterra, matemático y físico italiano cuyas investigaciones del análisis funcional son la base del desarrollo de las series matemáticas que llevan su nombre.

3.1 Introducción y desarrollo de las series de Volterra

La teoría de Volterra introdujo los funcionales en el mundo de las matemáticas, llamando así a las operaciones de funciones cuyo resultado es un número [7]. A su vez, una de las propiedades de los funcionales que demostró fue que cualquier funcional $G[x]$, continuo en el campo de funciones continuas podía ser representado por la expansión

$$G[x] = \sum_{n=0}^{\infty} F_n[x] \quad (3.1)$$

donde $F_n[x]$ es un funcional regular homogéneo de la forma

$$F_n[x] = \int_a^b \cdots \int_a^b h_n(\xi_1, \dots, \xi_n) x(\xi_1) x(\xi_2) \cdots x(\xi_n) d\xi_1 d\xi_2 \cdots d\xi_n, \quad (3.2)$$

donde n se denomina el grado del funcional. A partir de esto, Wiener percibió que la salida de un sistema no lineal $y(t)$ era un funcional de su entrada $x(t)$, por lo que $y(t)$ podría ser expresada como una expansión funcional de $x(t)$

$$y(t) = \sum_{n=1}^{\infty} y_n(t), \quad (3.3)$$

siendo la expresión completa de cada funcional de grado n de la forma

$$y_n(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \dots, \tau_n) x(t - \tau_1) x(t - \tau_2) \cdots x(t - \tau_n) d\tau_1 d\tau_2 \cdots d\tau_n. \quad (3.4)$$

Cabe mencionar que se puede demostrar que conforme aumenta n , el grado del funcional, la contribución de éste a $y(t)$ disminuye, por lo que la Ec. (3.3) se suele aproximar como

$$y(t) = \sum_{n=1}^N y_n(t). \quad (3.5)$$

Así pues, una serie de volterra es la unión de la Ec. (3.4) y la Ec. (3.5), y con ella se consigue expresar de forma clara y ordenada todas las posibles combinaciones de la señal de entrada de un dispositivo no lineal consigo misma con todos los retardos posibles, y cada una de estas combinaciones está multiplicada por un coeficiente $h_n(\tau_1, \tau_2, \dots, \tau_n)$ denominado *kernel* que indica el peso que tiene esa combinación exacta en el comportamiento final del dispositivo. Como apunte final, hay que tener en cuenta que la Ec. (3.4) supone que las señales son continuas, pero lo normal es trabajar con ellas mediante programas computacionales, que trabajan en tiempo discreto y con memoria finita. Por ello, se definen nuevas expresiones de los funcionales $y_n(t)$ y la suma $y(t)$ en tiempo discreto como

$$y[k] = \sum_{n=1}^N y_n[k] = \sum_{n=1}^N \sum_{q_1=0}^Q \cdots \sum_{q_n=0}^Q h_n(q_1, \dots, q_n) \prod_{j=1}^n x[k - q_j] \quad (3.6)$$

Hay varias perspectivas desde las que se puede observar y analizar esta herramienta matemática, y en este documento se van a exponer dos de ellas.

3.1.1 Serie de Volterra como desarrollo de una serie de Taylor

Una forma de contruir una serie de Volterra es como desarrollo de un modelo no lineal instantáneo que puede ser descrito por una expansión de serie de Taylor. Como inicio, se tomará el prototipo de modelo polinomial sin memoria (ML, del inglés *Memory-Less*) de [8], que inspira la misma idea que la Ec. (2.7):

$$y(t) = a_0 + \sum_{n=1}^N a_n x^n(t). \quad (3.7)$$

Los coeficientes polinómicos a_n se hallan por la serie de Taylor de la relación de la entrada y salida, expandida alrededor de un punto de operación x_0 . Este punto de operación viene definido por la tensión de polarización

o de DC cuando se modela el comportamiento no lineal de un amplificador. Se obtiene por tanto la expresión

$$\begin{aligned} y(x(t)) &= y(x)|_{x=x_0} + \frac{1}{2!} \frac{dy}{dx} \Big|_{x=x_0} (x-x_0) + \frac{1}{3!} \frac{d^2y}{dx^2} \Big|_{x=x_0} (x-x_0)^2 + \dots \\ &= a_0 + a_1 x(t) + a_2 x(t)^2 + \dots \end{aligned} \quad (3.8)$$

Sin embargo, si en vez de ser una relación sin memoria, se considera la salida $y(t)$ como función de $x(t)$ y de los valores de x pasados, se describen así los efectos de memoria que la serie de Taylor no tiene en cuenta:

$$y(t) = f(x, x_1, x_2, \dots, x_n) \text{ donde } x = x(t), x_1 = x(t - \tau_1), x_2 = x(t - \tau_2), \dots \quad (3.9)$$

Por tanto, la expansión de la serie de Taylor con esto quedaría

$$\begin{aligned} y(x(t)) &= y(x)|_{x=x_0} + \frac{1}{2!} \frac{dy}{dx} \Big|_{x=x_0} (x-x_0) + \frac{1}{2!} \frac{dy}{dx_1} \Big|_{x=x_0} (x_1-x_0) + \dots \\ &+ \frac{1}{3!} \frac{d^2y}{dx^2} \Big|_{x=x_0} (x-x_0)^2 + \frac{1}{3!} \frac{d^2y}{dx_1^2} \Big|_{x=x_0} (x_1-x_0)^2 + \dots \\ &+ \frac{1}{3!} \frac{d^2y}{dx dx_1} \Big|_{x=x_0} (x-x_0)(x_1-x_0) + \dots \end{aligned} \quad (3.10)$$

que es una serie multinomial. Analizando esta expresión se observan fácilmente los términos de memoria $(x_1 - x_0)$ y los denominados términos cruzados de la serie de Volterra $(x - x_0)(x_1 - x_0)$, localizables en la Ec. (3.6). En conclusión, la serie de Volterra puede verse como una generalización de una serie de Taylor con memoria.

3.1.2 Serie de Volterra como desarrollo de un sistema lineal

Otra forma de contruir una serie de Volterra es pensar primero en los sistemas lineales y su representación. En la literatura, como en [9] o en [8] se representa la relación entrada-salida de un sistema lineal mediante la convolución con la respuesta al impulso, siendo ésta la que modela el comportamiento del sistema, que se expresa como

$$y(t) = h(t) * x(t) = \int_0^t h(\tau)x(t - \tau)d\tau. \quad (3.11)$$

Para modelar los primeros efectos no lineales, un primer paso sería multiplicar la señal de entrada dos veces, obteniendo así la posibilidad de efectos cuadráticos. Para modelar los posibles efectos de memoria, cada señal debería tener un retraso propio por lo que se obtendría la expresión

$$y_2(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_2(\tau_1, \tau_2)x(t - \tau_1)x(t - \tau_2)d\tau_1 d\tau_2, \quad (3.12)$$

y continuando con este razonamiento hasta el n -ésimo orden, la expresión sería

$$y_n(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_n(\tau_1, \tau_2, \dots, \tau_n)x(t - \tau_1)x(t - \tau_2) \dots x(t - \tau_n)d\tau_1 d\tau_2 \dots d\tau_n. \quad (3.13)$$

Los límites de integración se han extendido a $[-\infty, \infty]$ para dar más generalidad a estas expresiones, aunque el resultado sería el mismo ya que los sistemas con los que se trabajará serán causales e invariantes en el tiempo. Por lo tanto, el modelo completo sería la suma de todos estos comportamientos, desde el lineal hasta el de orden n -ésimo, obteniendo así la misma expresión que en la Ec. (3.3). El *kernel* $h_n(\tau_1, \tau_2, \dots, \tau_n)$, equivalente al de Volterra, se puede ver como el coeficientes de cada componente de este modelo, regulando así la importancia de cada expresión en el comportamiento final del dispositivo no lineal. En conclusión, otra forma de ver la serie de Volterra es como una convolución generalizada al n -ésimo orden, o de la misma forma, la convolución de un sistema lineal es un caso particular de la serie de Volterra.

3.2 Modelos de comportamiento de amplificadores de potencia basados en series de Volterra

Este trabajo está centrado en los modelos de amplificadores de potencia, con el objetivo de que una vez modelada la distorsión que genera el amplificador, mediante técnicas de predistorsión digital se pueda eliminar dichos efectos. Por tanto, el primer paso es el modelaje, para el cual las series de Volterra han sido la herramienta más usada. Sin embargo, al tratarse de una serie infinita, es imposible trabajar computacionalmente con ella. Por ello, en esta sección se introducirán los modelos más importantes en la literatura.

Antes de todo, es importante dejar claro que lo más usual no es trabajar con la señal en RF tal y como se transmitiría, sino que se usa la envolvente compleja de la señal, normalmente denotada como \tilde{x} y definida según la expresión

$$x(t) = \text{Re} \{ \tilde{x}(t) e^{j\omega_c t} \}. \quad (3.14)$$

Siguiendo la Ec. (2.5), la envolvente compleja de ésta sería $\tilde{x}(t) = A(t) e^{j\theta(t)}$. Para más facilidad al lector, en el resto del documento se usará $x(t)$ para designar a esta envolvente compleja. Además, y como se comentó anteriormente, computacionalmente es imposible trabajar en tiempo continuo, por lo que las expresiones de las señales y de los modelos se expresarán siempre en tiempo discreto. Además, se usará el término *regresor* para referirse a cada una de las posibles combinaciones de la señal de entrada con los distintos desplazamientos temporales, las cuales son multiplicadas por los *kernels* para formar la serie de Volterra.

3.2.1 Modelo completo de Volterra

El modelo completo de Volterra, o modelo FV, del inglés *Full-Volterra*, es una implementación directa de la serie de Volterra en función de la envolvente compleja, con alguna variación. La expresión de este modelo es [10]

$$y_{FV}[k] = \sum_{p=1}^P \sum_{q_1=0}^Q \sum_{q_2=q_1}^Q \cdots \sum_{q_p=q_{p-1}}^Q \sum_{q_{p+1}=0}^Q \sum_{q_{p+2}=q_{p+1}}^Q \cdots \quad (3.15)$$

$$\cdots \sum_{q_{2p-1}=q_{2p-2}}^Q h_{2p-1}(q_1, \dots, q_{2p-1}) \times \prod_{i=1}^p x[k - q_i] \prod_{j=p+1}^{2p-1} x^*[k - q_j].$$

Lo primero que hay que mencionar es que el orden de los posibles regresores es siempre impar, siendo el orden máximo $2P - 1$. Esto se debe a que el propio desarrollo de la serie de Volterra completa en términos de la envolvente compleja lleva a que solo existan órdenes impares [11]. Como se está trabajando con envolventes complejas, para que la frecuencia fundamental de $y[k]$ sea la misma que $x[k]$, si en cada producto hay $2p - 1$ factores de x , debe haber $p - 1$ de esos factores que estén conjugados. Por ese motivo el productorio se divide en dos. En cuanto a los retrasos, para que no se repitan combinaciones de éstos, los sumatorios están encadenados, tomando así los valores de todas las posibles combinaciones de desplazamientos temporales, hasta un valor Q , que corresponde a la profundidad de la memoria, mencionada en la Sección 2.3.

3.2.2 Modelo Polinomial con Memoria Generalizado

Este modelo, también conocido como modelo GMP por sus siglas en inglés, *Generalized Memory Polynomial* [12], parte del modelo polinomial con memoria (MP). El modelo MP es un modelo muy simple que únicamente consiste en incorporar efectos de memoria simples al modelo no lineal estático [8]. La expresión que resume este modelo es

$$y_{MP}[k] = \sum_{n=1}^N \sum_{m=0}^M a_{nm} x[k - q_m] |x[k - q_m]|^{n-1}. \quad (3.16)$$

Pese a que la notación sea distinta, en esencia es una serie de Volterra, pero que sólo incluye los términos denominados "diagonales". En otras palabras, los términos de segundo grado son del tipo $x^2[k], x^2[k - 1], \dots, x^2[k - N]$; los de tercer orden, del tipo $x^3[k], \dots, x^3[k - N]$; sin embargo, no hay términos cruzados de memoria, como por ejemplo $x[k] \cdot x[k - 1] \cdot x[k - 3]$. Cada polinomio es función de una única señal retrasada. Esta es la característica principal del modelo MP, ya que los términos aquí presentes son los que suelen resultar predominantes a la hora de modelar amplificadores de potencia. Sin embargo, truncar tanto los términos de la serie de Volterra puede ser excesivo. Por este motivo, se introdujo en [12] el modelo GMP, que permitía

introducir algunos términos cruzados, en concreto, los diagonales superiores e inferiores de los existentes en el MP. De esta forma, la expresión que sigue el modelo generalizado es

$$\begin{aligned}
 y_{GMP}[k] = & \sum_{n=1}^N \sum_{m=0}^M a_{nm} x[k - \tau_m] |x[k - \tau_m]|^{n-1} \\
 & \sum_{n=1}^N \sum_{m=0}^M \sum_{s=1}^S b_{nms} x[k - \tau_m] |x[k - \tau_m - \tau_s]|^{n-1} \\
 & \sum_{n=1}^N \sum_{m=0}^M \sum_{s=1}^S c_{nms} x[k - \tau_m] |x[k - \tau_m + \tau_s]|^{n-1}.
 \end{aligned} \tag{3.17}$$

Se observa entonces que el primer sumatorio equivale al modelo MP, en la Ec. (3.16), y con los siguientes se consigue añadir las componentes cruzadas. Con la variable S se decide cuántos nuevos elementos cruzados se van a añadir, y cada uno de estos nuevos términos tiene su correspondiente *kernels*, b_{nms} o c_{nms} . De esta forma, el modelo GMP cuenta con menos componentes que el modelo FV, pero los que se tienen en cuenta en el primero son los más importantes, por lo que se consigue una menor carga computacional sin perder mucha información. Resulta interesante comentar que los modelos MP y GMP son propuestas *ad-hoc*, es decir, no están basadas en ningún razonamiento matemático. En este sentido, sus autores los propusieron siendo posible que contuvieran tanto términos pares como impares, es decir, $n = 1, 2, 3, \dots$. Sin embargo, para poder hacer comparaciones entre el modelo FV y el MP o GMP, no se deberían tener en cuenta los términos pares, al igual que tampoco se debería incluir los términos no causales del tercer sumatorio del modelo GMP.

3.2.3 Modelo de series de Volterra de valores complejos

El modelo CVS (del inglés *Complex-valued Volterra Series*) es una extensión de las series de Volterra al caso de un sistema no lineal con señales complejas [13].

Idealmente, la envolvente compleja de la señal de salida de RF es una transformación lineal de la señal banda base de valor complejo $x[k] = x_I[k] + jx_Q[k]$. Operar con señales complejas como si fueran de valor real no está justificado, y la conversión de modelos paso de banda de valor real en modelos banda base de valor complejo no siempre es inmediata. Además, para explotar la caracterización estadística completa de los datos en el procesado de señales de valor complejo, se requiere acceder a la información contenida en la correlación complementaria introduciendo transformaciones ampliamente lineales (WL, del inglés *Widely Linear*) [14]. La correlación complementaria se define como la correlación de la señal, de valor complejo, con su complejo conjugado, y tiene valor cero para señales propias. Si se tiene en cuenta la información contenida en la correlación complementaria de señales impropias, el tipo de procesado correcto puede proporcionar mejoras significativas en las prestaciones.

Cuando se considera un modulador I/Q lineal con imperfecciones, la envolvente compleja de la señal de RF a la salida depende explícitamente tanto de la señal de entrada $x[k]$ como de la señal imagen $x^*[k]$, y dicho modulador puede considerarse un sistema WL. Por otro lado, un modelo más realista de un modulador I/Q debería incluir los posibles efectos no lineales derivados de los elementos que lo componen, para lo que es necesaria una función no lineal que transforme $x[k]$ en la envolvente compleja de la salida de RF. Para las señales de valor real, el enfoque de las series de Volterra es una opción natural. Sin embargo, no hay una transformación formal equivalente para el caso de sistemas no lineales con señales de entrada de valor complejo y los investigadores han desarrollado modelos que proporcionan soluciones particulares a problemas específicos en sus respectivos campos. Sin embargo, el uso de estos modelos particulares no está justificado para describir el comportamiento de otro tipo de sistemas.

En este contexto, aparece el modelo CSV, que no es más que la representación mediante una serie doble de Volterra, en la que las dos señales de entrada son $x[k]$ y $x^*[k]$. La expresión final del modelo sería [13]

$$\begin{aligned}
 y[k] = & h_{0,0} + \sum_{n=1}^{\infty} \left\{ \sum_{q_n=0}^{\mathcal{Q}_{n,0}} h_{n,0}[\mathbf{q}_n] \prod_{r=1}^n x[k - q_r] \right. \\
 & + \sum_{m=1}^{n-1} \sum_{q_n=0}^{\mathcal{Q}_{n-m,m}} \sum_{p_m=0}^{P_{n-m,m}} h_{n-m,m}[\mathbf{q}_{n-m}, \mathbf{p}_m] \prod_{r=1}^{n-m} x[k - q_r] \times \prod_{s=1}^m x^*[k - p_s] \\
 & \left. + \sum_{q_n=0}^{\mathcal{Q}_{0,n}} h_{0,n}[\mathbf{q}_n] \prod_{s=1}^n x^*[k - q_s] \right\}.
 \end{aligned} \tag{3.18}$$

Este modelo por tanto puede considerarse válido para modelar las imperfecciones de moduladores I/Q, para describir la envolvente compleja de la salida de un amplificador de potencia en la banda del armónico fundamental a partir de la envolvente compleja de la entrada o incluso para la compensación conjunta de las imperfecciones debidas al modulador I/Q y el amplificador presentes en un transmisor de comunicaciones. Obsérvese que el modelo CVS contiene términos de segundo orden del tipo xx , xx^* y x^*x^* , y otros órdenes pares superiores. El modelo FV es un caso particular del CVS que sólo presenta términos impares en los que las versiones retrasadas de x aparecen $m + 1$ veces y las versiones retrasadas de x^* aparecen m veces. Como caso particular del FV y por tanto del CVS, el modelo MP solo contempla términos del tipo $|x[k - q]|^{2m}x[k - q]$ si éste cuenta únicamente con términos impares.

Por último, el aumento en la complejidad del modelo CVS con respecto a otros casos como FV o GMP es evidente debido a su estructura con un mayor número de parámetros. Por eso, la propuesta del modelo CVS va ligada a la utilización de alguna estrategia de reducción del número de coeficientes, foco principal del capítulo siguiente.

4 Técnicas de selección de componentes

*She told me a bit of madness is key
to give us new colors to see*

MIA DOLAN EN *La Ciudad De Las Estrellas: La La Land*

El principal problema que tienen las series de Volterra y en definitiva los modelos de comportamiento de amplificadores de potencia es que, como las no linealidades que éstos presentan pueden ser muy diversas, necesitan contar con muchas diferentes combinaciones de órdenes y de memoria. Para entender por qué esto es un problema, se va a exponer un pequeño ejemplo: Tras hacer muchas pruebas experimentales y mediciones, se concluye que en el comportamiento de un amplificador, la componente $x(k) \cdot x(k-1) \cdot x^*(k)$ es muy importante, por lo que es bastante conveniente tenerla en cuenta a la hora de modelarlo, para que el error de modelado sea lo mínimo posible. Sin embargo, siguiendo por ejemplo el modelo FV, explicado en la Subsección 3.2.1, incluir esta componente de tercer orden y de dos muestras de profundidad de memoria implica incluir todas las demás componentes de estas características, un total de 39 componentes más. Por tanto, ¿resulta eficiente tener en cuenta 40 coeficientes cuando se sabe que el único importante es uno concreto? En definitiva, cuanto mayor orden y mayor memoria se quiere tener en cuenta con un modelo, el error de modelado es menor pero el número de componentes que hay que tener en cuenta crece exponencialmente. Esto hace que la complejidad computacional aumente considerablemente, por lo que la solución más usual es el podado del modelo, o *pruning*, que sencillamente es eliminar términos $h_n(q_1, \dots, q_n) = 0$.

Este *pruning* puede seguir dos estrategias distintas: realizar un podado *ad-hoc* [15], que elimina términos teniendo en cuenta información *a priori*, o realizar algún tipo de razonamiento, ya sea cualitativo o siguiendo alguna estrategia matemática. Este capítulo trata de estos últimos: técnicas de ordenación y selección de coeficientes aplicables a modelos de comportamiento basados en series de Volterra. Se debe mencionar que realmente los modelos GMP y MP de la Sección 3.2 son el resultado de realizar un podado *ad-hoc* de las componentes de la serie de Volterra, siendo soluciones propuestas por los autores.

4.1 Obtención de coeficientes de los modelos de comportamiento

Hasta ahora se ha hablado de los modelos de comportamiento y los coeficientes o *kernels* de estos, pero no se ha abordado la forma de actuar para hallarlos. Es trivial que para modelar un subsistema de comunicaciones se necesitan medidas de señales de entrada y salida de éste. Con estos datos es relativamente sencillo hallar un modelo válido gracias a la notación matricial. Con esta herramienta se puede representar cualquier modelo con la expresión

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{h}}. \quad (4.1)$$

$\hat{\mathbf{y}}$ es un vector columna que contiene los M valores de la envolvente compleja de la señal de salida modelada, o lo que es lo mismo $\hat{\mathbf{y}} \in \mathbb{C}^{M \times 1}$. Por otro lado, $\mathbf{X} \in \mathbb{C}^{M \times N}$ es una matriz con N columnas que equivalen a los N regresores del modelo, explicados en la Sección 3.2, estando formado cada uno por M muestras; y $\hat{\mathbf{h}} \in \mathbb{C}^{N \times 1}$ representa los N coeficientes estimados del modelo. Analizando por tanto la Ec. (4.1), se observa que en efecto la relación entre $\hat{y}[k]$ y $x[k]$ es no lineal. Sin embargo, la relación entre $\hat{y}[k]$ y los coeficientes sí es lineal, lo que supone que con medidas válidas de la señal de salida con su correspondiente señal de entrada

para formar el vector $\hat{\mathbf{y}}$ y la matriz \mathbf{X} , es posible obtener el vector \mathbf{h} simplemente realizando operaciones algebraicas. Sin embargo, esto no es tan fácil, porque realmente los modelos no son perfectos y la ecuación que relaciona las dos señales medidas $x[k]$ e $y[k]$ con los coeficientes \mathbf{h} es

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{e}. \quad (4.2)$$

Esta ecuación, conocida como la "ecuación de medida", refleja el error de modelado como un vector $\mathbf{e} \in \mathbb{C}^{M \times 1}$ y la señal de salida medida como $\mathbf{y} \in \mathbb{C}^{M \times 1}$. Siguiendo esta expresión, una de las posibles maneras de obtener los valores de los *kernels* del modelo es mediante la minimización del sumatorio del error cuadrático residual (RSS, del inglés *Residual Sum of Squared error*), que no es más que obtener la solución de mínimos cuadrados. El RSS se define como la norma al cuadrado del vector error

$$\text{RSS} = \|\mathbf{e}\|_2^2 = \|\mathbf{X}\mathbf{h} - \mathbf{y}\|_2^2. \quad (4.3)$$

Minimizando el RSS con respecto a \mathbf{h} se obtiene una estimación de los coeficientes, $\hat{\mathbf{h}}$ que se define como

$$\hat{\mathbf{h}} = (\mathbf{X}^H \mathbf{X})^{-1} \mathbf{X}^H \mathbf{y} = \mathbf{X}^\dagger \mathbf{y}, \quad (4.4)$$

donde H representa el operador hermítico y \mathbf{X}^\dagger es la pseudoinversa de Moore-Penrose [16]. En definitiva, esta es una estimación válida de los coeficientes del modelo y se conoce como solución LS (del inglés, *Least Squares*). Sin embargo, cuando las columnas de la matriz \mathbf{X} presentan una alta correlación, la inversa $(\mathbf{X}^H \mathbf{X})^{-1}$ presenta grandes problemas de estabilidad y produce un modelo con una varianza muy alta. Los modelos con una varianza alta no son deseables, porque presentan soluciones muy diferentes cuando se realiza varias veces el mismo experimento con pequeños cambios a su entrada. Para combatir la alta varianza del modelo, el primer paso es realizar una transformación a $\hat{\mathbf{h}}$ o a la forma de obtenerlo que la reduzca. Esto se realiza con la Regularización de Ridge, que es un caso particular de la Regularización de Tikhonov, y propone realizar

$$\min \|\mathbf{X}\mathbf{h} - \mathbf{y}\|_2 + \lambda \|\mathbf{h}\|_\ell \quad (4.5)$$

con $\ell = 2$. La solución analítica se obtiene añadiendo al producto $\mathbf{X}^H \mathbf{X}$ de la Ec. (4.4) una matriz diagonal para reducir el número de condicionamiento¹. La solución de los *kernels* estimados que se obtiene es por tanto

$$\hat{\mathbf{h}} = (\mathbf{X}^H \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^H \mathbf{y}, \quad (4.6)$$

donde λ es el parámetro de la regularización, y se obtiene por lo general mediante una validación cruzada de los datos. Con esto por tanto se soluciona la alta varianza de la solución LS, pero aún así, el número de coeficientes es alto y el propósito principal del algoritmo debe ser reducirlos, o lo que es lo mismo, que un cierto número de coeficientes tengan valor nulo y esas componentes estén inactivas. Esto se denomina una solución *k-sparse*, que significa que únicamente k coeficientes son distintos de cero.

En conclusión, hay dos problemas que hay que solucionar: los modelos tienen muchos coeficientes y la solución LS es muy inestable. Por lo tanto, para el primer problema habría que idear técnicas que localicen los de mayor importancia para llegar a una solución de compromiso entre la minimización del error de modelado y la de la complejidad computacional. Para el segundo, a pesar de ser la solución matemáticamente óptima se obtienen comportamientos perjudiciales para el propósito final del proyecto, por lo que habría que hallar otra estrategia o hacer alguna transformación al modelo para obtener modelos más estables.

4.2 Estrategias de reducción de componentes activos

En la literatura hay multitud de posibles estrategias que se pueden tomar para solucionar la inestabilidad de la solución LS y cantidad de coeficientes. En este trabajo, se han analizado tres técnicas: las dos primeras forman parte de la familia de algoritmos voraces (*greedy algorithm* en inglés), que son algoritmos iterativos cuya principal característica es que eligen la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima. *Orthogonal Matching Pursuit* (OMP) e *Iterated Ridge Regression* (IRR) son los dos métodos discutidos en el presente documento. Por otro lado, otra forma de actuar sería realizar una transformación de la matriz de regresores \mathbf{X} para reducir la dimensión y pasar a un espacio ortogonal,

¹ El número de condicionamiento es la relación entre el mayor autovalor de la matriz y el menor, y representa la inestabilidad del sistema. A menor número de condicionamiento, menos variante es la solución obtenida.

reduciendo así la varianza de las componentes y en consecuencia la inestabilidad de éste. Una de las técnicas más básicas y la que se va a analizar en este trabajo será el Análisis de Componentes Principales (PCA, del inglés *Principal Components Analysis*).

4.2.1 Orthogonal Matching Pursuit

Con estos algoritmos, se desea obtener una solución k -sparse. Sin embargo, la Ec. (4.6) no ofrece esta característica deseada, pero si la normalización ℓ_2 de la Ec. (4.5) se cambia a una normalización ℓ_1 , sí se obtiene este comportamiento. La minimización de ℓ_1 se conoce comunmente como técnica LASSO (del inglés *Least Absolute Selection and Shrinkage Operator*) [17]. El problema de esta transformación es que la Ec. (4.5) con $\ell = 1$, no se puede resolver analíticamente, por lo que se abordan distintas vías. Una de ellas es la utilización de algoritmos voraces, seleccionando en cada iteración una componente basándose en algún criterio de optimización local. Es el caso del algoritmo OMP [18], cuyos pasos principales se detallan en el Alg. 1. En particular, el algoritmo OMP elige la componente que presenta una correlación normalizada máxima entre los regresores de la matriz \mathbf{X} y el vector de residuo \mathbf{r} que se redefine en cada iteración como la diferencia de la señal de salida medida, \mathbf{y} , y la señal de salida modelada $\hat{\mathbf{y}}$. Se debe comentar que en el Alg. 1 se denota como $S^{(t)}$ al conjunto de soporte en la iteración t , esto es, al set de elementos seleccionados hasta esta iteración, que "sirven de soporte" al modelo; \mathbf{X}_i se refiere a la columna i -ésima de la matriz \mathbf{X} ; y como antes se ha mencionado, \mathbf{X}^\dagger es la pseudoinversa de Moore-Penrose. La selección del coeficiente i -ésimo se hace en los pasos 3 y 4 y sigue el criterio de máxima correlación

$$\mathbf{i}^{(t)} = \arg \max_{i \notin S^{(t-1)}} \left| \frac{\mathbf{X}_{\{i\}}^H \mathbf{r}^{(t-1)}}{\|\mathbf{X}_{\{i\}}\|_2} \right|. \quad (4.7)$$

Finalmente, el algoritmo para cuando se seleccionan el total de coeficientes del modelo, obteniendo de esta forma las componentes ordenadas de mayor a menor importancia, o cuando se llega al número de coeficientes requeridos.

Algoritmo 1 Resumen del Algoritmo OMP para Series de Volterra

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

Salida: $S^{(t)}$

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

Versiones más avanzadas de este algoritmo han sido desarrolladas, como por ejemplo el CoSaMP, abreviatura de *Compressed-Sampling-Based Unstructured Pruning*, que realmente se trate del OMP añadiendo una umbralización en la iteración [19]; o el DOMP, siglas para *Doubly Orthogonal Matching Pursuit*, que realiza otra ortogonalización en la iteración además de la propia del OMP [20].

4.2.2 Iterated Ridge Regression

La método IRR aplica una de las soluciones a la Ec. (4.6), que no se puede resolver analíticamente. Se propone en [17] y consiste en realizar una regresión iterativa de dicha ecuación, con la inclusión del término $\text{diag}(|\hat{\mathbf{h}}|^{-1})$. Se ha demostrado que este proceso proporciona una buena solución en términos de estabilidad en cada iteración ya que realiza un seguimiento del resultado por el algoritmo Newton-Raphson [21]. De esta forma, la expresión de los *kernels* estimados es

$$\hat{\mathbf{h}} = \left(\mathbf{X}^H \mathbf{X} + \lambda \text{diag}(|\hat{\mathbf{h}}|^{-1}) \right)^{-1} \mathbf{X}^H \mathbf{y}. \quad (4.8)$$

El proceso que sigue esta técnica de obtención y selección de coeficientes de modelos de comportamiento basados en series de Volterra se resume en el Alg. 2, y consiste en únicamente dos pasos por iteración t :

1. Poda la estimación del vector de *kernels* de Volterra, reteniendo solamente las k componentes con el mayor valor absoluto, que corresponde al paso 4 del algoritmo. Hay que comentar que $\text{supp}(\cdot)$ devuelve los índices de los elementos no nulos y $H_k(\cdot)$ devuelve las k componentes del argumento con mayor valor absoluto.
2. Obtiene una nueva estimación aplicando la Ec. (4.8) para dichos elementos, que corresponde al paso 5. El paso 6 únicamente pone a cero aquellos elementos que no pertenecen al set de soporte del modelo.

Como se puede apreciar, además de la matriz de regresores y de la señal de salida medida, el algoritmo recibe otros dos argumentos de entrada: k , que es el número de componentes del modelo que se desea que estén activas, y λ , que es el parámetro de la regularización, cuyo valor óptimo se puede encontrar mediante la técnica de validación cruzada.

Algoritmo 2 Resumen del Algoritmo de Regresión de Ridge Iterativa (IRR) para Series de Volterra

Entrada: \mathbf{X} , y , λ , k

Salida: S , $\hat{\mathbf{h}}$

- 1: *Inicialización:* $\hat{\mathbf{h}}^{(0)} = (\mathbf{X}^H \mathbf{X} + \lambda)^{-1} \mathbf{X}^H y$
 - 2: $S^{(0)} \leftarrow \text{supp}(\hat{\mathbf{h}}_0)$
 - 3: **for** $t = 1$ hasta que el criterio de parada se cumpla **do**
 - 4: $S^{(t)} \leftarrow \text{supp}(H_k(\hat{\mathbf{h}}_{t-1}))$
 - 5: $\hat{\mathbf{h}}_{S^{(t)}}^{(t)} \leftarrow (\mathbf{X}_{S^{(t)}}^H \mathbf{X}_{S^{(t)}} + \lambda \text{diag}(|\hat{\mathbf{h}}_{S^{(t)}}^{(t-1)}|)^{-1})^{-1} \mathbf{X}_{S^{(t)}}^H y$
 - 6: $\hat{\mathbf{h}}_{S^{(t)}}^{(t)} \leftarrow 0$
 - 7: **end for**
-

4.2.3 Análisis de Componentes Principales

También conocida como PCA (del inglés *Principal Components Analysis*) Esta técnica tiene un enfoque totalmente distinto a las dos anteriores. Se trata de un método estadístico usado para transformar los datos de entrada, en este caso la matriz \mathbf{X} , que posiblemente tenga una correlación alta, en un set de datos incorrelados, ordenados de forma que la varianza vaya decreciendo a medida que se avanza en ellos. De esta forma, casi toda la varianza presente en los nuevos datos está en las primeras componentes, llamadas componentes principales, y las demás se pueden despreciar. Aquí reside la clave del uso de esta técnica, ya que sirve para eliminar redundancias y reducir por tanto el orden del modelo.

Esta idea se entiende mejor observando la Figura 4.1 basada en una de [22]. En ella se aprecia que con los datos de entrada proyectados sobre los vectores u_1 y u_2 se obtienen las varianzas σ_1 y σ_2 , siendo ambas bastante altas. Sin embargo, si se realiza un cambio de base, pasando de la base u a la base ortogonal e la varianza se concentra mucho más en $\hat{\sigma}_2$ (componente principal), mientras que $\hat{\sigma}_2$ resulta una varianza muy pequeña y se puede despreciar.

Tal y como se expone en [23], la técnica PCA se basa en la ecuación

$$\tilde{\mathbf{X}}_{PCA} = \mathbf{XV} \quad (4.9)$$

donde \mathbf{V} es una matriz cuadrada que se obtiene encontrando los autovectores de la matriz de correlación $\mathbf{X}^H \mathbf{X}$. De esta forma, la matriz se proyecta en un espacio vectorial ortogonal, resultando la matriz $\tilde{\mathbf{X}}_{PCA}$. Dicha matriz se puede usar directamente en la Ec. (4.4) para obtener una estimación de los coeficientes del modelo.

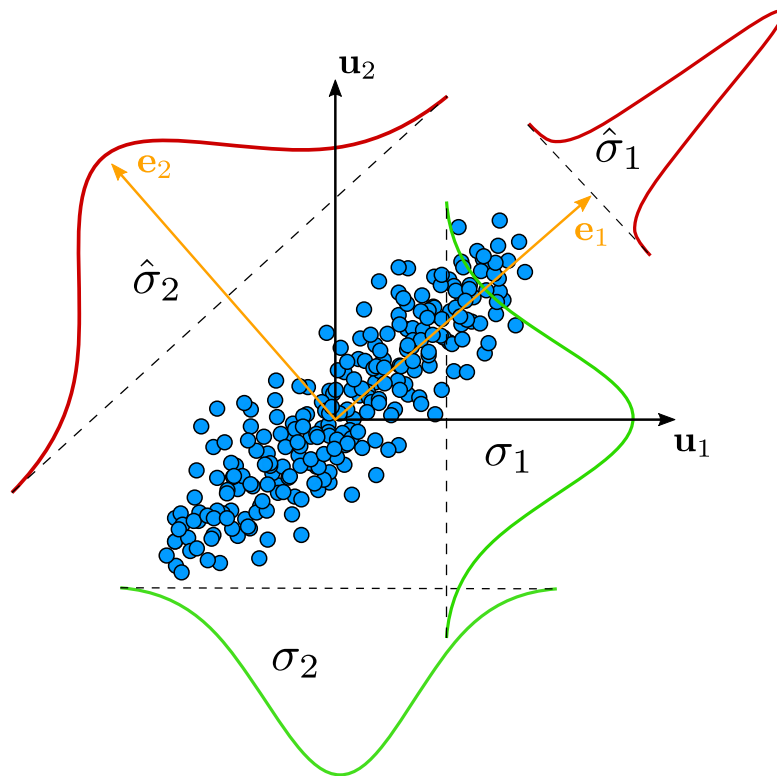


Figura 4.1 Representación gráfica de una transformación de bases que permite concentrar la mayoría de la varianza de los datos en una variable, pudiéndose despreciar la otra.

5 Resultados

Todo el mundo muere. Todo el mundo se marcha. Lo que importa son las cosas que construimos juntos antes de que lo hagan. Lo que importa es la parte de ellos que continúa en ti cuando no están.

"SAGA DE ENDER: HIJOS DE LA MENTE", DE ORSON SCOTT CARD

La intención de este trabajo es realizar una comparación de las técnicas de selección de coeficientes expuestas en el capítulo anterior en la Sección 4.2, en concreto, del algoritmo OMP, del IRR y de la técnica PCA. Para ello, se ha realizado un montaje experimental básico en el que se han medido varias señales de salida de un amplificador de potencia, a partir de una misma señal de entrada diseñada con el programa matemático MATLAB[®]. A estas mediciones se les han aplicado distintas técnicas de postprocesado y a continuación se han usado para realizar un modelo *Full-Volterra* y para poner así a prueba a las tres técnicas mencionadas. Los resultados obtenidos serán comentados y mostrados en diversas figuras.



Figura 5.1 Setup experimental en el cual se han realizado las medidas para este proyecto.

En primer lugar, es importante explicar qué montaje experimental se ha hecho. Para ello, se adjunta la Figura 5.1. En ella se puede observar cada uno de los dispositivos que se han usado. Siguiendo la cascada

lógica, en primer lugar se ha usado un modulador I/Q comercial de Rohde & Schwarz SMU200A, esto es, un generador vectorial de señal, seguido de un preamplificador ZHL42W de MiniCircuits. A continuación, se encuentra la placa de evaluación del amplificador de potencia CGH40010 GaN HEMT de Cree. Y finalmente, además de un acoplador direccional y un atenuador, se ha usado un analizador vectorial de señal PXA-N9030A de Agilent Technologies.

La señal de entrada que se ha empleado en el sistema anterior ha sido generada en base al estándar del enlace descendente de la cuarta generación móvil (LTE, del inglés *Long-Term Evolution*), en concreto, una señal OFDM de 15 MHz. La frecuencia a la que se ha trabajado ha sido de 3.6 GHz, frecuencia de operación de la placa de evaluación. La potencia media de entrada del pre-amplificador era de -18 dBm, mostrando una relación de pico a valor medio (PAPR, del inglés *Peak-to-Average Power Ratio*) de 11 dB. Por tanto, esta señal es introducida en el sistema y se realizan un total de tres medidas de la señal de salida del sistema. Posteriormente, los pasos típicos de la creación de un modelo de comportamiento de un amplificador de potencia han sido seguidos. Estos pasos son mostrados en la Figura 6.14 del libro de John Wood [8]. En primer lugar, se sincronizó cada señal de salida con la señal de entrada en dos pasos ya que es clave que para comparar las señales y habiendo visto lo importante que son los efectos de memoria, la sincronización debe ser perfecta. Con la Figura 5.2 se observa que se logra dicho propósito.

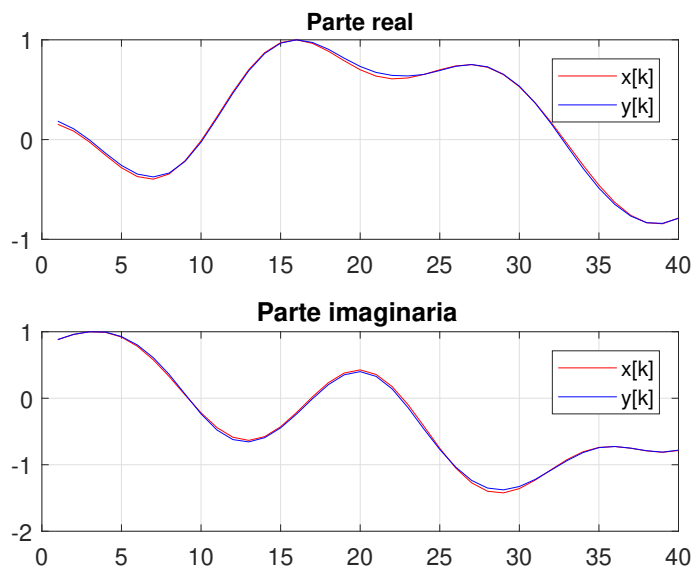


Figura 5.2 Representación de la parte real e imaginaria de las primeras muestras normalizadas de la señal de entrada y de salida, observando una sincronización perfecta.

Una vez se hace la sincronización de las tres señales de salida medidas con la señal de entrada, se realiza un promedio de las tres, con el objetivo de disminuir el ruido de cada muestra. Se obtiene por tanto un vector formado por las muestras de la señal de salida postprocesada y otro con la señal de entrada, ambos de 368640 muestras. Es importante mencionar que la construcción del modelo se divide en dos etapas. La primera es la identificación, en la que se usa una señal de entrada y su correspondiente señal de salida para hallar una estimación de los coeficientes \mathbf{h} del modelo, y la diferencia de la señal de salida modelada y la señal que se ha usado para la identificación permite hallar el NMSE de identificación. La segunda fase es la validación del modelo, en la que se halla la señal de salida modelada cuando la entrada es otra señal, por lo que se pone a prueba cuán buena es la estimación de los coeficientes anteriormente hecha. El error de la señal modelada y la que se ha medido en el laboratorio otorga el NMSE de validación, o de modelado, y es el más importante, ya que muestra la extrapolación que hace el modelo cuando la señal de entrada cambia, conservando la potencia.

Así pues, se ha utilizado como señal de identificación el 20% de la señal donde se encuentra el mayor pico de potencia instantánea, esto es, un total de 73728 muestras, para construir la matriz de regresores. Se ha usado el modelo FV con una profundidad de memoria $Q = 3$ y el orden máximo $2P - 1 = 13$; sin embargo, para órdenes superiores a 5 solo se han considerado las componentes sin memoria, o componentes estáticas, porque las componentes con memoria a órdenes tan altos son prácticamente despreciables y considerarlas todas implicaría una carga computacional desproporcionada. De esta forma, el número de regresores tratados ha sido de 248, por lo que la matriz de regresores tiene las dimensiones 368640×248 . El código de la función con la que ha sido generada se muestra en el Código 5.1 de MATLAB[®], que recibe de entrada \mathbf{x} , el vector

columna formado por las muestras de la señal de entrada; q , que es un array de celdas donde cada una indica cada combinación posible de retrasos de las señales que forman cada regresor; y c , que es otro array de celdas con las que se indica mediante "banderas" las señales de cada regresor que deben ir conjugadas. Los arrays q y c son contruidos con la función fv desarrollada por el Profesor Titular de la Universidad de Sevilla Luis Javier Reina Tosina. Se adjunta en el Apéndice A.

Código 5.1 Función generatriz de la matriz de regresores de un modelo Full-Volterra.

```
function [Xmat, Rmat] = buildX(x, q, c)
% Crea la matriz de regresores de un modelo Full Volterra
% Inputs:
% - x: vector columna con las muestras de la señal de entrada
% - q: array de celdas con las combinaciones de los retrasos. Salida de la
funcion fv
% - c: array de celdas con indicadores de los elementos conjugados. Salida de
la funcion fv
% Output:
% - Xmat: matriz de regresores del modelo
% - Rmat: array de celdas con las cadenas de texto que representan cada
regresor
% Realizada por Abraham Pérez Hernández. 2018.
Xmat = zeros(length(x),length(q));
Rmat = cell(length(q),1);
for i = 1:length(q) %Para cada combinacion de retrasos
    qi = q{i};
    ci = c{i};
    Xi = ones(length(x),1);
    Ri = '';
    for j = 1:length(qi) %Para cada retraso
        xj = circshift(x, -qi(j)); %La x retrasada
        if ci(j) %Si debe ser conjugada
            xj = conj(xj);
            if qi(j)~=0
                reg = sprintf('conj(x(k-%d))',qi(j));
            else
                reg = sprintf('conj(x(k))');
            end
        else
            if qi(j)~=0
                reg = sprintf('x(k-%d)',qi(j));
            else
                reg = sprintf('x(k)');
            end
        end
        Xi = Xi.*xj;
        if isempty(Ri)
            Ri = reg;
        else
            Ri = strcat(Ri,'*',reg);
        end
    end
    Xmat(:,i) = Xi;
    Rmat{i} = Ri;
end
end
```

Posteriormente, una vez montada la matrix \mathbf{X} y teniendo el vector \mathbf{y} , se han puesto a prueba los tres algoritmos para hallar la estimación de los coeficientes del modelo seleccionados por cada uno, tal cual se muestra en el Código 5.2, acompañándose de las funciones definidas en el Apéndice A. De esta forma, se puede comparar el error de modelado que cada algoritmo ofrece en función del número de componentes usadas, como se ve en la Figura 5.3. Además, se ha tomado como ejemplo las 10 primeras componentes elegidas por cada técnica y se ha hallado el espectro del error, esto es, la diferencia entre la señal modelada por estas componentes y la señal de salida medida en el laboratorio. Estos espectros están representados en la Figura 5.4. Además, se adjunta en la Tabla 5.1 los tiempos de ejecución de los algoritmos con un ordenador con procesador Intel Xeon a 3.6 GHz y 64GB de memoria RAM. Hay que puntualizar que el algoritmo PCA no es iterativo, por lo que el tiempo que se expresa es el tiempo total de ejecución dividido entre el número total de componentes procesadas.

Tabla 5.1 Duración de la realización de las iteraciones de las técnicas.

Algoritmo	Tiempo de ejecución medio por iteración
OMP	1.647 s
IRR	0.672 s
PCA	2.025 s

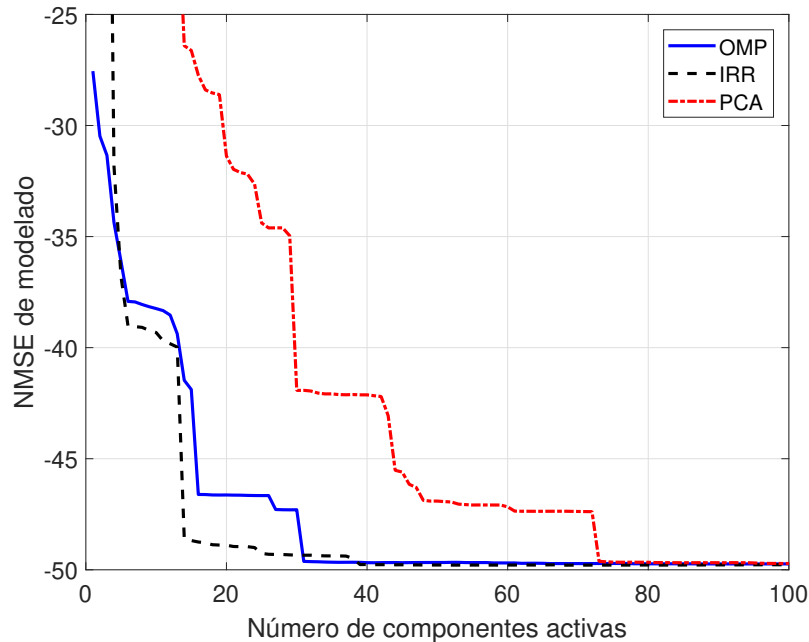


Figura 5.3 Evolución del NMSE de modelado de cada técnica en función del número de componentes seleccionadas.

Se puede apreciar cómo el NMSE de validación del modelo (Figura 5.3) de las técnicas OMP e IRR tienen una característica muy descendente en las primeras selecciones de coeficientes, mientras que el correspondiente a la técnica PCA es mucho más lenta. Se observa cómo con 14 componentes activas, el algoritmo IRR tiene un error de modelado de -48.6 dB y a partir de ahí evoluciona cuasi-constante, mientras que el OMP necesita 31 componentes para presentar un NMSE menor a -48 dB. El caso del PCA es mucho más grave, ya que el error que presenta no llega a este valor hasta que no ha activado las 73 primeras componentes que ha ordenado. En la Figura 5.4 se observa cómo el error del PCA es mucho más notable que el error del OMP o el que presenta el IRR, todos únicamente con 10 componentes activas. En cuanto al tiempo de ejecución mostrado en la Tabla 5.1 se puede observar cómo las iteraciones del IRR son 2.45 veces más rápidas que las del OMP y 3 veces más rápidas que las del PCA. Sin embargo, la técnica IRR presenta un mejor comportamiento cuantas más iteraciones se hagan requiriendo el mismo número de coeficientes. En este trabajo por cada componente se han realizado 10 iteraciones del IRR, resultado un total de 2480.

Se puede concluir entonces que el algoritmo PCA presenta un comportamiento mucho menos beneficioso

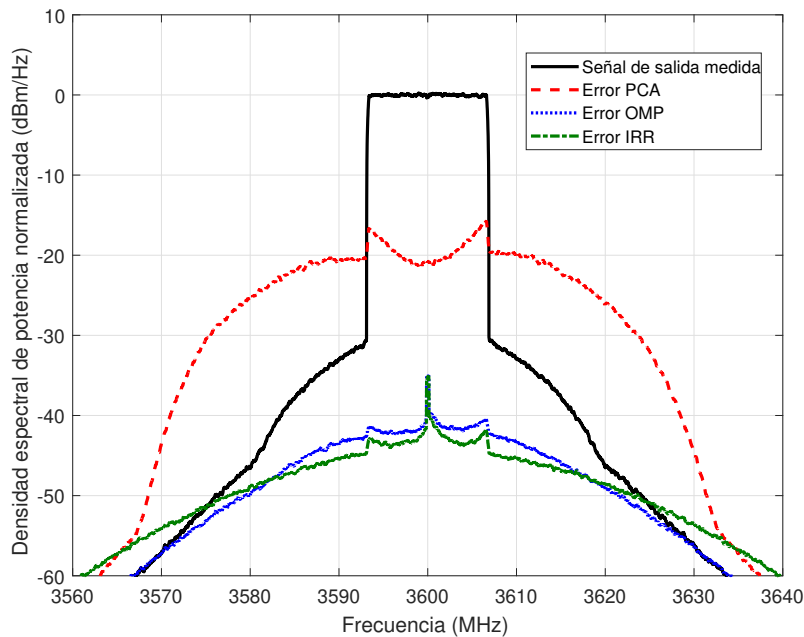


Figura 5.4 Densidad espectral de potencia de la señal modelada y del error de los modelos obtenidos con las técnicas PCA, OMP e IRR.

que las otras dos técnicas evaluadas. A su vez, el algoritmo OMP es bastante bueno y útil si se desea realizar un barrido del número de coeficientes seleccionados, pero cuando dicho número es fijado por decisión previa o por restricciones de la implementación, la técnica del IRR presenta un comportamiento mucho más rápido consiguiendo un modelado bastante efectivo.

Código 5.2 Código de la simulación que realiza la identificación del modelo y la correspondiente validación con cada una de las técnicas analizadas.

```
close all; clear; clc;
cd('confset'); configuration; cd ..;
addpath('toolbox');
load('measurements\y_lte15MHz_ovs6_CreeZHL_18_ppc');

t_ini = clock;
x = xsw;
y = ymed;

indid = sel_indices(x,y,0.1);
xid = x(indid);
yid = y(indid);
xval = x;
yval = y;

%% IRR

ref_model.calculo = 'IRR';
timer = tic;
[modelo_out] = model_fv_omp_itRidge_pca(yid, xid, ref_model);
t_irr = toc(timer);
for incoef=1:length(modelo_out.h_full)
    modelo_out.h = modelo_out.h_full(:,incoef);
    [modelo_out] = model_fv_omp_itRidge_pca(yval, xval, modelo_out);
    nmse_irr(incoef) = modelo_out.nmse;
```

```

    if incoef==10
        ymod_irr_10 = modelo_out.ymod;
        nmse_irr_10 = modelo_out.nmse;
        s_irr_10 = modelo_out.s;
    end
end

modelo_out_irr = modelo_out;

%% OMP
cd('confset'); configuration; cd ..;
addpath('toolbox');
load('measurements\y_lte15MHz_ovs6_CreeZHL_18_ppc');

x = xsw;
y = ymed;

indid = sel_indices(x,y,0.1);
xid = x(indid);
yid = y(indid);
xval = x;
yval = y;

ref_model.calculo = 'OMP';
ref_model.mode = 1; % 1 para OMP clásico, 2 para DOMP.
timer = tic;
[modelo_out] = model_fv_omp_itRidge_pca(yid, xid, ref_model);
t_omp = toc(timer);
[~, num_h] = size(modelo_out.h_full);
for incoef=1:num_h
    modelo_out.h = modelo_out.h_full(:,incoef);
    [modelo_out] = model_fv_omp_itRidge_pca(yval, xval, modelo_out);
    nmse_omp(incoef) = modelo_out.nmse;
    if incoef==10
        ymod_omp_10 = modelo_out.ymod;
        nmse_omp_10 = modelo_out.nmse;
        s_omp_10 = modelo_out.s;
    end
end

modelo_out_omp = modelo_out;

%% PCA
cd('confset'); configuration; cd ..;
addpath('toolbox');
load('measurements\y_lte15MHz_ovs6_CreeZHL_18_ppc');

x = xsw;
y = ymed;

indid = sel_indices(x,y,0.1);
xid = x(indid);
yid = y(indid);
xval = x;
yval = y;

```

```
ref_model.calculo = 'PCA';
timer = tic;
[modelo_out] = model_fv_omp_itRidge_pca(yid, xid, ref_model);
t_pca = toc(timer);
[~, num_h] = size(modelo_out.h_full);
for incoef=1:num_h
    modelo_out.h = modelo_out.h_full(:,incoef);
    [modelo_out] = model_fv_omp_itRidge_pca(yval, xval, modelo_out);
    nmse_pca(incoef) = modelo_out.nmse;
    if incoef==10
        ymod_pca_10 = modelo_out.ymod;
        nmse_pca_10 = modelo_out.nmse;
        s_pca_10 = modelo_out.s;
    end
end
end

modelo_out_pca = modelo_out;

%% END
t_end = clock;
t_exe = datevec(datenum([0 0 0 0 0 etime(t_end,t_ini)]));

disp(sprintf('Tiempo total de ejecución: %d días, %d horas, %d minutos y
%4.2f segundos\n',t_exe(3),t_exe(4),t_exe(5),t_exe(6)))
```


6 Conclusiones y líneas futuras de trabajo

De oídas te había oído; mas ahora mis ojos te ven.

Job 42, 5

Este trabajo ha servido para comparar el algoritmo IRR, recientemente presentado en [21], con otras dos técnicas de selección de componentes de modelos de comportamientos de amplificadores de potencia basados en series de Volterra como son el algoritmo OMP y la técnica PCA. Como resultado se ha visto que se trata de un algoritmo iterativo muy efectivo en términos de error de modelado cuando se sabe el número de componentes activas que se quiere mantener, por lo que una forma de actuar bastante buena sería realizar un barrido del número de coeficientes mediante el algoritmo OMP y una vez se decida mediante técnicas como el criterio BIC, del inglés *Bayesian Information Criterion* [18], o la técnica de promediado y umbralización basada en el BIC [13], se aplicaría el algoritmo IRR para obtener un error de modelado más bajo que con los coeficientes obtenidos por el algoritmo OMP.

Los futuros trabajos que se podrían hacer en seguimiento de este proyecto serían dar el siguiente paso en la linealización de amplificadores de potencia: predistorsión digital (DPD). Una posibilidad sería realizar una comparativa de cómo se desenvolvería el algoritmo IRR en la etapa de predistorsión con respecto al OMP, o a otros posibles algoritmos como DOMP [20], CoSaMP [19], o *Subspace Pursuit* (SP) [24]. Además, en vez de realizar únicamente las medidas en el laboratorio, sería mucho más realista realizar la etapa de predistorsión también en el laboratorio, usando una FPGA por ejemplo como predistorsionador, o la plataforma WebLab de la Universidad de Chalmers para ello [25].

Como un trabajo futuro a más largo plazo, se podrían aplicar las tecnologías de redes neuronales artificiales y aprendizaje profundo que se están desarrollando en la actualidad [26], con las que se preveen muy buenos resultados.

Apéndice A

Códigos de MATLAB®

A.1 Función generatriz de parámetros del modelo FV

Código A.1 Función que genera las distintas combinaciones de retrasos y vectores binarios que indican los elementos conjugados del modelo *Full-Volterra*, desarrollada por el profesor Luis Javier Reina Tosina.

```
function y = fv(varargin)
    % Full Volterra model library of regressors
    % Inputs:
    % - p: model order (p<=13). Default P = 5
    % - q: delay length (vector). Default Q = 5
    % Output:
    % - y.q: cell array with set of delay tap vectors
    % - y.c: cell array with set of binary vectors (conj vs nonconj elements)
    % - y.nr: number of regressors
    % Version: 2014.02.19

    if nargin == 0
        p = []; q = [];
    elseif nargin == 1
        p = varargin{1}; q = [];
    elseif varargin{1} == 'p'
        p = varargin{2}; q = [];
    elseif varargin{1} == 'q'
        p = []; q = varargin{2};
    else
        p = varargin{1};
        q = varargin{2};
    end

    if isempty(p)
        p = 5;
    end

    if isempty(q)
        q = 5;
    end

    if p > 13
        disp('Feature not implemented: model order > 13');
        return;
    end
```

```
end

Q = q(:);
if length(Q) > p
    disp('Unmatched delay tap vector');
    return;
elseif length(Q) == 1
    Q = repmat(Q, (p+1)/2, 1);
end

y.nr = 0;
ind = 0;

for P = 1:2:p
    switch P

        case 0
            ind = ind + 1;
            y.q{ind} = 'dc';
            y.c{ind} = 0;

        case 1
            for q1 = 0:Q(1)
                ind = ind + 1;
                y.q{ind} = q1;
                y.c{ind} = 0;
            end

        case 3
            for q1 = 0:Q(2)
                for q2 = q1:Q(2)
                    for q3 = 0:Q(2)
                        ind = ind + 1;
                        y.q{ind} = [q1, q2, q3];
                        y.c{ind} = [0, 0, 1];
                    end
                end
            end

        case 5
            for q1 = 0:Q(3)
                for q2 = q1:Q(3)
                    for q3 = q2:Q(3)
                        for q4 = 0:Q(3)
                            for q5 = q4:Q(3)
                                ind = ind + 1;
                                y.q{ind} = [q1, q2, q3, q4, q5];
                                y.c{ind} = [0, 0, 0, 1, 1];
                            end
                        end
                    end
                end
            end

        case 7
            ind = ind + 1;
            y.q{ind} = [0, 0, 0, 0, 0, 0, 0];
    end
end
```

```

        y.c{ind} = [0, 0, 0, 0, 1, 1, 1];

    case 9
        ind = ind + 1;
        y.q{ind} = [0, 0, 0, 0, 0, 0, 0, 0, 0];
        y.c{ind} = [0, 0, 0, 0, 0, 1, 1, 1, 1];

    case 11
        ind = ind + 1;
        y.q{ind} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        y.c{ind} = [0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1];

    case 13
        ind = ind + 1;
        y.q{ind} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
        y.c{ind} = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1];

    end
end
y.nr = ind;
end

```

A.2 Función de identificación y validación del modelo FV

Código A.2 Función que realiza la identificación o validación de un modelo *Full-Volterra*, según las distintas estrategias comparadas.

```

function [modelo] = model_fv_omp_itRidge_pca(ymed, xmed, modelo)
    %Desarrollada principalmente por Juan Antonio Becerra González, con
    modificaciones de Abraham Pérez Hernández, 2018.
    xmed = xmed(:)-mean(xmed);
    ymed = ymed(:)-mean(ymed);

    % Asignación de variables
    P = modelo.P; Q = modelo.Q;

    % Comprobación de seguridad en caso de que el modelo sea sin memoria
    if isempty(P)
        P = 5;
    end
    if isempty(Q)
        Q = 5;
    end
    N = length(ymed);

    % Cálculo de la señal de índices
    if modelo.extension_periodica
        n = [N-Qpmax+1:N , 1:N , 1:Qnmax];
    else
        n = 1:N;
    end
    x = xmed;
    y = ymed;

```

```

if isfield(modelo,'s')
    s = modelo.s;
else
    s = [];
end

% Construcción de la matriz de predictores.
md_reg = fv(P,Q);

if isempty(modelo.h)
    [w_mat, Rmat] = buildX(x,md_reg.q, md_reg.c);
else
    X_mat = buildX(x,md_reg.q, md_reg.c);
    ymod = X_mat*modelo.h;
end

% Se calculará el modelo si la variable h viene vacía.
if isempty(modelo.h)
    switch(modelo.calculo)
        case 'pinv'
            % Calculo con la pseudo inversa
            h_vec = pinv(w_mat)*y;
            ymod = w_mat* h_vec;
        case 'OMP'
            mode = modelo.mode;
            nmax = modelo.nmax;
            [h_vec, ve2, s, nopt, t_exec, h_full, nmse_v] = omp(w_mat, y,
                nmax,s,Rmat,mode);
            ymod = w_mat* h_vec;
        case 'PCA'
            [h_vec, h_full, nmse_v] = pcafit(w_mat, y);
            ymod = w_mat* h_vec;
        case 'IRR'
            [h_vec, ve2, s, h_full, nmse_v] = iteratedRidge(w_mat, y);
            ymod = w_mat* h_vec;
    end
end

nmse = 20*log10(norm(ymod-y)/norm(y));
nmse_n = 20*log10(norm(ymod/norm(ymod)-y/norm(y))/norm(y/norm(y)));

%% Guardado de parámetros de salida
if exist('ML','var')
    modelo.ML = ML;
end
if exist('nopt','var')
    modelo.nopt = nopt;
end
if exist('span','var')
    modelo.span = span;
end
if exist('t_exec','var')
    modelo.t_exec = t_exec;
end
if exist('ve','var')

```

```

    modelo.ve2 = ve2;
end
if exist('Rmat','var')
    modelo.Rmat = Rmat;
end
if exist('s','var')
    modelo.s = s;
end
if exist('nmse_v','var')
    modelo.nmse_v = nmse_v;
end
if exist('h_vec','var')
    modelo.h = h_vec;
    modelo.numcoef = length(h_vec);
end
if exist('h_full','var')
    modelo.h_full = h_full;
end

modelo.ymod = ymod;
modelo.nmse = nmse;
modelo.nmse_n = nmse_n;
modelo.fecha = date;
if exist('w_mat','var')
    modelo.X = w_mat;
    modelo.y_id = y;
end
end
end

```

A.3 Algoritmos y funciones auxiliares de las técnicas OMP, IRR y PCA

Código A.3 Algoritmo *Orthogonal Matching Pursuit*.

```

function [h, ve2, s, nopt, t_iter, h_full, nmse] = omp(X, y, nmax, s, Rmat,
mode)
[~,N] = size(X);
for in=1:N
    normA(in) = norm(X(:,in));
    Xn(:,in)=X(:,in)/normA(in);
end
Zn = Xn;

if isempty(s)
    h = zeros(N, 1);
    ve2 = zeros(1, min(nmax,N));
    r = y;
else
    for n = 1:length(s)
        h(1:n,n) = pinv(X(:,s(1:n))))*y;

        yLS = X(:,s(1:n))*h(1:n,n);
        r(:,n+1) = y-yLS;
        ve2(n) = var(y-yLS);
        nmse(n) = 20*log10((norm(yLS-y)./norm(y)));
    end
end

```

```

end
end

for n = length(s)+1:min(nmax,N)
    tic
    fprintf('%d ', n);
    % Sorting correlation
    if mode == 1
        Cx = abs(Xn'*(r(:,n)));
    else
        Cx = abs(Zn'*(r(:,n)));
    end

    % We set the correlation of the already selected coefficients to 0,
    % avoiding a new selection.
    Cx(s) = 0;
    [Cxsort ind] = sort(Cx,'descend');

    s(n) = ind(1);

    %figure(9);clf;stem(Cxsort(1:50));hold on; stem(ind(ind2),Cxsort(ind(
        ind2)), 'r')

    if (mode==1||mode==2)
        h(sort(s),n) = diag(normA(sort(s)).^-1)*pinv(Xn(:,sort(s)))*y;
    else
        if length(s)==1
            h(s,n) = pinv(X(:,s))*y;
            Cant = inv(X(:,s)'*X(:,s));
        else
            Cant = pinv_m(X(:,s(1:end-1)),Cant,X(:,s(end)))
            h(s,n) = Cant*X(:,s)';
        end
    end
    end
    yLS = X(:,s)*h(s,n);
    r(:,n+1) = y-yLS;
    ve2(n) = var(y-yLS);
    nmse(n)=20*log10(norm(yLS-y,2)/norm(y,2));
    disp(sprintf('%d/%d - %s - nmse: %4.2f - correlation value: %0.2g',n,
        nmax,Rmat{s(n)},nmse(n),Cxsort(1)));
    t_iter(n) = toc;
end

M = length(y);
BIC = 2*M*log(ve2)+2*(1:min(nmax,N))*log(2*M); % complex
[~, nopt] = min(BIC);

h_full=h;

end

```

Código A.4 Algoritmo *Iterated Ridge Regression* y funciones auxiliares.

```

function [h, ve2, s, h_full, nmse]=iteratedRidge(X,y)
    [M, N] = size(X);
    normA = zeros(N,1);

```

```

for in=1:N
    normA(in) = norm(X(:,in));
    Xn(:,in)=X(:,in)/normA(in);
end

lambda = cross_validation(Xn,y);

kvec = 1:N;
h_out = zeros(N, length(kvec));

for ik = 1:length(kvec)
    k = kvec(ik);
    % addk = min(2*k,N);
    addk = k;

    % Initilize at Ridge Regression Solution
    s = 1:N;

    h = (Xn'*Xn + lambda*eye(N))\ (Xn'*y);

    yest = Xn*h;
    rant = 0;
    rnorm = norm(y-yest);
    ynorm = norm(y);
    nmse_ridge = 20*log10(rnorm/ynorm);

    for n = 1:10
        tic
        %threshold = 10^-4;
        %[s,~] = find(abs(h) > threshold);
        s = Hk(h,k);

        h(s) = (Xn(:,s)'*Xn(:,s) + diag(lambda*abs(1./h(s))))\ (Xn(:,s)'*y);
        h(setxor(s,1:N))=0;

        yest = Xn(:,s)*h(s);

        rant = rnorm;
        rnorm = norm(y-yest);
        nmse(n) = 20*log10(rnorm/ynorm);
        ncoef(n)=length(s);
        incr(n) = rant-rnorm;

        t_iter(n,ik) = toc;
    end
    %figure;plot(t_iter); title('Iteration time');
    h_out(:,ik) = diag(normA.^-1)*h;

    nmse_k(ik) = nmse(end);
    ve2(ik) = var(y-yest);
end

figure;plot(t_iter); title('Iteration time');
figure;clf;plot(kvec,nmse_k); title('IRR Identification NMSE')

t_iter = 0;

```

```

    h_full=h_out;
    h = h_out(:,end);

end

function lambda = cross_validation(Xn,y)
    lambda_v = 10.^[-20:-6, -5.5:0.5:-2];

    Nfold = 10;
    [M, N] = size(Xn);

    Lfold = floor(length(y)/Nfold);

    Xn = Xn(1:Lfold*Nfold,:);
    y = y(1:Lfold*Nfold,:);

    fold_ind = kron(1:Nfold,ones(1,Lfold))';
    for ilambda = 1:length(lambda_v)
        for ifold=1:Nfold
            [indregr,~] = find(fold_ind~=ifold);
            [indval,~] = find(fold_ind==ifold);
            h = (Xn(indregr,:)'*Xn(indregr,:) + lambda_v(ilambda)*eye(N))\ (Xn(
                indregr,:)'*y(indregr));

            nmse(ifold,ilambda) = 20*log10(norm(y(indval)-Xn(indval,:)*h)/norm(y
                (indval)));
        end
    end

    figure
    plot(mean(nmse))

    [~,indlambda] = min(mean(nmse));
    lambda = lambda_v(indlambda);
end

function indout = Hk(A,k)
    k = min(k,length(A));

    [~,indsort]=sort(A,'descend');
    indout = indsort(1:k);
end

function indout = supp(A)
    [~,indout] = find(A==0);
end

```

Código A.5 Función que aplica la técnica de Análisis de Componentes Principales.

```

function [h, h_full, nmse] = pcafit(X, y)
    [~,N] = size(X);
    normA = zeros(N,1);
    for in=1:N
        normA(in) = norm(X(:,in));
        Xn(:,in)=X(:,in)/normA(in);
    end

```



```
end

kvec = 1:N;
h_full = zeros(N, length(kvec));

[PCALoadings,PCAScores,~] = pca(Xn,'Economy',false);

for in=1:N
    hpca = pinv(PCAScores(:,1:in))*y;
    yfitPCR = Xn*PCALoadings(:,1:in)*hpca;

    nmse(in) = 20*log10(norm(y - yfitPCR)/norm(y));

    h_full(:,in) = diag(normA.^-1)*PCALoadings(:,1:in)*hpca;

    yest = Xn(:,:)*h_full(:,in);
    ve2(in) = var(y-yest);
end

M = length(y);
BIC = 2*M*log(ve2)+2*(kvec)*log(2*M); % complex
[~, nopt] = min(BIC);

h = h_full(:,end);
end
```


Índice de Figuras

1.1	Esquema gráfico del concepto de la predistorsión para linealizar amplificadores de potencia	2
2.1	Balance de potencias de un amplificador de potencia, sirviendo como ejemplo de cualquier dispositivo activo	4
2.2	Ganancia estática real presentada por el amplificador ZHL42W de Minicircuits frente a una entrada de un tono a 850 MHz, obtenida por medidas experimentales en el laboratorio	5
2.3	Recrecimiento espectral observado en un sistema no lineal excitado por un estímulo modulado de banda estrecha	8
2.4	Características AM/AM y AM/PM de un amplificador de potencia, donde se aprecia la dispersión de los puntos debida a los efectos de memoria	10
3.1	Retrato de Vito Volterra, matemático y físico italiano cuyas investigaciones del análisis funcional son la base del desarrollo de las series matemáticas que llevan su nombre	11
4.1	Representación gráfica de una transformación de bases que permite concentrar la mayoría de la varianza de los datos en una variable, pudiéndose despreciar la otra	21
5.1	Setup experimental en el cual se han realizado las medidas para este proyecto	23
5.2	Representación de la parte real e imaginaria de las primeras muestras normalizadas de la señal de entrada y de salida, observando una sincronización perfecta	24
5.3	Evolución del NMSE de modelado de cada técnica en función del número de componentes seleccionadas	26
5.4	Densidad espectral de potencia de la señal modelada y del error de los modelos obtenidos con las técnicas PCA, OMP e IRR	27

Índice de Tablas

5.1 Duración de la realización de las iteraciones de las técnicas

26

Índice de Códigos

5.1	Función generatriz de la matriz de regresores de un modelo Full-Volterra	25
5.2	Código de la simulación que realiza la identificación del modelo y la correspondiente validación con cada una de las técnicas analizadas	27
A.1	Función que genera las distintas combinaciones de retrasos y vectores binarios que indican los elementos conjugados del modelo <i>Full-Volterra</i> , desarrollada por el profesor Luis Javier Reina Tosina	33
A.2	Función que realiza la identificación o validación de un modelo <i>Full-Volterra</i> , según las distintas estrategias comparadas	35
A.3	Algoritmo <i>Orthogonal Matching Pursuit</i>	37
A.4	Algoritmo <i>Iterated Ridge Regression</i> y funciones auxiliares	38
A.5	Función que aplica la técnica de Análisis de Componentes Principales	40

Bibliografía

- [1] I. E. Agency, “More Data, Less Energy. Making Network Standby More Efficient in Billions of Connected Devices,” International Energy Agency, Tech. Rep., 2014.
- [2] A. Katz, J. Wood, and D. Chokola, “The Evolution of PA Linearization: From Classic Feedforward and Feedback Through Analog and Digital Predistortion,” *IEEE Microwave Magazine*, vol. 17, no. 2, pp. 32–40, Feb 2016.
- [3] J. C. Pedro and N. B. Carvalho, *Intermodulation Distortion in Microwave and Wireless Circuits*. Artech House, 2003.
- [4] M. Allegue-Martínez, “Modeling and Compensation of Non-linear Effects in Wireless Communications Systems,” Ph.D. dissertation, University of Sevilla, 2012.
- [5] M. J. Madero-Ayora, “Analysis and Experimental Characterization of Nonlinear Circuits Applied to Wireless Communications Systems,” Ph.D. dissertation, University of Sevilla, 2008.
- [6] F. J. Payán-Somet, *Principios de Comunicaciones Digitales I : Fundamentos*. Editorial Universidad de Sevilla-Secretariado de Publicaciones, 2014.
- [7] V. Volterra, *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Publications, 1959.
- [8] J. C. Wood, *Behavioral Modeling and Linearization of RF Power Amplifiers*. Artech House Publishers, 2014.
- [9] S. S. Soliman and M. D. Srinath, *Señales y Sistemas continuos y discretos*. Prentice Hall Iberia, 1999.
- [10] A. Pérez-Hernández, J. A. Becerra, M. J. Madero-Ayora, and C. Crespo-Cadenas, “A Comprehensive Performance Benchmark of Component Selection Techniques for Volterra Behavioral Models,” in *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, 2018.
- [11] S. Benedetto, E. Biglieri, and R. Daffara, “Modeling and Performance Evaluation of Nonlinear Satellite Links-A Volterra Series Approach,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-15, no. 4, pp. 494–507, jul 1979.
- [12] D. R. Morgan, Z. Ma, J. Kim, M. G. 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.
- [13] C. Crespo-Cadenas, M. J. Madero-Ayora, J. Reina-Tosina, and J. A. Becerra-González, “Formal deduction of a Volterra series model for complex-valued systems,” *Signal Processing*, vol. 131, pp. 245–248, 2017.
- [14] Scharf, *Statistical Signal Processing*. Pearson India, 2010.
- [15] J. Kim and K. Konstantinou, “Digital predistortion of wideband signals based on power amplifier model with memory,” *Electronics Letters*, vol. 37, no. 23, p. 1417, 2001.

- [16] V. K. Rohatgi, A. Ben-Israel, and T. N. E. Greville, "Generalized Inverses: Theory and Applications," *International Statistical Review / Revue Internationale de Statistique*, vol. 44, no. 2, p. 301, aug 1976.
- [17] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996, Available: <http://www.jstor.org/stable/2346178>.
- [18] J. Reina-Tosina, M. Allegue-Martínez, C. Crespo-Cadenas, C. Yu, and S. Cruces, "Behavioral Modeling and Predistortion of Power Amplifiers Under Sparsity Hypothesis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 63, no. 2, pp. 745–753, Feb 2015.
- [19] A. Abdelhafiz, A. Kwan, O. Hammi, and F. M. Ghannouchi, "Digital Predistortion of LTE-A Power Amplifiers Using Compressed-Sampling-Based Unstructured Pruning of Volterra Series," *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 11, pp. 2583–2593, Nov 2014.
- [20] J. A. Becerra, M. J. Madero-Ayora, J. Reina-Tosina, C. Crespo-Cadenas, J. García-Frías, 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.
- [21] J. A. Becerra, A. Pérez-Hernández, M. J. Madero-Ayora, and C. Crespo-Cadenas, "Efficient linearization of a RF Transmitter under 5G waveforms through Iterated Ridge Regression," in *2018 91st ARFTG Microwave Measurement Conference (ARFTG)*, June 2018, pp. 1–4.
- [22] Q. A. Pham, D. López-Bueno, G. Montoro, and P. L. Gilabert, "Adaptive Principal Component Analysis for Online Reduced Order Parameter Extraction in PA Behavioral Modeling and DPD Linearization," in *2018 IEEE/MTT-S International Microwave Symposium - IMS*, June 2018, pp. 160–163.
- [23] P. L. Gilabert, G. Montoro, D. López, N. Bartzoudis, E. Bertran, M. Payaró, and A. Hourtane, "Order reduction of wideband digital predistorters using principal component analysis," in *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*, June 2013, pp. 1–7.
- [24] J. A. Becerra, M. J. Madero-Ayora, A. Pérez-Hernández, and C. Crespo-Cadenas, "Algoritmo Subspace Pursuit para la Selección de Parámetros de Modelos de Volterra," in *XXXIII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2018)*, 2018.
- [25] 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, Feb 2015.
- [26] T. Gotthans, G. Baudoin, and A. Mbaye, "Digital predistortion with advance/delay neural network and comparison with Volterra derived models," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*. IEEE, sep 2014.