Índice del Capítulo:

1. Introducción.....2
2.- Series de Volterra 3
3.- Método del aprendizaje indirecto.....5

1.- Introducción

Veremos el modelo de predistorsión adaptativa usando polinomios con memoria, propuesto por G. Tong Zhou [1]. Como es bien sabido hasta hace poco los sistemas de predistorsión propuestos y aplicados eran fundamentalmente sin memoria. Las codificaciones utilizadas y los anchos de banda con los que se trabajaba no exigían mayores prestaciones para estos sistemas. Con la aparición de la multiplexación CDMA y la codificación OFDM, los anchos de banda utilizados y las tensiones pico a pico manejadas obligan a tener en cuenta efectos de memoria inducidos en los HPA.

IEEE

1.1.- Series de Volterra

No podemos hablar de análisis de no linealidades en sistemas de comunicaciones inalámbricas sin hablar de las series de Volterra. La serie de Volterra es una expansión funcional. En sí misma es una herramienta muy utilizada para el análisis de circuitos no lineales, existiendo una gran cantidad de artículos dedicados a sus aplicaciones para el análisis de dichos circuitos. El matemático a que debe su nombre esta herramienta demostró que todo funcional *G*[*x*] continuo en el espacio de las funciones continuas puede ser representado por medio de la expansión:

$$G(x) = \sum_{n=0}^{\infty} F_n(x)$$

en la que $F_n(x)$ **es un funcional regular homogéneo de la forma:**

$$F_n(x) = \int_a^b \ldots \int_a^b h_n(\xi_1,\ldots,\xi_n)x(\xi_1)x(\xi_2)\ldots x(\xi_n)d\xi_1 d\xi_2 \ldots . d\xi_n$$

El primero en aplicar esta expansión funcional en series de Volterra al análisis de sistemas no lineales fue Wiener.

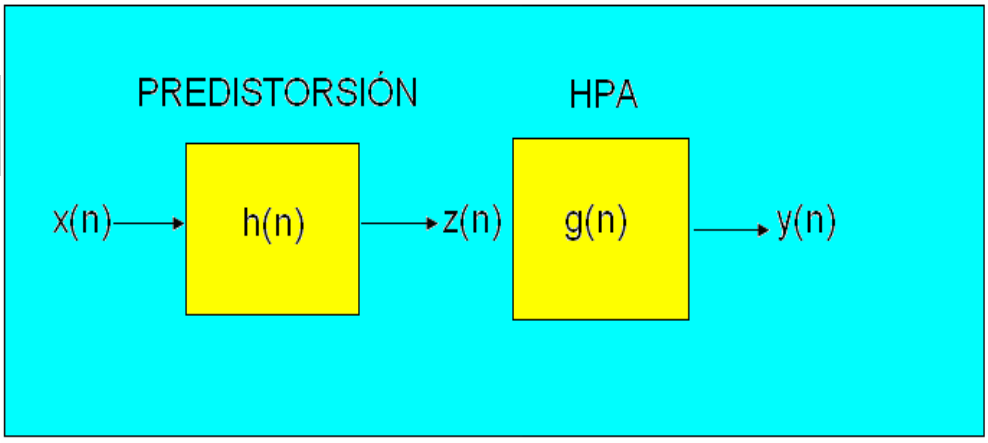
Si la salida *y*(*t*) de un sistema puede expresarse mediante un funcional de su entrada *x*(*t*), tal como predijo Volterra, entonces tanto x(t) como y(t) se pueden relacionar por medio de una serie de funcionales. Este método de análisis se emplea fundamentalmente en el dominio de la frecuencia utilizando una descripción adecuada de los dispositivos no lineales mediante modelos adecuados para los sistemas bajo

análisis. El análisis mediante series de Volterra se ha aplicado con éxito en el análisis de circuitos con elementos no lineales.

1.2.- Método del aprendizaje indirecto

Atendiendo a estos criterios ampliamente explicados y detallados en los estudios realizados por el Profesor Javier Reina Tosina de la Universidad de Sevilla, tutorado por el Dr. Carlos Crespo Cadenas, se puede pasar a explicar con más detalle la técnica propuesta por G. Tong Zhou [1].

Se parte de la idea de que tenemos el siguiente esquema:



Donde g(n) representa el modelo del HPA con el que vamos a trabajar y que modelaremos por una serie de Volterra de orden tres y h(n) representa el modelo del Predistorsionador, que evidentemente al ser el inverso de un sistema no lineal, también será un sistema no lineal y que también lo modelaremos como una serie de Volterra de orden tres.

$$z(n) = \sum_{q=0}^{Q} h_1(q)x(n-q) + \sum_{q_1,q_2,q_3=0}^{Q} h_3(q_1,q_2,q_3)x(n-q_1)x(n-q_2)x^*(n-q_3)$$

Podemos ver que los núcleos o Kernels

$\{h_1(q)\}_{q=0}^{Q}$ **y** $\{h_3\{q_1,q_2,q_3)\}_{q_1,q_2,q_3=0}^{Q}$ **representan al sistema no linear. Si**

Q=0, tenemos que:

$$z(n) = h_1(0)x(n) + h_3(0,0,0)x(n)\left|x(n)\right|^2$$

Que representa un sistema sin memoria

En el caso de que $h_3(q_1,q_2,q_3) = 0$ **, excepto en la diagonal** $q_1 = q_2 = q_3$

$$z(n) = \sum_{q=0}^{Q}[h_1(q)x(n-q) + h_3(q,q,q)x(n-q)\left|x(n-q)\right|^2]$$

Que representa por su lado un sistema con memoria. Se puede ver esta ecuación como un resultado intermedio entre la exactitud de la serie completa de Volterra para capturar todo los efectos de memoria que se pueden generar en un HPA y los resultados del modelo sin memoria anteriormente expuesto.

Resumiendo, con este resultado se reduce el orden del modelo del predistorsionador, haciendo que capture los efectos de las no linealidades del HPA, al mismo tiempo que reducimos considerablemente el orden de la serie.

Una vez realizado el modelo del Predistorsionador, el

objetivo es la adaptabilidad de este mismo a la variación del HPA, con una arquitectura de aprendizaje indirecto.

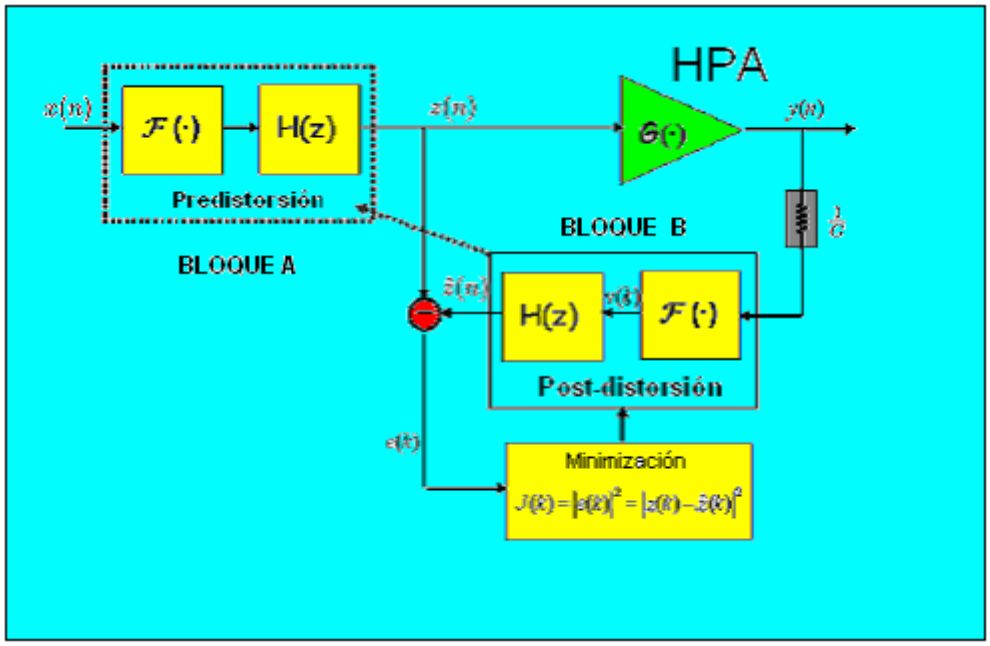


Fig Modelo adaptativo

Con

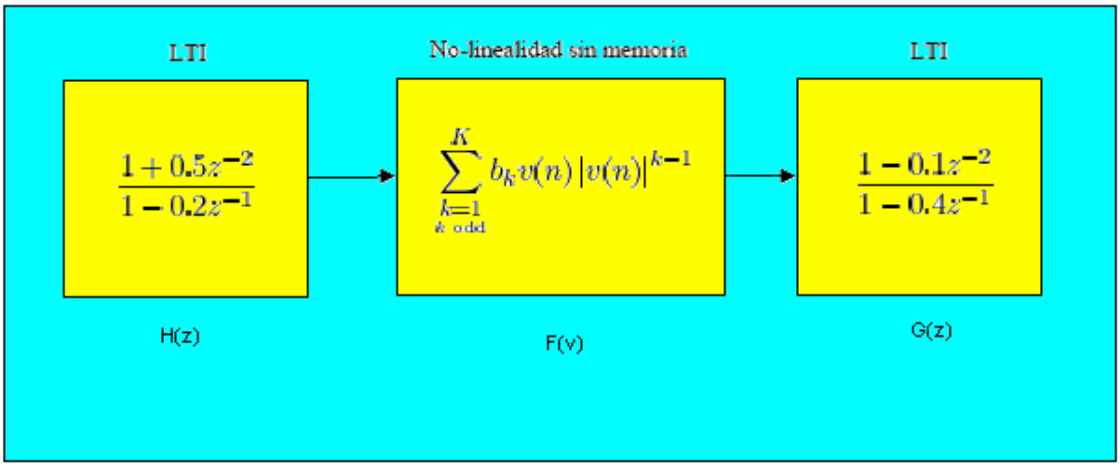


Fig. Modelo de Hammerstein-Wiener, para HPA

O con modelo directo de Volterra

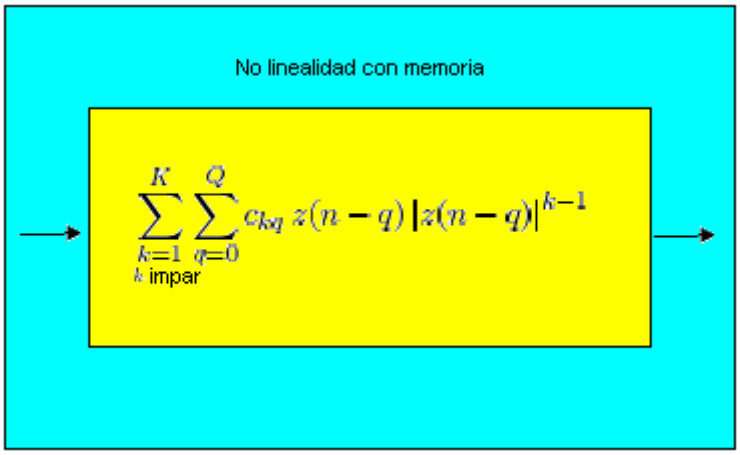


Fig. Modelo de Volterra, para HPA

Nosotros hemos analizado ambos modelos, con resultados muy similares

Para la resolución de estos modelos propuestos tenemos que asumir que el HPA es invertible, y por tanto e(n)=0 lo que implicaría que y(n)=Gx(n).Donde consideramos que las características del HPA varían lentamente con el tiempo (variación de constante con el tiempo, con la temperatura etc...) El objetivo por lo tanto es conseguir obtener los parámetros del bloque A, lo cual se puede hacer offline y después reengancharlo en la cadena de bloques.

Para la identificación de los parámetros recurrimos a los propuesto para esta técnica. Renombramos $h_k(q,q,q....q)$ **por** a_{kq} **con lo que tenemos:**

$$z(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} a_{kq}\, x(n-q) \left| x(n-q) \right|^{k-1}$$

Donde hemos generalizado el modelo del

predistorsionador y hemos reducido la variable K solo a términos impares con lo que K=2L -1. Esto se debe a que en la mayoría de los diseños de predistorsionadores se utilizan nada más que exponentes impares, que se ajustan mejor a las especificaciones.

Lo que nos queda pues es:

$$x(n-q)\left|x(n-q)\right|^{k-1} = [x(n-q)]^{l+1}[x^*(n-q)]^l$$

Si la respuesta fuera la deseada $x(n) = \frac{y(n)}{G}$ **, significaría que z(n) es linear, y por tanto, estimar los parámetros** a_{kq} **no sería más que aplicar el método de los mínimos cuadrados. Definiendo:**

$$u_{kq}(n) = \frac{y(n-q)}{G}\left|\frac{y(n-q)}{G}\right|^{k-1}$$

$$\vec{z} = \vec{U}\vec{a}$$

Donde $z = [z(0),\ldots..,z(N-1)]^T$

$\vec{U}$ **=** $[\vec{u}_{10},\ldots.,\vec{u}_{K0},\ldots..,\vec{u}_{1Q},\ldots...,\vec{u}_{KQ}]$ **con** $\vec{u}_{kq} = [u_{kq}(0),\ldots..,u_{kq}(N-1)]^T$ **y**

$\vec{a}$ **=** $[a_{10},\ldots.,a_{K0},\ldots...a_{1Q},\ldots..,a_{KQ}]^T$

Donde el resultado de los mínimos cuadrados es:

$$\hat{a} = (\vec{U}^H\vec{U})^{-1}\vec{U}\vec{z}$$

con $(.)^H$ entendemos el complejo conjugado transpuesto.

Por lo que ya hemos obtenido el valor de los coeficientes del predistorsionador de una manera sencilla. Estos son los coeficientes que ha de tener el Predistorsionador. En varias iteraciones conseguiremos que estos converjan.

Con esto ya hemos conseguido identificar de una manera relativamente sencilla los parámetros del DPD.

En el siguiente capítulo daremos una visión práctica de estos resultado a fin de poder evaluar de manera cuantitativa la eficiencia de los mismos.