

Ecuación difusa adaptativa basada en agrupamiento por neuronas para canales de comunicación no lineales y variantes en el tiempo¹

Adaptive Fuzzy Equalization Based on Neuron Grouping for Time-Varying Non-Linear Channels²

Equalização difusa adaptativa baseada em agrupamento por neurônios para canais de comunicação não lineares e variantes no tempo³

*Miguel Melgarejo-Rey⁴
Andrés Gaona-Barrera⁵
Carlos Barreto-Suárez⁶*

¹ Fecha de recepción: 22 de noviembre de 2010. Fecha de aceptación: 23 de mayo de 2011. Este artículo es desarrollado por el Laboratorio de Automática, Microelectrónica e Inteligencia Computacional (LAMIC) de la Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

² Submitted on: November 22, 2010. Accepted on: May 23, 2011. This article was developed by the Automation, Microelectronics and Computational Intelligence Laboratory (LAMIC) of the Universidad Distrital Francisco José de Caldas, Bogotá, Colombia.

³ Data de recepção: 22 de novembro de 2010. Data de aceitação: 23 de maio de 2011. Este artigo foi desenvolvido pelo Laboratório de Automática, Microeletrônica e Inteligência Computacional (LAMIC) da Universidade Distrital Francisco José de Caldas, Bogotá, Colômbia.

⁴ Ingeniero electrónico, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Magíster en Ingeniería Electrónica, Universidad de los Andes, Bogotá, Colombia. Investigador, Universidad Distrital Francisco José de Caldas. Correo electrónico: mmelgarejo@udistrital.edu.co.

⁵ Ingeniero electrónico, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Magíster en Ingeniería Electrónica y de Computadores, Universidad de los Andes, Bogotá, Colombia. Investigador, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Correo electrónico: aegaona@udistrital.edu.co.

⁶ Estudiante de Ingeniería Electrónica, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia. Correo electrónico: carlobar@gmail.com.

Resumen

Este artículo presenta una aproximación para la ecualización de canales no lineales variantes en el tiempo, basada en sistemas difusos y entrenamiento de neuronas individuales. El método tiene dos etapas: la primera usa aprendizaje supervisado, con el fin de estimar los estados del canal y proveer una sintonía inicial de los parámetros del ecualizador difuso, y la segunda ajusta dinámicamente al ecualizador para seguir el comportamiento variante del canal por medio de aprendizaje no supervisado. La propuesta se compara con una red de base radial sobre la ecualización de un canal de comunicaciones variante en el tiempo reportado previamente en la literatura. Los experimentos se llevan a cabo por medio de simulaciones de Monte Carlo. Los resultados muestran que el método propuesto tiene un mejor desempeño que una red de base radial en términos de la tasa de bits erróneos de un sistema de comunicación.

Palabras clave

Comunicaciones digitales, ecualizadores (electrónica), filtros adaptivos, redes neurales (computadores), sistemas difusos.

Abstract

This paper presents an approach for time varying non-linear channel equalization based on fuzzy systems and single-neuron training. The method consists of two stages: the first one uses supervised learning in order to determine channel states and to provide an initial tuning of the fuzzy equalizer parameters. The second one dynamically adjusts the equalizer to follow the varying behavior of the channel through unsupervised learning. This proposal is compared with a radial basis network over the equalization of a time-varying communication channel reported in previous works. Experiments are carried out through Monte Carlo simulations. Results show that the proposed approach presents a performance than that of a radial basis function in terms of the bit error rate of a communication system.

Key words

Digital communications, equalizers (electronics), adaptive filters, neural networks (computer science), fuzzy system.

Resumo

Este artigo apresenta uma aproximação para a equalização de canais não lineares variantes no tempo, baseada em sistemas difusos e treinamento de neurônios individuais. O método tem duas etapas: a primeira usa aprendizagem supervisada, com o fim de estimar os estados do canal e fornecer uma sintonia inicial dos parâmetros do ecualizador difuso, e a segunda ajusta dinamicamente ao equalizador para seguir o comportamento variante do canal por meio de aprendizagem não supervisada. A proposta se compara com uma rede de base radial sobre a equalização de um canal de comunicações variante no tempo relatado anteriormente na literatura. Os experimentos foram realizados através de simulações de Monte Carlo. Os resultados mostram que o método proposto tem um melhor desempenho que uma rede de base radial em termos da taxa de bits errôneos de um sistema de comunicação.

Palavras chave

Comunicações digitais, equalizadores (eletrônica), filtros adaptativos, redes neurais (computadores), sistemas difusos.

Introducción

En aplicaciones reales, la respuesta de los canales de comunicaciones digitales puede variar en función de la frecuencia de operación y algunos factores físicos. Esto ocasiona que las señales transmitidas sufran distorsiones en amplitud y fase, tales efectos son nocivos especialmente cuando las velocidades de transmisión aumentan. Cuando esto sucede los datos transmitidos se solapan entre sí y no pueden ser reconocidos en el receptor. Este fenómeno se conoce como interferencia intersímbolo (ISI, por sus siglas en inglés) (Proakis, 2001). La interferencia intersímbolo se puede corregir por medio de un ecualizador, cuyo papel es cancelar los efectos dañinos que produce el canal en las señales transmitidas.

La ecualización puede hacerse ya sea con el conocimiento de un modelo que describa el canal o sin este. Los ecualizadores que no requieren un modelo del canal son relativamente más simples. En estos casos, el problema de ecualización se entiende como un ejercicio de clasificación. La ecualización se puede llevar a cabo por medio de diferentes tipos de clasificadores, algunos de los cuales se basan en redes de base radial (RBF, por su sigla en inglés) (Assaf et ál., 2006; Chen et ál., 1993; Mulgrew, 1996) o sistemas difusos (Liang y Mendel, 2000; Patra y Mulgrew, 1998; Olarte et ál., 2005; Figueroa y Corrales, 2006; Lee, 1996). Dado que en estos ecualizadores no se establece un modelo del canal, es necesario estimar sus características durante una fase de entrenamiento.

Otro aspecto para considerar es el hecho de que las propiedades del canal pueden ser variantes en el tiempo. Esto genera la necesidad de hacer al ecualizador robusto ante los cambios que se puedan presentar. Lo anterior se logra haciendo que los parámetros del ecualizador se ajusten de forma automática a la dinámica cambiante del canal de comunicación (Assaf et ál., 2006).

El ecualizador que se desarrolla en este trabajo se basa en un filtro difuso (Liang y Mendel, 2000; Wang y Mendel, 1993; Patra y Mulgrew, 1998; Olarte et ál., 2005; Figueroa y Corrales, 2006). Junto con el filtro se implementa una metodología por medio de la cual se estiman los estados del canal y además se

ajustan los parámetros del ecualizador sobre la marcha. El método se centra en hacer agrupamiento entrenando neuronas individuales mediante el algoritmo *Least-Mean-Square* (LMS) (Bishop, 1995; Haykin, 1999).

El ecualizador propuesto se evalúa estimando su desempeño en simulación, según la metodología propuesta en (Liang y Mendel, 2000); el desempeño, de acuerdo con la tasa de datos errados presente a la salida del ecualizador. Esto se hace teniendo en cuenta diferentes aspectos como el ruido en el canal, la variabilidad de sus parámetros y las condiciones de entrenamiento.

Con el fin de comparar se considera una red de base radial. Se escoge esta técnica teniendo presente que se ha reportado en la literatura su uso como ecualizador de canales (Assaf et ál., 2006; Chen et ál., 1993; Mulgrew, 1996). Este artículo se organiza de la siguiente manera: en la sección 1 se presentan las características del modelo que representa al canal de comunicaciones. En la sección 2 se abarca todo el desarrollo del modelo propuesto. Los resultados de las simulaciones se presentan y discuten en la sección 3. Finalmente, se establecen algunas conclusiones en la sección 4.

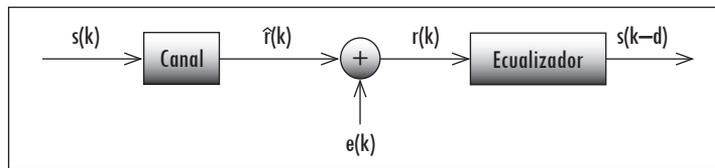
1. Modelo de un sistema de comunicaciones digital

Un modelo de un sistema de comunicación digital se muestra en la Figura 1. A la entrada del sistema hay una secuencia binaria $s(k)$ que se transmite a través de un canal de comunicación. En casos reales el canal no se comporta idealmente y, como consecuencia, la secuencia binaria se corrompe por los efectos de la ISI y ruido del canal. El efecto de la ISI que se genera en un canal dispersivo y variante en el tiempo se puede describir como (Assaf et ál., 2006):

$$r(k) = \sum_{i=0}^{n_b-1} a_i(k) \cdot s(k-i) + e(k) \quad (1)$$

Donde n_b es el número de símbolos que se traslapan a causa de la ISI, $s(k-i)$ es el dato transmitido en el instante $k-i$, $a_i(k)$ es el peso que se asigna al i -ésimo símbolo que interviene en la ISI y su dinámica es cambiante en el tiempo. $e(k)$ es ruido blanco gaussiano aditivo que afecta el canal. En el modelo se asume que los datos transmitidos $s(n)$ son independientes e igualmente probables (Chen et ál., 1993).

Figura 1. Representación del sistema de comunicaciones



Fuente: presentación propia de los autores.

Un ecualizador está situado a la salida del canal y su propósito es recuperar la señal $s(k)$ a partir del vector de la señal con ruido $\vec{r}(k) = [r(k), r(k-1), \dots, r(k-m+1)]^T \in R^m$, donde m representa el orden del ecualizador. En la ecualización de canal los símbolos que se transmiten se tratan como estados. Así que la tarea del ecualizador consiste en clasificar la señal $\vec{r}(k)$ en uno de los estados que se definen para el canal (Chen et ál., 1993). La señal recuperada se representa como $s(k-d)$, donde d es un retardo que se considera entre la detección y la salida del ecualizador (Patra y Mulgrew, 1998).

2. Ecuación difusa adaptativa asociada con agrupamiento realizado entrenando neuronas individuales

El método que se propone en este trabajo se presenta en la Figura 2. En este modelo el principal componente es el filtro difuso, el cual se encarga de realizar la ecualización de canal. Asociado al filtro difuso, se implementa un método para hacer agrupamiento por medio de neuronas. Este consiste en definir una neurona de dos entradas y una salida por cada uno de los estados del canal. Cada vez que se recibe un dato se estima a cuál estado del canal pertenece el vector $\vec{r}(k)$ y se entrena la neurona que corresponda por medio del bloque “Entrena Neurona”. Luego se extrae de la neurona, la información necesaria para ajustar los parámetros del ecualizador.

El entrenamiento del método está compuesto por dos etapas: en la primera se realiza la inicialización del ecualizador. Para esto se hace la transmisión de una trama inicial $s(k)$ conocida por el receptor. Esto con el fin de efectuar entrenamiento supervisado. En la segunda etapa, ya no se conocen los datos transmitidos, por lo que se considera un método de entrenamiento no supervisado. En ambos casos el entrenamiento se hace en forma secuencial (Bishop, 1995). El bloque ‘Entrena Neurona’ requiere conocer a qué estado pertenece el vector de entrada $\vec{r}(k)$, tanto en el entrenamiento supervisado como no supervisado. Esta información se extrae por medio de los bloques “Determina Estado” y “Clasifica Entrada”, respectivamente.

Una vez que se efectúa el entrenamiento de la neurona designada se sintonizan los parámetros del filtro difuso. El ajuste se hace solo a los parámetros que estén relacionados con la neurona entrenada en cada momento. A continuación se describen las características del filtro difuso y los procesos que se desarrollan durante las etapas de su entrenamiento.

2.1 Filtro difuso

El filtro difuso tiene la tarea de inferir qué dato se transmite en cada momento. Para determinar las características del filtro es necesario conocer el número de datos que se traslapan por la ISI (n_b) y el orden del ecualizador (m). Con lo anterior es posible calcular el número de estados del canal de dimensión m (n_s), el número de funciones de pertenencia M definidas por cada una de las m dimensiones y el número de entradas del filtro (m), entre otros (Assaf et ál., 2006; Haykin, 1999). Las funciones de pertenencia del filtro se configuran de acuerdo con lo propuesto en (Patra y Mulgrew, 1998) para lograr una función de decisión óptima.

El sistema difuso que se propone es de tipo Takagi-Sugeno sin normalización de m entradas y una salida (Wang, 1997). Se definen funciones de pertenencia gaussiana para que el modelo de filtro se ajuste a la forma de un ecualizador bayesiano (Patra y Mulgrew, 1998). Las reglas del sistema difuso son de la forma:

Si $r(k)$ es A_l^l y ... y $r(k - m + 1)$ es A_m^l , entonces $y^l = C^l$

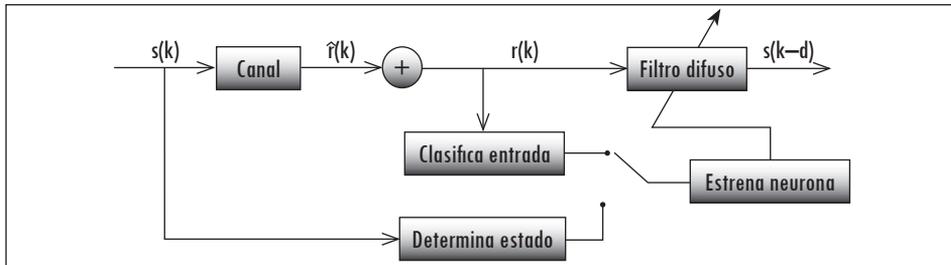
Donde A_l^l son las funciones de pertenencia, C^l son constantes binarias y $l = 1, 2, \dots, n_s$. Los parámetros que se ajustan en el filtro difuso son las medias (c_l^{ij}) y desviaciones estándar (σ) de las funciones de pertenencia.

2.2 Entrenamiento supervisado

Por lo general, únicamente se conocen los valores de m y n_b de un canal de comunicaciones (Liang y Mendel, 2000; Wang y Mendel, 1993; Patra y Mulgrew, 1998; Olarte et ál., 2005; Figueroa y Corrales, 2006; Lee, 1996). Con esta información se puede calcular el número de estados del canal (n_s) y el número de funciones de pertenencia (M) en cada dimensión. Sin embargo, la información es insuficiente para determinar las características de las funciones de pertenencia y el valor binario C^l que le corresponde a cada estado del canal. Ya que la información desconocida es necesaria para ajustar el filtro difuso, se propone realizar una etapa de entrenamiento supervisado en la que se estimen dichos parámetros. En la Figura 2, la etapa de entrenamiento supervisado se representa por el lazo

que conecta a la secuencia de datos transmitida $s(k)$ con el bloque “Determina Estado”, y este a su vez con el bloque “Entrena Neurona”.

Figura 2. Representación del modelo propuesto



Fuente: presentación propia de los autores.

Ya que se conoce el número de estados del canal y la secuencia $s(k)$, es posible asignar a cada estado del canal un determinado vector $\vec{s}(k) = [s(k), \dots, s(k - n_n - m + 2)]$. Así, el valor binario C^l de cada estado corresponde al valor de $s(k)$ que se le asigna a cada uno de ellos. El bloque “Determina Estado” usa la secuencia de datos $\vec{s}(k)$ para encontrar a qué estado del canal pertenecen los datos recibidos $\vec{r}(k)$ y realizar el entrenamiento de la respectiva neurona. El entrenamiento de las neuronas en esta etapa permite ajustar los parámetros de las funciones de pertenencia de acuerdo con el canal de comunicación.

2.3 Entrenamiento no supervisado

Luego del entrenamiento supervisado, el filtro difuso adopta una configuración determinada. Debido a que las características del canal son variantes en el tiempo, es necesario ajustar los parámetros del filtro de acuerdo con la dinámica del canal. Esto se hace durante el entrenamiento no supervisado. Durante esta etapa no se conocen los datos transmitidos, por lo que se debe implementar un mecanismo que determine a cuál estado pertenece el dato entrante en cada momento. Para esto solo se dispone de la información de los datos recibidos $r(k)$. El bloque “Clasifica Entrada” es el encargado de efectuar esta tarea. Este bloque consiste en un sistema difuso que determina a qué punto de la constelación pertenece el dato de entrada. Sabiendo a cuál estado del canal pertenecen los datos $r(k)$, es posible entrenar las respectivas neuronas para ajustar los parámetros del filtro difuso.

El bloque “Clasifica Entrada” es un sistema difuso que determina a qué punto de la constelación pertenece un determinado dato $r(k)$. Este bloque tiene la misma estructura del filtro difuso, aunque la principal diferencia radica en que busca

clasificar el vector $\bar{r}(k)$ en uno de los n_s estados del canal representados con S^l . Este sistema es de tipo Takagi-Sugeno sin normalización con m entradas y una salida. Las reglas de inferencia son de la forma:

$$\text{Si } r(k) \text{ es } A_1^l \text{ y } \dots \text{ y } r(k - m + 1) \text{ es } A_m^l, \text{ entonces } y^l = S^l$$

Donde A_i^l son las funciones de pertenencia, S^l toma valores enteros entre 1 y n_s , y $l = 1, 2, \dots, n_s$. Se puede observar que en los sistemas difusos “Filtro Difuso” y “Clasifica Entrada” se definen reglas similares. La diferencia radica en las proposiciones de los consecuentes. Este hecho se puede aprovechar para implementar un solo sistema de inferencia difusa de dos entradas y dos salidas. Para esto es necesario realizar la inferencia y la *defusificación* de los dos sistemas difusos por aparte.

2.4 Entrenamiento de las neuronas

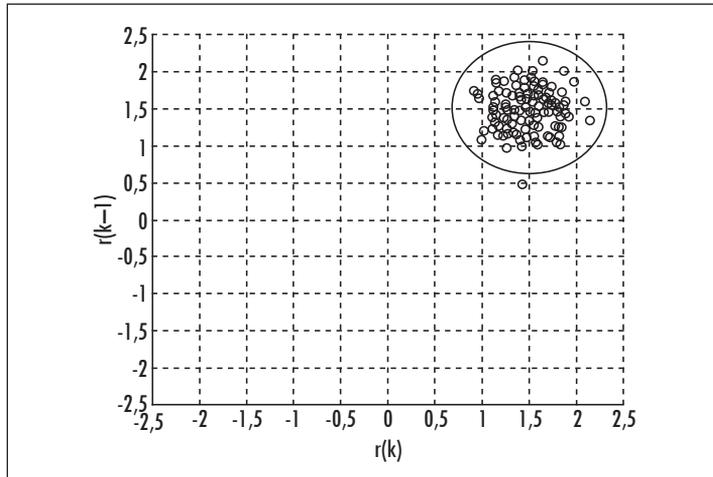
Una vez se conoce a qué estado del canal pertenece el dato entrante, se entrena la neurona asignada a dicho estado. El método que se propone busca hacer agrupamiento por medio del entrenamiento de neuronas individuales (Bishop, 1995; Haykin, 1999). Para tal fin se considera inicialmente que la clasificación apropiada para una nube de puntos tiene forma de elipse (Figura 3a). Lo anterior se hace dado que una elipse tiene cinco grados de libertad (orientación, forma, escala y posición en los ejes X e Y); mientras que los círculos sólo tienen tres (escala y posición en los ejes X e Y). Es necesario tener en cuenta que la orientación de la elipse no se tiene en cuenta para el desarrollo. La ecuación que representa una elipse con centro en el punto (c_x, c_y) es:

$$\frac{(x - c_x)^2}{a^2} + \frac{(y - c_y)^2}{b^2} = c^2 \quad (2)$$

Donde x y y representan las señales $r(k)$ y $r(k - 1)$, respectivamente. El punto (c_x, c_y) corresponde a la posición del estado del canal al que pertenece el dato entrante, mientras que a , b y c son parámetros que determinan la forma de la elipse. Al hacer la transformación sobre los datos: $u = (x - c_x)^2$ y $v = (y - c_y)^2$ resulta la ecuación lineal:

$$\frac{u}{a^2} + \frac{v}{b^2} = c^2 \quad (3)$$

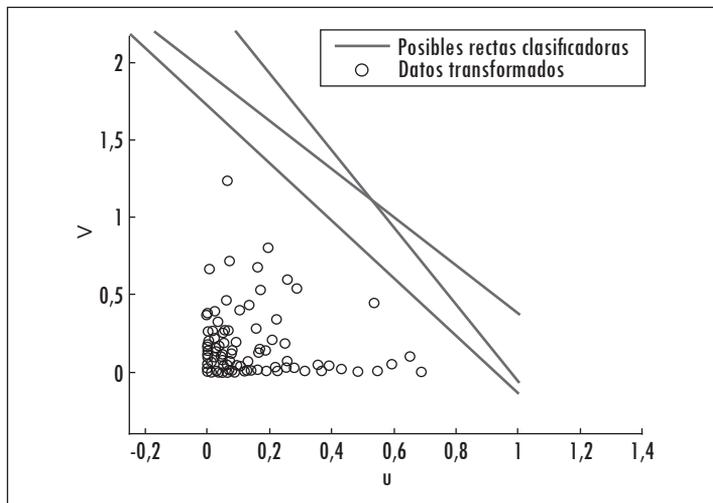
Figura 3a. Representación de la clasificación ideal para una nube de puntos y algunas clasificaciones realizadas por una neurona por medio de una elipse



Fuente: presentación propia de los autores.

En la transformación propuesta se desplazan los datos alrededor del origen para luego elevarlos al cuadrado. Como consecuencia, los datos se sitúan cerca al punto de origen y siempre en el primer semiplano del plano cartesiano (Figura 3b).

Figura 3b. Representación de la clasificación ideal para una nube de puntos y algunas clasificaciones realizadas por una neurona: datos transformados y algunas clasificaciones posibles



Fuente: presentación propia de los autores.

Luego de transformar los datos según la ecuación (3), el clasificador que separaría convenientemente los puntos corresponde a una línea recta. Dicha ecuación se puede encontrar entrenando en forma secuencial una neurona de dos entradas y una salida. La ecuación que representa la neurona descrita es:

$$w_0 + x_1 \cdot w_1 + x_2 \cdot w_2 = 0 \quad (4)$$

Donde x_1 y x_2 representan las variables u y v respectivamente. w_0 , w_1 y w_2 son los pesos de la neurona. La relación entre los pesos de la neurona y los coeficientes de la ecuación lineal es:

$$c^2 = -w_0 \quad (5)$$

$$a^2 = \frac{1}{w_1} \quad (6)$$

$$b^2 = \frac{1}{w_2} \quad (7)$$

Usando las expresiones anteriores se pueden extraer los parámetros de la elipse clasificadora a partir del entrenamiento individual de neuronas. Estos coeficientes se emplean en la sección 2.6 para extraer la desviación estándar σ_e de las funciones de pertenencia del filtro difuso.

Es necesario resaltar que la clasificación de los puntos transformados se puede hacer por medio de infinidad de rectas (Figura 3b). Aunque estas permitan clasificaciones adecuadas, con cada una se obtienen diferentes resultados en la clasificación final de los puntos. Un factor que determina la forma de la clasificación es la pendiente de la recta. En el caso de rectas con pendientes positivas se generan hipérbolas en lugar de elipses.

Para que la clasificación resultante se ajuste a la nube de puntos, debe procurarse que las rectas se sitúen justo en la frontera de los datos transformados. Por ello es conveniente introducir un discriminante de dos clases y se requieren dos clases de datos, pero en este caso la única clase existente (clase 1) corresponde al conjunto de datos que se transforma según la ecuación (3).

Dado que no se tiene una segunda clase, se propone que esta se obtenga a partir de la clase conocida. Para esto se transforman los datos de la clase 1 de tal forma que la nueva clase se sitúe junto a la original. Es necesario buscar que las dos clases sean separables linealmente y además que la clasificación resultante corresponda con los datos originales. Lo último se garantiza haciendo que las dos clases no se solapen entre sí y que la distancia que las separa no sea muy grande.

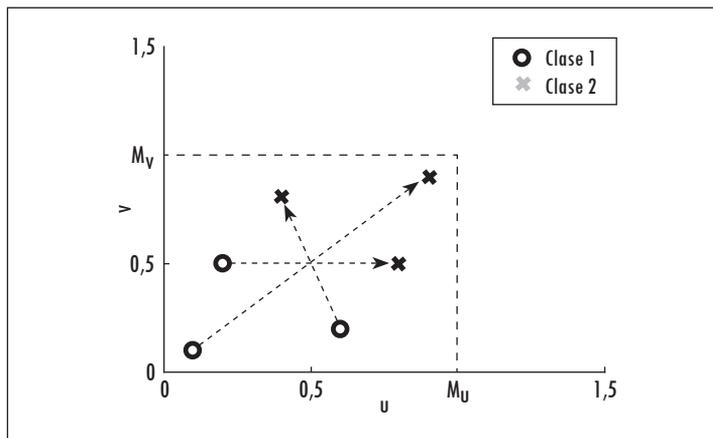
La transformación que se propone consiste en hacer un desplazamiento de los datos conocidos. Para esto es necesario definir los márgenes M_U y M_V en los ejes u y v , respectivamente, como se indica en la Figura 4. Teniendo como referencia los márgenes, cada punto de la clase 1 se desplaza a la posición opuesta (Figura 4). La posición de los puntos de la nueva clase se calcula según:

$$P_{2Ui} = |M_U - P_{1Ui}| \tag{8}$$

$$P_{2Vi} = |M_V - P_{1Vi}| \tag{9}$$

Donde $\{P_{1Ui}, P_{1Vi}\}$ y $\{P_{2Ui}, P_{2Vi}\}$ son los vectores de posición de los datos pertenecientes a las clases 1 y 2, respectivamente. Usando valor absoluto en las expresiones se garantiza que los datos generados se ubiquen siempre en el primer semiplano del plano cartesiano. Ya que el entrenamiento se hace de forma secuencial, la creación de la nueva clase debe hacerse con los datos disponibles en cada momento i .

Figura 4. Representación del proceso de creación de la clase 2



Fuente: presentación propia de los autores.

La elección de los márgenes es importante, dado que estos afectan la creación de la nueva clase y, en consecuencia, influyen en la clasificación que hacen las neuronas. Puesto que este trabajo trata canales variantes en el tiempo se hace que los márgenes M_U y M_V se ajusten durante el entrenamiento por medio de:

$$M_{U_{i+1}} = \frac{M_U + \max (P_{U_i}, M_U)}{2} \tag{10}$$

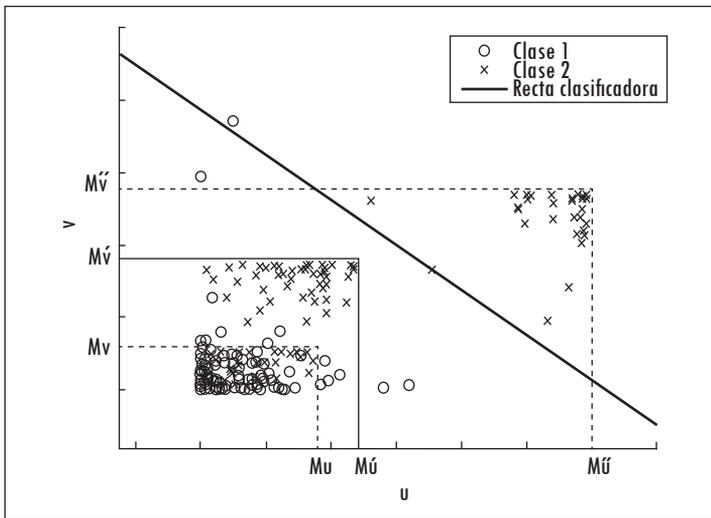
$$M_{V_{i+1}} = \frac{M_V + \max (P_{V_i}, M_V)}{2} \tag{11}$$

Con lo anterior se busca que mediante una ecuación sencilla se corrijan los márgenes durante el proceso de entrenamiento supervisado y no supervisado. Las ecuaciones propuestas permiten ajustar los márgenes solamente cuando se encuentre algún punto que esté por fuera de ellos. Además, los ajustes que se hacen son proporcionales a la distancia entre el punto y el margen en cuestión.

Las ecuaciones (12) y (13) se derivan de forma empírica. Sin embargo, algo para tener en cuenta es que los márgenes no pueden reducirse por medio de ellas. Es posible que esta propiedad sea inconveniente en algunas aplicaciones de agrupamiento. Para tales casos sería posible reemplazar las ecuaciones por otras más favorables. El valor inicial de los márgenes M_U y M_V se estima como un parámetro de diseño del sistema.

En la Figura 5 se muestra la clasificación realizada por una neurona. Allí se puede ver la clase inicial y la clase que se deriva a partir de esta. En este caso se observan unas agrupaciones de puntos en la segunda clase. Dichas agrupaciones evidencian los ajustes que se hacen en los márgenes M_U y M_V durante el entrenamiento.

Figura 5. Clasificación realizada utilizando un discriminante de dos clases



Fuente: presentación propia de los autores.

Las neuronas se entrenan utilizando el algoritmo LMS (Bishop, 1995; Haykin, 1999), porque en la literatura se ha reportado su uso en el entrenamiento de ecualizadores (Assaf et ál., 2006; Chen et ál., 1993). Para cada función de pertenencia en el filtro difuso se debe establecer el valor de la media y la desviación estándar (σ_e). El valor de la desviación se extrae a partir de la información de las neuronas, mientras que la media se calcula mediante la media estocástica. El cálculo dichos parámetros se debe realizar teniendo en cuenta la información aportada por los estados relacionados con cada función de pertenencia. En este caso, el cálculo de los parámetros se hace por medio del promedio de los valores suministrados por los estados del canal involucrados.

2.5 Cálculo de las medias de los conjuntos difusos

La media de las funciones de pertenencia corresponde a la posición de los estados del canal. La posición de los estados del canal se calcula estimando el promedio de los puntos que pertenece a cada estado. Lo anterior se hace respecto a cada eje, que en el caso de un canal de segundo orden son $r(k)$ y $r(k-1)$. Dado que este cálculo debe hacerse en forma secuencial, se propone calcular la media estocástica (Bishop, 1995):

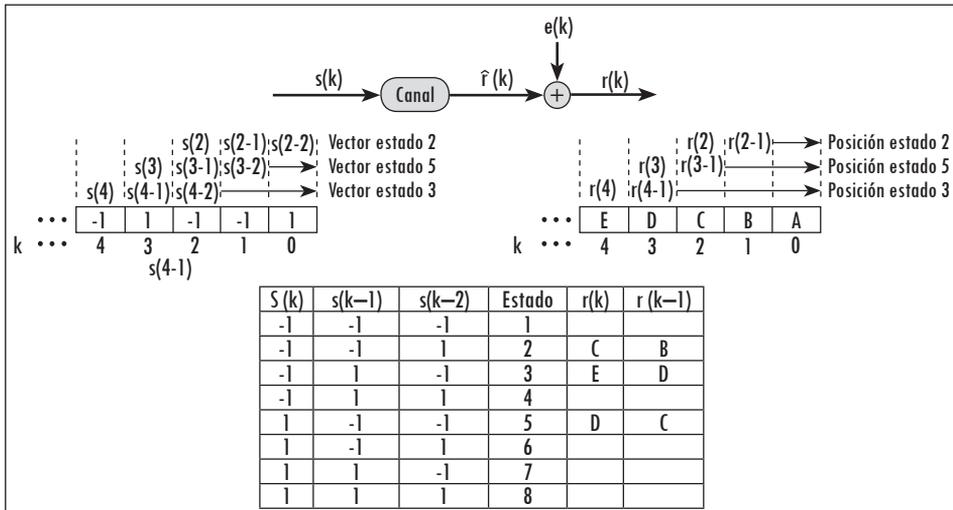
$$P_{N+1} = P_N + \frac{Dato_{N+1} - P_N}{1 + N} \quad (12)$$

Donde P_N es el promedio calculado en el instante N , mientras que $Dato_{N+1}$ y P_{N+1} corresponden al dato recibido y el promedio calculado en el instante $N + 1$.

Dado que inicialmente no se conoce el modelo del canal, no es posible determinar dónde se localizan los puntos de la constelación. Por lo anterior, se propone estimar la posición inicial de los estados a partir de la información conocida en el receptor durante el entrenamiento supervisado. En dicha etapa se dispone de la secuencia de inicial $s(k)$ y los datos recibidos $r(k)$. Para determinar la posición inicial de los estados se lee un vector de datos $\bar{s}(k)$ y se le asigna a cada estado del canal un vector $\bar{r}(k)$ que indica la posición del estado.

El proceso para un canal con $m = 2$ y $n_b = 2$ se representa en la Figura 6. Allí se muestra una secuencia de inicialización binaria $s(k)$ transmitida a través del canal junto con las señales $r(k)$ afectadas por la ISI y el ruido del canal (representadas con las letras A, B, C, D y E).

Figura 6. Representación del proceso de selección de la posición inicial de los estados



Fuente: presentación propia de los autores.

2.6 Cálculo de la desviación estándar σ_e de los conjuntos difusos

Eventualmente la clasificación encontrada por medio de las neuronas encerraría el 100% de los datos. A partir de las ecuaciones (5) a (7) se encuentra la elipse que representa el clasificador para un determinado estado. Luego se proyecta la elipse hacia los ejes del plano cartesiano. Con esto se delimita la región que encierra la clasificación en cada eje. Para establecer el valor de σ_e se considera un punto de la función de pertenencia que esté entre 0,01 y 0,012. En este caso se tienen en cuenta puntos de la forma $\pm k \cdot \sigma_e$, donde k es un número entero. Una ventaja de seleccionar un múltiplo de σ_e es que simplifica las ecuaciones (13) y (14). Las ecuaciones que permiten calcular la varianza en cada eje son:

$$\sigma_x^2 = \frac{-W_0}{\kappa^2 \cdot W_1} \tag{13}$$

$$\sigma_y^2 = \frac{-W_0}{\kappa^2 \cdot W_2} \tag{14}$$

3. Simulaciones

Inicialmente se plantea un modelo del canal de comunicación. Luego se evalúan las propuestas para realizar el cálculo de σ_e y la media de las funciones de pertenencia. Finalmente, se evalúa el desempeño de la metodología propuesta en comparación con una red de base radial. En los experimentos se usan 200 y 1.800 datos para los entrenamientos supervisados y no supervisados.

En las simulaciones que se presentan el valor inicial de M_U y M_V es 0,1. En general se recomienda que el valor de los márgenes sea cercano o igual a cero. La recomendación se hace, ya que en términos reales habría desconocimiento del modelo del canal e incapacidad del método para reducir los márgenes. Por lo tanto, es posible que al seleccionar los márgenes estos sean inadecuados y no puedan ser corregidos por el método que se propone. Por otro lado, al escoger márgenes con valores pequeños el sistema tiene la posibilidad de ajustarlos de forma autónoma.

3.1 Modelo del canal de comunicación

En las simulaciones se considera un canal de segundo orden estudiado en otros trabajos (Liang y Mendel, 2000). El modelo que se emplea para representar los efectos de un canal no lineal y variante en el tiempo sobre los datos transmitidos (Liang y Mendel, 2000) es:

$$r(k) = a_1 \cdot s(k) + a_2 \cdot s(k-1) - 0,9 \cdot (a_1 \cdot s(k) + a_2 \cdot s(k-1))^3 + e(k) \quad (15)$$

Donde $r(k)$ es la señal recibida, $s(k)$ y $s(k-1)$ son los datos transmitidos en los instantes k y $k-1$, los coeficientes a_1 y a_2 caracterizan el comportamiento del canal y $e(k)$ es ruido blanco gaussiano aditivo que afecta el canal de comunicación. Para generar datos de un canal con los parámetros a_1 y a_2 variantes en el tiempo, se toma el método usado en (Liang y Mendel, 2000). Este método usa un modelo de Markov de segundo orden en el que una fuente de ruido gaussiano alimenta un filtro Butterworth pasabajos de segundo orden con frecuencia de corte de 0,1. En los experimentos la variable β representa la desviación estándar del ruido gaussiano.

3.2 Aproximación y validación de la media estocástica

En la sección 2.5 se plantea el uso de la media estocástica. En las simulaciones se realiza una aproximación que consiste en asignar a N un valor constante. Con el fin de observar el efecto del parámetro N en la ecuación (12) se generan datos según la ecuación (15) y sobre los datos pertenecientes a cada estado del canal se realiza el promedio por la forma convencional y la forma estocástica. Enseguida se calcula el error de la forma estocástica respecto al promedio real en cada uno de los estados y se selecciona el error más alto. Con el fin de observar si hay alguna relación de los resultados con el número de datos empleados en las pruebas se utilizan series de 200 y 1.800 datos para hacer las simulaciones.

Esas cantidades se utilizan en el entrenamiento supervisado y en el no supervisado más adelante. Para observar la condición de error mínimo de la media estocástica con diferentes condiciones del canal se hace $\beta = \{0,16; 0,28\}$ y $\text{SNR} = \{15 \text{ dB}, 25 \text{ dB}\}$. Los resultados de los experimentos se registran en la Tabla 1. El procedimiento que se describe se realiza variando N de 1 a 100 y se hacen 100 simulaciones de Monte Carlo en cada caso.

Tabla 1. Valores de N para los que se encuentra el error mínimo con diferentes condiciones del canal

Datos	SNR=15 dB		SNR=25 dB	
	$\beta = 0,16$	$\beta = 0,28$	$\beta = 0,16$	$\beta = 0,28$
200	N= 5	N=7	N=5	N=5
	Error= 14,87%	Error=17,93%	Error= 8,942%	Error=14%
1800	N=43	N=39	N=44	N=50
	Error= 5,79%	Error=6,182%	Error=3,298%	Error=5,143%

Fuente: presentación propia de los autores.

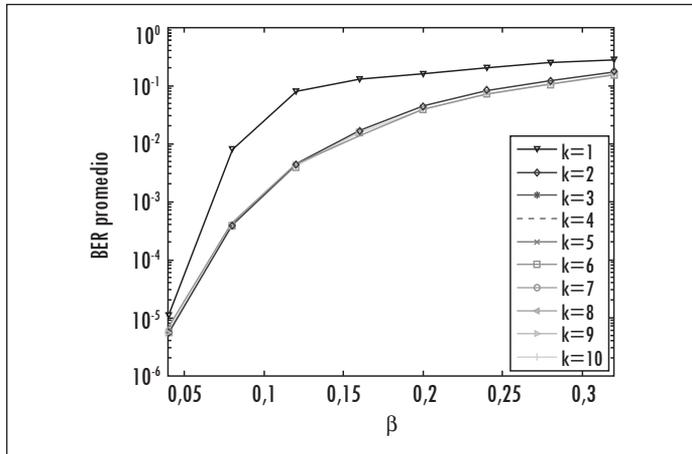
Gracias a los resultados que se muestran en la Tabla 1 se observa que el valor de N para el que se presenta el error mínimo varía de acuerdo con el número de datos usados. Sin embargo, el valor de N varía con las condiciones del canal. En el caso de 200 datos, N varía entre 5 y 7; mientras que para el caso de 1.800 datos N varía entre 39 y 50.

3.3 Estimación de k para el cálculo de σ_e

A continuación se valida la propuesta que se menciona en la sección 2.6. Para tal fin se evalúa el desempeño del método que se propone al variar k de 1 a 10. En la validación se ajusta la SNR en 20 dB y se varía β de 0,04 a 0,6 en saltos de 0,04.

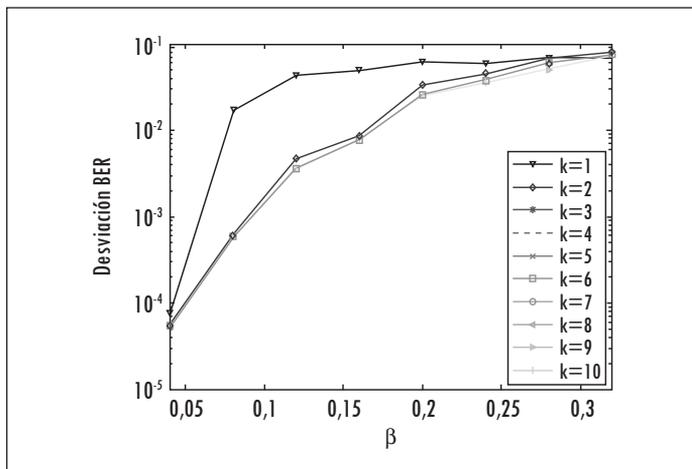
En la Figura 7 se observa que el peor desempeño se presenta con $k = 1$. Por otro lado, los mejores resultados se registran cuando k toma valores entre 3 y 10. En estos casos no se aprecian diferencias significativas en el desempeño. Haciendo $k = 2$ se obtienen tasas de error cercanas a las mejores registradas. Para las simulaciones que siguen se hace $k = 3$, pues la pertenencia en dicho punto es de 0,011.

Figura 7a. Desempeño del método propuesto variando k de 1 a 10, con un SNR ajustado en 20 dB y variando β de 0,04 a 0,32 en pasos de 0,04: promedio de 100 simulaciones de Monte Carlo por cada valor de β



Fuente: presentación propia de los autores.

Figura 7b. Desempeño del método propuesto variando k de 1 a 10, con un SNR ajustado en 20 dB y variando β de 0,04 a 0,32 en pasos de 0,04: desviación de 100 simulaciones de Monte Carlo por cada valor de β



Fuente: presentación propia de los autores.

3.4 Evaluación de la metodología propuesta con respecto a una RBN

En la evaluación se compara el desempeño del método propuesto con el de una red de base radial. La red que se implementa es de dos entradas y una salida. Las funciones de base que se eligen son funciones gaussianas. Inicialmente se asumen desviaciones de 0,5 para todas las funciones. Los pesos de la red se encuentran implementando el algoritmo de aprendizaje supervisado LMS (Bishop, 1995; Haykin, 1999) y los centros de la red se estiman de forma secuencial por medio de un algoritmo competitivo de aprendizaje no supervisado (Assaf et ál., 2006). El número de neuronas ocultas es igual al número de puntos de la constelación que se requiere discriminar, es decir, n_s . El valor inicial de los centros se determina empleando el método que se describe en la sección 2.5.1.

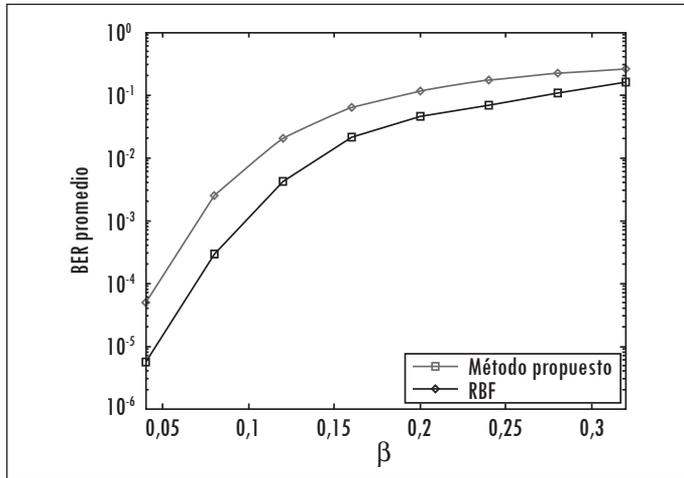
En el siguiente experimento se iguala N a 5 y 43 para realizar el entrenamiento supervisado y el no supervisado, respectivamente. Además, se usan 2.000 datos y para cada prueba se hacen 100 simulaciones de Monte Carlo (Liang y Mendel, 2000). Las diferencias en el desempeño de los métodos se encuentran promediando la diferencia porcentual del método propuesto con respecto a la RBN, mediante la siguiente expresión:

$$D = \frac{1}{Q} \sum_{i=1}^Q \frac{m_{1i} - m_{2i}}{m_{1i}} \cdot 100 \quad (16)$$

Donde Q es el número de puntos evaluados en cada prueba y m_{1i} y m_{2i} son los resultados de la red de base radial y el método propuesto, respectivamente. Para los casos en que $D > 0$ el método propuesto tiene una mejora promedio de D respecto a una RBN.

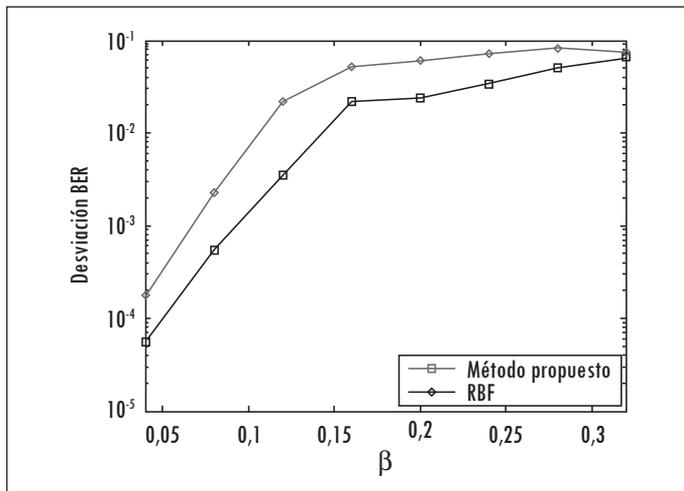
Se busca probar los métodos ante condiciones variantes en el tiempo del canal. En este caso se ajusta la SNR en 20 dB y se varía β entre 0,04 y 0,32 con pasos de 0,04. En la Figura 8 se muestran los resultados de las simulaciones del método propuesto y la red de base radial. Se observa que con el método propuesto se obtiene una mejora promedio D del 68,2276% en el desempeño respecto a la red de base radial. El desempeño de nuestra propuesta es mejor para todos los valores de β que se consideraron.

Figura 8a. Desempeño del método propuesto y la RBF con SNR = 20 dB y variando β de 0,04 a 0,32 en pasos de 0,04: promedio de 100 simulaciones de Monte Carlo por cada valor de β



Fuente: presentación propia de los autores.

Figura 8b. Desempeño del método propuesto y la RBF con SNR = 20 dB y variando β de 0,04 a 0,32 en pasos de 0,04: desviación de 100 simulaciones de Monte Carlo por cada valor de β



Fuente: presentación propia de los autores.

4. Conclusiones y trabajo futuro

En este trabajo se presenta una metodología para la ecualización de canales de comunicaciones digitales no lineales y variantes en el tiempo. Entre sus aportes se encuentra el desarrollo del método para agrupar por medio del entrenamiento de neuronas individuales y los aspectos relacionados con la sintonización dinámica del filtro difuso, como son las propuestas para estimar la media estocástica aproximada, la posición inicial de los estados del canal y la desviación estándar de las funciones de pertenencia del filtro difuso.

La metodología propuesta es híbrida neurodifusa y combina la capacidad para resolver problemas no lineales de los sistemas difusos junto con la habilidad de aprendizaje de las neuronas. Los resultados de las simulaciones mostraron que la ecualización realizada por medio del método propuesto es mejor que la obtenida mediante una red de base radial.

En los experimentos se observó que con la metodología planteada se obtiene una mejora en el desempeño del 68,2276% respecto a la red de base radial para todos los valores de β considerados. Además, se evidenció que en realidad lo que marca la diferencia en el desempeño es la capacidad de adaptación que cada sistema tiene ante cambios en el canal.

Como trabajo futuro sería interesante mejorar el método propuesto incluyendo más términos en la ecuación (3). Esto con el fin de dar al método la capacidad para modificar la orientación de la elipse. De esta forma se le daría un grado de libertad adicional a la elipse que representa la clasificación. El método que se propone para hacer agrupamiento es diseñado para problemas de agrupamiento en dos dimensiones. Sería conveniente extender la propuesta para poder resolver problemas de agrupamiento de n dimensiones.

Referencias

- ASSAF, R.; EL ASSAD, S. y HARKOUSS, Y. Adaptive equalization of nonlinear time varying-channels using radial basis network. *Information and Communication Technologies, 2006. ICTTA '06. 2nd.* 2006.
- BISHOP, C. M. *Neural networks for pattern recognition.* Oxford: Oxford University Press. 1995.
- CHEN, S.; MULGREW, B. y GRANT, P. M. A clustering technique for digital communications channel equalization using radial basis function networks. *Neural Networks, IEEE Transactions on.* 1993, vol. 4, núm. 4, pp. 570-590.
- FIGUEROA, J. A. y CORRALES, D. A. Realización hardware de un algoritmo de sintonía automática para ecualizadores difusos de canales de comunicación no lineales de orden Dos. *Ingeniería.* 2006a, pp. 68-74.

- HAYKIN, S. *Neural networks. A comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall PTR, 1999.
- LEE, K. Y. Complex fuzzy adaptive filter with LMS algorithm. *Signal Processing, IEEE Transactions on*. 1996, vol. 44, núm. 2, pp. 424-427.
- LIANG, Q. y MENDEL, J. M. Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters. *Fuzzy Systems, IEEE Transactions on*. 2000, vol. 8, núm. 5, pp. 551-563.
- MULGREW, B. Applying radial basis functions. *Signal Processing Magazine, IEEE*. 1996, vol. 13, núm. 2, pp. 50-65.
- OLARTE, F.; LADINO, P. y MELGAREJO, M. Hardware realization of fuzzy adaptive filters for non linear channel equalization. *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. 2005, vol. 2, pp. 932-935.
- PATRA, S. K. y MULGREW, B. Efficient architecture for Bayesian equalization using fuzzy filters. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*. 1998, vol. 45, núm. 7, pp. 812-820.
- PROAKIS, J. G. *Digital communications*. New York: McGraw-Hill, 2001.
- WANG, L. X. *A course in fuzzy systems and control*. Upper Saddle River, NJ: Prentice-Hall Int, 1997.
- WANG, L.-X. y MENDEL, J. M. Fuzzy adaptive filters, with application to nonlinear channel equalization. *Fuzzy Systems, IEEE Transactions on*. 1993, vol. 1, núm. 3, pp. 161-170.

