Modelamiento difuso con técnicas de Clustering

Diana Marcela González1 Chaparro[1]

Sergio Barato Quintero [2]

RESUMEN

En este documento se presentan técnicas para derivar modelos difusos Takagi-Sugeno-Kang (TSK) de sistemas complejos, no lineales y semidesconocidos a partir de métodos de clustering (agrupamiento). Se utilizan tres algoritmos: Gustafson-Kessel (GK), Maximum Likelihood Estimation (MLE) y una modificación a la versión simplificada del algoritmo de Maximum Likelihood. Estos son evaluados en condiciones de presencia ruido. De los resultados de las simulaciones se demostró que el algoritmo menos vulnerable ante ruido es el GK. Adicionalmente, se encontró que en condiciones de poco ruido la generación de submodelos lineales eficientemente se obtuvo con el algoritmo MLE modificado.

Palabras clave: clustering difuso, centros, función objetivo, matriz de partición, modelo difuso TSK.

I. INTRODUCCION

Es de gran importancia el desarrollo de modelos matemáticos de sistemas reales. Estos permiten obtener simulaciones, analizar el comportamiento del sistema y diseñar procesos. Sin embargo, muchos sistemas físicos no son tratables por aproximaciones desde el punto de vista de modelamiento convencional, debido a la falta de conocimiento del sistema, fuerte comportamiento no lineal, y alto grado de incertidumbre.

El modelamiento difuso ha sido reconocido como una poderosa herramienta en el tratamiento de sistemas que presentan los problemas mencionados, debido a la capacidad de integrar información de diferentes fuentes tales como, conocimiento de expertos, modelos empíricos o mediciones. Los conjuntos difusos sirven como interfaz entre las variables cualitativas, los datos numéricos, las entradas y salidas del sistema. Entre las metodologías de identificación difusa se encuentran las técnicas de clustering. Estas se basan en el manejo de datos de entrada y salida del sistema, y su agrupamiento en subconjuntos (clusters). Como gran ventaja, estas técnicas presentan la capacidad de sobrellevar los problemas de no linealidades y la falta de conocimiento preciso del sistema.

II. IDENTIFICACION Y MODELAMIENTO DIFUSO

Las técnicas y algoritmos para el reconocimiento de los parámetros de un modelo de dato se conocen como identificación difusa. Los parámetros de una estructura difusa como funciones de pertenencia y el peso de las reglas son sintonizadas usando los datos de entrada y salida. En este proceso no se hace necesario un conocimiento a priori, en lugar de ello se espera que las funciones de pertenencia y reglas extraídas proporcionen una interpretación del comportamiento del sistema. La identificación es vista como una descomposición de un sistema no lineal, lo cual genera un balance entre la complejidad (no uniforme) y la exactitud del modelo.

El modelamiento difuso reúne el procesamiento lógico de información con estructuras matemáticas capaces de representar mapeos no lineales complejos. La estructura basada en reglas de los sistemas difusos, contribuye a los modelos difusos generados por tratamiento de datos, ya que generan una descripción cualitativa, que combinada con el conocimiento de expertos ayuda a comprender, validar y/o simplificar el modelo.

III. MODELOS TSK

Los modelos TSK son modelos difusos que constan de reglas R_i con la siguiente estructura [1] [2] [9] [10]:

$$R_i: \textbf{Si } x \text{ es } A_i \textbf{ entonces } y_i = \mathbf{a}_i^T x + \mathbf{b}_i \quad i = 1,2,\ldots,K \quad (1)$$

Donde x es el vector de entrada, A_i es el conjunto difuso (multidimensional), yi es la salida de la i-esima regla, a_i es un vector paramétrico y b_i es el desplazamiento escalar. La proposición de entrada x es A_i puede ser expresada como una combinación lógica de proposiciones unidimensionales definidas por cada componente del vector de entrada x:

$$R_i: \textbf{Si } x_1 \text{ es } A_{i1} \textbf{ y...y } x_p \text{ es } A_{ip} \textbf{ entonces } y_i = \mathbf{a}_i^T x + \mathbf{b}_i \quad (2)$$

Tal como en los modelos difusos Mamdani el grado de cumplimiento de la regla esta dado por

$$\beta_i(x) = \mu_{A_{i,1}}(x) \wedge \mu_{A_{i,2}}(x) \wedge \ldots \wedge \mu_{A_{i,p}}(x) \quad (3)$$

[1] Miembro Grupo de Investigación Laboratorio de Automática, Microelectrónica e Inteligencia Computacional LAMIC,de la Universidad Distrial.

[2] Miembro Grupo de Investigación Laboratorio de Automática, Microelectrónica e Inteligencia Computacional LAMIC,de la Universidad Distrial.

Artículo recibido en Abril de 2003
Aceptado en Junio 2003

Donde ∧ representa una T-norma[1][4]. La salida del sistema TSK es obtenida combinando las reglas de la siguiente forma:

$$y = \frac{\sum_{i=1}^{c} \beta_i(x)(a_i^T x + b_i)}{\sum_{i=1}^{c} \beta_i(x)} \qquad (4)$$

Tomando la normalización del grado de cumplimento como:

$$\varphi_i(x) = \frac{\beta_i(x)}{\sum_{i=1}^{c} \beta_i(x)} \qquad (5)$$

La ecuación (4) se puede rescribir como:

$$y = \sum_{i=1}^{c} \varphi_i(x)(a_i^T x + b_i) \qquad (6)$$

La anterior expresión muestra que los modelos TSK puede cumplir el papel de regresores de funciones, es decir pueden aproximar con cierto margen de exactitud cualquier función y = f(x).

IV. TECNICAS DE CLUSTERING

Las técnicas de clustering surgieron dentro de la disciplina patrones de reconocimiento. Esta técnica busca hallar subgrupos con cierto grado de similaridad dentro de una colección de datos. Además de identificar cada subgrupo, determina los parámetros representativos de cada uno. Originalmente se planteó la idea de que cada dato solo podría pertenecer a un solo subgrupo, denominado "hard" clustering. Mas adelante se tomó la idea de clustering difuso en el cual particionamiento de datos se hace de forma tal que la transición entre conjuntos es gradual en lugar de abrupta como lo haría el "hard clustering" [3]-[4]. Para el estudio, la similaridad es definida como una *distancia*, la cual puede ser medida entre vectores que contienen los datos y un elemento prototipo del cluster, normalmente el centro [4].

La identificación difusa es una efectiva herramienta para la aproximación de sistemas no-lineales inciertos basándose en datos medidos; existen varias técnicas para esta identificación y entre ellas están los métodos de agrupamiento o clustering difuso.

Existen varios algoritmos de clustering los cuales tienen en cuenta: la forma geométrica, densidad y relación espacial de cada cluster, además de la distancia entre ellos [3][4]. Se encuentran algoritmos iterativos basados en la optimización de una función objetivo diferenciable, con la cual se mide la conveniencia de la partición. En este documento se trabaja sobre este tipo específico de algoritmos. Una de las primeras técnicas de agrupamiento planteada que utiliza una función objetivo es conocida como *Fuzzy C-means* [4]. Una desventaja de esta técnica es que la medida de similaridad entre datos genera clusters circulares. En este artículo se trabajan tres algoritmos: Gustafson-Kessel, Maximum Likelihood, y una modificación a la versión simplificada del Maximum Likelihood. Los dos primeros proponen una distancia adaptiva con el fin de detectar clusters de diferentes formas geométricas y orientación, es decir, generan clusters hiperelipsoidales que pueden ser aproximados a hiperplanos. La desventaja de estos algoritmos es que al derivar los antecedentes (funciones de pertenencia) que forman el modelo TSK, se producen errores por proyección o poca interpretabilidad cuado se corrige este error. El tercer algoritmo maneja clusters de forma hiperelipsoidal pero paralelos a los ejes de las variables de entrada. Debido a esto no se producen errores al derivar los antecedentes teniendo una alta interpretabilidad; la desventaja es que se pierde predictividad de modelo.

A. ALGORITMO GUSTAFSON-KESSEL

Este algoritmo se basa en la optimización de la siguiente función objetivo [3][4][5]:

$$\sum_{i=1}^{c}\sum_{k=1}^{N} (\mu_{ik})^m d_{ij}^2 \qquad (7)$$

Donde c es el número de clusters, N es el número de datos que se toman de la dispersión, μ_{ik} es la función de pertenencia del k-esimo dato al i-esimo cluster, m es un índice que indica la fusividad entre las fronteras de cada cluster. Entre más alto sea m más difusas son las fronteras de cada cluster y d_{ij}^2 es una distancia (norma) entre puntos denominada norma Mahalanobis, la cual es de la siguiente forma:

$$d_{ij}^2 = (z_k - v_i)^T A_i (z_k - v_i)$$

Donde z_k, es un vector columna de dimensión s que contiene el k-ésimo dato tomado, v_i también es un vector columna de dimensión s denominado centro del cluster y A_i es una matriz simétrica cuyo tamaño es (s x s), esta matriz indica la orientación de cada cluster. En este caso la norma Mahalanobis mide la distancia entre el k-ésimo dato z_k y el centro del i-ésimo cluster vi.

En el momento de minimizar se toman como variables μ_{ik}, A_i y v_i, teniendo en cuenta las siguientes restricciones [4][5]:

$$\sum_{i=1}^{c} \mu_{ik} = 1 \qquad 1 \le k \le N \qquad (8)$$

$$|A_i| = \rho_i \quad \rho_i > 0 \quad \forall i \qquad (9)$$

Donde ρ_i es denominado el volumen de cada cluster.

Para minimizar (7) con las restricciones (8) y (9) se utilizan multiplicadores de Lagrange. Sin las restricciones la minimización conduciría a respuestas triviales $\mu_{ik} = 0$ y $A_i = 0$. La primera restricción implica que la suma de los grados de pertenencia de cada dato a los clusters debe ser igual a uno, lo cual es acorde con la teoría de probabilidad. La segunda limita el volumen de cada cluster a un valor determinado.

En la teoría de probabilidad clásica se intenta predecir la probabilidad de que cierto dato o evento ocurra, de acuerdo a parámetros de una *función de densidad de probabilidad* (pdf) dada.

El proceso de minimización se realiza de forma alternante, es decir, se minimiza una variable suponiendo las otras constantes y se iguala a cero. Como resultado la minimización se obtiene las siguientes expresiones, sobre las cuales se realiza el proceso iterativo.

$$v_i = \frac{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m} \quad (10)$$

$$F_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^m (z_k - v_i)(z_k - v_i)^T}{\sum_{k=1}^{N} (\mu_{ik})^m} \quad (11)$$

$$Ai = \left[(\rho_i \det(F_i)^{1/n} F_i^{-1} \right] \quad (12)$$

$$d_{ij}^2 = (z_k - v_i)^T A_i (z_k - v_i) \quad (13)$$

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c} (d_{ik}/d_{jk})^{2/(m-1)}} \quad (14)$$

Las iteraciones inician determinando el número de cluster c, el exponente m (por lo general es 2) y seleccionando de manera aleatoria la matriz de partición difusa U de tamaño (c x N) la cual contiene los grados de pertenencia de los datos y cumple con (8). La matriz de partición difusa se encuentra organizada así:

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{c1} & \mu_{c2} & \cdots & \mu_{cN} \end{bmatrix} \quad (15)$$

Posteriormente se calcula el valor de los centros con (10). Teniendo el valor de v_i se calcula la matriz de covarianza difusa F_i. A partir de (12) se calcula A_i necesaria para hallar la distancia de la expresión (13). La ecuación (13) describe una hiperelipse rotada. La dirección y dimensión de cada uno de los ejes rotados está establecida por los autovectores y autovalores de la matriz A_i respectivamente Lo anterior fija la geometría de los clusters. La actualización de U se logra mediante (14). El proceso concluye cuando la diferencia entre U actual y U anterior es menor que el limite de tolerancia ε (0.01 ó 0.001).

Este algoritmo tiene la ventaja de ser prácticamente insensible a los parámetros de inicialización. Presenta limitaciones en cuanto al volumen de los cluster debido a la restricción en (9) incapacitándolo para identificar clusters con diferentes volúmenes. Además, si el sistema presenta linealidades o la cantidad de datos es muy pequeña, Ai puede presentar problemas de singularidad, interrumpiendo el proceso iterativo. En [5] se presentan dos técnicas para corregir los problemas matemáticos que pueden surgir en el cálculo de la matriz de covarianza Ai.

B. ALGORITMO MAXIMUM LIKELIHOOD ESTIMATION (MLE)

El concepto de likelihood está cercanamente relacionado con el de probabilidad [5][7][4]. En la teoría de probabilidad clásica se intenta predecir la probabilidad de que cierto dato o evento ocurra, de acuerdo a parámetros de una *función de densidad de probabilidad* (pdf) dada. El objetivo de MLE es encontrar los valores de los parámetros que maximicen el valor una pdf establecida a partir de los datos o eventos obtenidos.

Sin embargo, es más sencillo obtener los parámetros de la pdf a partir de la maximización del logaritmo natural de dicha función, esta técnica se conoce como *log likelihood* [7]. La pdf de la cual se obtiene los parámetros es una combinación lineal o suma de pdf's normales multivariables y cada una de estas representa un cluster.

Una función de densidad de probabilidad normal multivariable tiene la siguiente forma [4][7]:

$$g(x|i,\theta_i) = \frac{1}{(2\pi)^{\frac{s}{2}} |V_i|^{\frac{1}{2}}} e^{\frac{-(x-\mu_i)^T V_i^{-1} (x-\mu_i)}{2}} \quad (16)$$

La cual se entiende como la probabilidad que suceda el evento x dados los parámetros θ_i, estos son la i-ésima matriz de covarianza V_i y la i-ésima media μ_i (elemento prototipo del i-ésimo cluster). s es la dimensión de los datos, por tanto las dimensión de la matriz de covarianza es (s x s) y la de la media es s.

Al realizar una mezcla de pdf's normales la pdf resultante es de la forma:

$$f(x;\alpha) = \sum_{i=1}^{k} P_i g(x|i,\theta_i) \quad (17)$$

Que indica la suma de pdf's normales con coeficientes P_i. Ahora los parámetros de la función se conocen como α y son V_i, μ_i y P_i. Si se toma a P_i como la probabilidad a priori de seleccionar la i-ésima función, es decir, de seleccionar el i-ésimo cluster, se debe cumplir de acuerdo a la teoría clásica de probabilidad que:

$$\sum_{i=1}^{k} P_i = 1 \quad (12) \quad (18)$$

Suponiendo N eventos (datos) se desea encontrar una pdf que haga más probable a todos los eventos, es decir, más probable el evento 1 "y" el eventeo2 "y"… el evento N. La pdf resultante estará dada por:

$$H(x;\alpha) = f(x_1;\alpha) f(x_2;\alpha) \ldots f(x_N;\alpha) \quad (19)$$

Observe que esta pdf tiene exactamente los mismos parámetros que $f(x_k,\alpha)$. Ahora el logaritmo de H es: Al

$$Ln(H(x;\alpha)) = Ln(f(x_1;\alpha) f(x_2;\alpha) \ldots f(x_N;\alpha)) = \sum_{i=1}^{N} Ln f(x;\alpha) \quad (20)$$

AL aplicar MLE a esta función se buscan los parámetros α que maximicen Ln (H(x;a)), bajo la condición (18). Debido a esto se aplican multiplicadores de Lagrange para llegar a las siguientes expresiones:

$$P_i = \frac{1}{N}\sum_{k=1}^{N}\frac{P_i g(x_k|i;\theta_i)}{f(x_k;\alpha)} \quad (20)$$

$$\mu_i = \frac{\sum_{k=1}^{N}\frac{P_i g(x_k|i;\theta_i)}{f(x_k;\alpha)}x_k}{\sum_{k=1}^{N}\frac{P_i g(x_k|i;\theta_i)}{f(x_k;\alpha)}} \quad (21)$$

$$V_i = \frac{\sum_{k=1}^{N}\frac{P_i g(x_k|i;\theta_i)}{f(x_k;\alpha)}(x_k-\mu_i)(x_k-\mu_i)^T}{\sum_{k=1}^{N}\frac{P_i g(x_k|i;\theta_i)}{f(x_k;\alpha)}} \quad (22)$$

Ahora aplicando la regla de Bayes [8] se obtiene:

$$\frac{P_i g(x_k|i;\theta_i)}{f(x_k;\alpha)} = \frac{P_i\, g(x_k|i;\theta_i)}{\sum_{i=1}^{k} P_i g(x_k|i;\theta_i)} \quad (24)$$

$$= p(i;\theta_i|x_k)$$

Entonces se puede rescribir de la siguiente forma el algoritmo:

$$\mu_i = \frac{\sum_{k=1}^{N} p(i;\theta_i|x_k)x_k}{\sum_{k=1}^{N} p(i;\theta_i|x_k)} \quad (25)$$

$$V_i = \frac{\sum_{k=1}^{N} p(i;\theta_i|x_k)(x_k-\mu_i)(x_k-\mu_i)^T}{\sum_{k=1}^{N} p(i;\theta_i|x_k)} \quad (26)$$

$$P_i = \frac{1}{N}\sum_{k=1}^{N} p(i;\theta_i|x_k) \quad (21)$$

$$d_{ij}^{\,2} = \frac{(2\pi)^{\frac{s}{2}}|V_i|^{\frac{1}{2}}}{P_i} e^{\frac{(x-\mu_i)^T V_i^{-1}(x-\mu_i)}{2}} \quad (27)$$

Al igual que en GK, (27) describe la ecuación de elipses rotadas que determina la geometría del cluster. Sin embargo, este algoritmo no tiene restricciones en cuanto al volumen de cada cluster. El proceso de actualización de los parámetros y convergencia del proceso es similar al utilizado en GK. Una de las desventajas que presenta este algoritmo es que debido a su distancia exponencial es sensible a los parámetros de inicialización.

En la versión de ejes paralelos simplificada la matriz de covarianza se toma como una matriz diagonal. Cada uno de los elementos de la matriz es la desviación estándar de cada una de las variables en cada cluster.

C. VERSIÓN DE EJES PARALELOS SIMPLIFICADA DEL MAXIMUM LIKELIHOOD ESTIMATION

Como ya se dijo el algoritmo MLE tiene como una de las variables a minimizar la matriz de covarianza. Esto produce clusters rotados de forma hiperelipsoidal, de acuerdo a la orientación de cada subgrupo.

En la versión de ejes paralelos simplificada la matriz de covarianza se toma como una matriz diagonal[9][10]. Cada uno de los elementos de la matriz es la desviación estándar de cada una de las variables en cada cluster[9]. Con este nuevo planteamiento cada una de las pdf's $g(x|i,\theta_i)$,que compone la mezcla se puede rescribir como:

$$g(x|i,\theta_i) = \prod_{r=1}^{s}\frac{1}{\sqrt{2\pi\sigma^2_{ir}}} e^{\frac{-(x_r-\mu_{ir})^2}{2\sigma^2_{ir}}} \quad (28)$$

donde x es el vector que representa un dato tomado, s la dimensión de cada dato, x_r representa la r-ésima componente del vector x, μ_{ir} es r-ésima componente del centro del i-ésimo cluster μ_i y σ^2_{ir} representa la desviación estándar de la r-ésima variable en el i-ésimo cluster. Siguiendo un procedimiento similar al descrito para obtener el algoritmo del MLE se llega a:

$$\mu_i = \frac{\sum_{k=1}^{N} p(i;\theta_i|x_k)x_k}{\sum_{k=1}^{N} p(i;\theta_i|x_k)} \quad (29)$$

$$\sigma^2_{ij} = \frac{\sum_{k=1}^{N} p(i;\theta_i|x_k)(x_{rk}-\mu_{ir})(x_{kr}-\mu_{ir})^T}{\sum_{k=1}^{N} p(i;\theta_i|x_k)} \quad (30)$$

$$P_i = \frac{1}{N}\sum_{k=1}^{N} p(i;\theta_i|x_k) \quad (31)$$

$$d_{ij}^{\,2} = \prod_{r=1}^{s}\frac{\sqrt{2\pi\sigma^2_{ir}}}{P_i} e^{\frac{(x_r-\mu_{ir})^2}{2\sigma^2_{ij}}} \quad (31)$$

$$p(i;\theta_i|x_k) = \frac{1}{\sum_{j=1}^{c}(d_{ik}/d_{jk})^2} \quad (33)$$

Debido a que la matriz es diagonal, solo se obtendrán clusters de forma elipsoidal pero paralelos a cada uno de los ejes del espacio muestreal. Obsérvese que teniendo en cuenta la ecuación (28), la funciona objetivo de la ecuación (17) se puede rescribir como:

$$f(z;\alpha) = \sum_{i=1}^{c} p(\eta_i)g(z|\eta_i) = \sum_{i=1}^{c} p(x,y/\eta_i)p(\eta_i) = \sum_{i=1}^{c} p(y/x,\eta_i)p(x/\eta_i)p(\eta_i) \quad (34)$$

El problema de derivar los modelos TSK se reduce a hallar los parámetros de la función lineal en los consecuentes y las funciones de pertenencia en los antecedentes.

En este caso cada dato es representado por un vector z y es dividido en un vector de entrada x y una constante de salida y. A $p(x_k/\eta_i)$ se le denomina distribución de entrada y es de la forma:

$$p(x_k/\eta_i)=\prod_{r=1}^{n}\frac{1}{\sqrt{2\pi\,\sigma^2_{ir}}}e^{\frac{-(x_r-\mu_{ir})^2}{2\sigma^2_{ir}}} \quad (35)$$

donde n es la dimensión de entrada y $p(y/x,\eta_i)$ se le denomina distribución de salida y es de la forma:

$$p(y/x,\eta_i)=\frac{1}{\sqrt{2\pi\sigma^2_{iy}}}e^{\frac{-(y-\mu_y)^2}{2\sigma^2_{iy}}} \quad (36)$$

D. MODIFICACIÓN DE LA VERSIÓN SIMPLIFICADA DEL MLE

Esta modificación se basa en interpretación dada por las ecuaciones (34), (35) y (36) a la versión simplificada del MLE y busca mejorar la interpretación a la hora de derivar modelos TSK. Esta variación plantea una distribución de entrada de la forma de la ecuación (35), y una distribución de salida en la que a la vez se evalúa un modelo lineal de la salida como función de las variables de entrada [9][10].

En esta técnica las distribuciones de entrada y salida están determinadas por:

$$p(x_k/\eta_i)=\prod_{r=1}^{n}\frac{1}{\sqrt{2\pi\,\sigma^2_{ir}}}e^{\frac{-(x_r-\mu_{ir})^2}{2\sigma^2_{ir}}} \quad (37)$$

$$p(y/x,\eta_i)=\frac{1}{\sqrt{2\pi\pi_i^2}}e^{\frac{(y-x^T\theta_i)^T(y-x^T\theta_i)}{2\sigma_i^2}} \quad (38)$$

Donde θ_i indica los parámetros del i-ésimo modelo lineal de salida. También se presenta la siguiente restricción:

$$\sum_{i=1}^{c}p(\eta_i)=1 \quad (39)$$

Siendo $p(\eta_i)$ la probabilidad a priori de cada cluster. Utilizando un proceso similar al desarrollado para obtener el algoritmo MLE se llega a las siguientes expresiones que genera el proceso iterativo.

$$p(\eta_i|x_k,y_k)=\frac{1}{\sum_{j=1}^{c}(d_{ik}/d_{jk})^2} \quad (40)$$

$$p(\eta_i)=\frac{1}{N}\sum_{k=1}^{N}p(\eta_i/x,y) \quad (41)$$

$$v_i^x=\frac{\sum_{k=1}^{N}x_k p(\eta_i/x_k,y_k)}{\sum_{k=1}^{N}p(\eta_i/x_k,y_k)} \quad (42)$$

$$\sigma_{ij}=\frac{\sum_{k=1}^{N}(x-v_i^x)(x-v_i^x)^T p(\eta_i/x_k,y_k)}{\sum_{k=1}^{N}p(\eta_i/x_k,y_k)} \quad (43)$$

$$\sigma_i=\frac{\sum_{k=1}^{N}(y-(a_i x+b_i))(y-(a_i x+b_i))^T p(\eta_i/x_k,y_k)}{\sum_{k=1}^{N}p(\eta_i/x_k y_k)} \quad (44)$$

$$\theta_i=(X_e^T\Phi_i X_e)^{-1}X_e^T\Phi_i y \quad (45)$$

X_e denota una matriz de regresión extendida, la cual se forma de la siguiente manera: primero que todo se forma una matriz llamada X, donde a cada fila le corresponde el valor de las componentes de entrada de cada uno de los datos. Después se adiciona una columna de unos obteniendo X_e, es decir $X_e=[X\ 1]$ y Φ_i denota una matriz diagonal que contiene las funciones de pertenencia de cada dato al i-ésimo cluster.

$$\Phi=\begin{bmatrix}p(\eta_i|x_1,y_1) & 0 & \cdots & 0\\ 0 & p(\eta_i|x_2,y_2) & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \cdots & p(\eta_i|x_N,y_N)\end{bmatrix} \quad (46)$$

$$dij=\prod_{r=1}^{n}\frac{\sqrt{2\pi\sigma^2_{ir}}}{p(\eta_i)}e^{\frac{(x_r-\mu_{ir})^2}{2\sigma^2_{ir}}}\sqrt{2\pi\sigma_i^2}\,e^{\frac{(y-x^T\theta_i)^T(y-x^T\theta_i)}{2\sigma_i^2}} \quad (47)$$

El proceso de actualización de los parámetros y convergencia del algoritmo es similar a GK y se utiliza (40) para la reconstrucción de U. Al igual que MLE este algoritmo presenta problemas de inicialización.

V. DERIVACIÓN DE MODELOS TSK A PARTIR DE LOS PARAMETROS OBTENIDOS EN LOS METODOS DE CLUSTERING

El problema de derivar los modelos TSK se reduce a hallar los parámetros de la función lineal en los consecuentes y las funciones de pertenencia en los antecedentes. Se debe tener en cuenta que el método modificado de la versión simplificada del MLE brinda ya todos los parámetros de los consecuentes. Por tanto solo se describirán métodos para derivar consecuentes a partir de las particiones y parámetros obtenidos en los métodos de clustering GK y MLE

A. DERIVACIÓN DE CONSECUENTES

Para derivar los consecuentes los dos métodos usados más comúnmente son: mínimos cuadrados totales y mínimos cuadrados ponderados [4]:

Mínimos cuadrados totales (TLS): Como resultado de las técnicas de clustering se obtienen clusters de forma hiperelipsoidal que pretenden aproximar hiperplanos, por tanto el eje más pequeño del hiperelipse debe tender a ser perpendicular al hiperplano que se quiere aproximar. Las direcciones y las magnitudes de los ejes de los hiperelipses están determinadas por los autovectores y los autovalores de cada matriz de covarianza respectivamente. El autovalor más pequeño indica cual de los ejes del hiperelipse es más pequeño y por tanto indica que el correspondiente autovector es perpendicular al hiperplano. Basándose en esta idea y teniendo en cuenta que el hiperplano debe pasar por cierto punto con determinadas coordenadas llamado centro se construye la ecuación de un hiperplano:

$$\varphi_i^T (z - v_i) = 0 \qquad (48)$$

Donde φ_i^T es el autovector correspondiente al autovalor más pequeño de la i-ésima matriz de covarianza y v_i es el i-ésimo centro. Ahora el autovector φ_i y el centro v_i se pueden rescribir partiéndolos en sus dimensiones de la siguiente forma:

$$\varphi_i^T = \left[(\varphi_i^x)^T, \varphi_i^y \right] \quad v_i^T = \left[(v_i^x)^T \; v_i^y \right] \qquad (49)$$

Donde $(\varphi_i^x)^T$ y $(v_i^x)^T$ son la parte de los vectores φ_i^T y v_i^T respectivamente que corresponden a las variables de entrada mientras que φ_i^y y v_i^y son las constantes que corresponden a la salida Teniendo en cuenta esto la ecuación (44) se puede rescribir de la siguiente forma.

$$\left[(\varphi_i^x)^T, \varphi_i^y \right] \left(\left[x^T, y \right]^T - \left[(v_i^x)^T, v_i^y \right]^T \right) = 0 \qquad (50)$$

Lo que conduce a:

$$y = \frac{-1}{\varphi_i^y} (\varphi_i^x)^T x + \frac{1}{\varphi_i^y} (\varphi_i^T) v_i \qquad (51)$$

Donde

$$a_i = \frac{-1}{\varphi_i^y} (\varphi_i^x)^T \qquad b_i \frac{1}{\varphi_i^y} (\varphi_i^T) v_i \qquad (52)$$

que son los consecuentes de la i-ésima regla. Esta solución es basada en la interpretación geométrica de los cluster, pero también esta demostrado [4] que es la solución al sistema:

$$\Delta y_k^{-i} = a_i^T \Delta x_k^{-i} \qquad (53)$$

Mínimos cuadrados ponderados (WLS): Este métodos plantea minimizar los errores de predicción de los modelos locales individuales, resolviendo como un conjunto de c independientes problemas (uno para cada cluster) de mínimos cuadrados ponderados. Lo anterior se obtiene a partir de la minimización del siguiente criterio:

Como resultado del clustering se obtienen funciones de pertenencia multivariable.

$$(y - X_e \theta_i)^T \Phi_i (y - X_e \theta_i) \qquad (54)$$

donde X_e es la misma matriz que se describió en la sección IV. Como resultado se obtiene la ecuación (45), es decir:

$$\theta_i = \left[a_i^T, b_i \right] \qquad (55)$$

Es conveniente utilizar TLS en lugar de WLS, en datos donde se considere que existe ruido, ya que TLS brinda de cierta forma robustez frente al ruido. También es importante resaltar que en la modificación a la versión simplificada de MLE en cada iteración del algoritmo se esta usando WLS, por tanto el propio algoritmo brinda los consecuentes sin necesidad de procedimientos extra.

B. DERIVACIÓN DE ANTECEDENTES

Como resultado del clustering se obtienen funciones de pertenencia multivariable. Es difícil la interpretación e implementación multidimensional de conjuntos difusos, por lo tanto los antecedentes en (1) deben ser reescritos como una combinación de proposiciones simples con conjuntos difusos unidimensionales. Para este fin se pueden aplicar dos estrategias: proyección ortogonal y proyección autovector [4].

Proyección Ortogonal: Proyecta la función de pertenencia multivariable sobre cada uno de los ejes correspondientes a las diferentes variables de entrada de la siguiente forma:

$$\mu_{Ai}(x_j) = \text{proj}_j \, U(i) \qquad (56)$$

Esta proyección presenta errores ya que los clusters no siempre están paralelos a los ejes y se produce el denominado error por proyección.

Proyección con autovector: Ya se describió como los autovectores indican la dirección de los ejes rotados de las hyperelipses, por tanto se puede utilizar esta propiedad para formar una matriz de rotación y luego proyectar ortogonalmente. Esta solución aunque más precisa, tiene el problema que es poco interpretable, ya las nuevas entradas se convierten en combinaciones lineales de la entradas originales (por el proceso de rotación de ejes a través de la matriz).

Una vez obtenidas las funciones de pertenencia univariables por cualquiera de los dos métodos de proyección, se deben usar una curva parametrizada que aproxime lo mejor posible el punto proyectado.

VI. SIMULACIONES Y RESULTADOS

En esta sección son desarrollados 3 ejemplos para verificar la robustez y exactitud en la generación de modelos. Para la generación de antecedentes se implementó la proyección ortogonal descrita en la sección V.B.. En cuanto a los parámetros de los con-

Comparando las gráficas 2(a) y 2(b), se puede evidenciar mejor el hecho que el algoritmo GK tiende a generar clusters de igual volumen.

secuentes fueron derivados a partir del método TLS para los algoritmos GK y MLE. WLS fue implementado para la modificación de MLE simplificado. En el primer ejemplo de aplicación se tomó una función compuesta de dos funciones lineales, descrita de la siguiente forma:

$$\begin{aligned} y_1 &= x_1 \qquad x_1 \in [0,1] \\ y_2 &= -3x_2 + 4 \qquad x_2 \in [1,2] \end{aligned} \tag{57}$$

Fue agregado ruido gaussiano con media 0 y desviación estándar de 0.2, N(0, σ^2). Se tomaron 1000 muestras por cada submodelo, es decir 2000 muestras en total. Los funciones de pertenencia obtenidas para los antecedentes son mostrados en las figura 1. La función, los datos con ruido agregado y los consecuentes obtenidos son mostrados en la figura 2 Además los parámetros encontrados para los consecuentes son tabulados en la tabla 1.

Se puede observar en la Fig.1(a) que la frontera (punto en el cual el valor de la función de pertenencia es 0.5) originada por el algoritmo GK entre ambos clusters esta alrededor de 1,2. Idealmente esta frontera se debería encontrar en 1, tal como lo sugiere la ecuación (57). Este desplazamiento es debido a la presencia de ruido, sin embargo, es una buena aproximación. Los mejores resultados se obtuvieron cuando se empleó el algoritmo MLE donde la frontera se encuentra cercana a 1.1 lo cual significa que agrupó la dispersión de manera más eficiente, tal como se ve en la Fig. 1(b). El peor resultado es obtenido al utilizar la modificación de MLE ya que la frontera se presenta de 0.75, además de esto las proyecciones que representan las funciones de pertenencia son las más dispersas como se observa en la Fig.1(c).

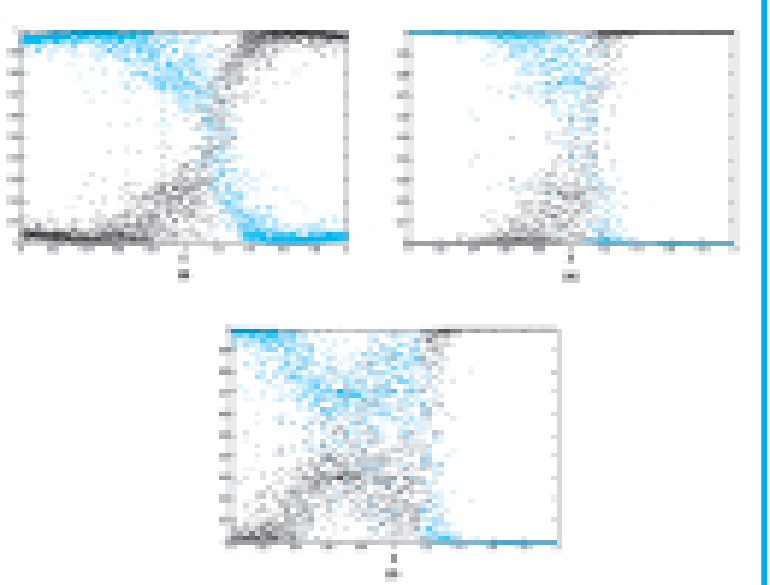

Figura 1. Funciones de pertenencia obtenidas a partir de (a)GK, (b)MLE, (c)MLE modificado, para el ejemplo de aplicación. 1.

TABLA 1. PARÁMETROS DEL CONSECUENTE OBTENIDOS CON CADA ALGORITMO.

	A1	B1	A2	B2
GK	-3.3874	4.3068	0.6955	0.0925
MLE	-3.9561	5.6393	0.6698	0.0683
MOD MLE	-0.9507	-0.3026	-0.6642	0.6653

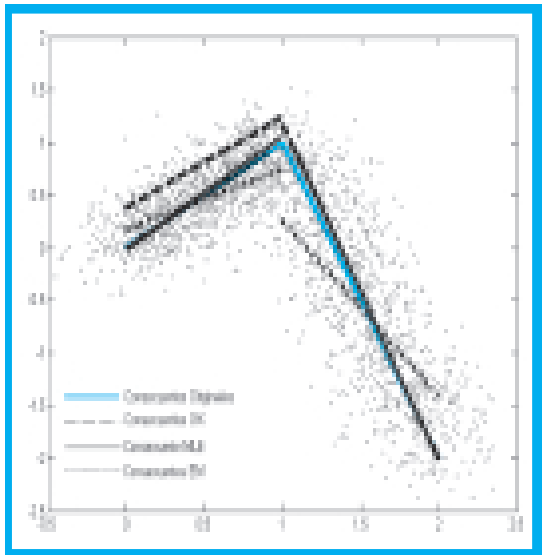

Figura 2. Comparación entre los consecuentes originales ("—") y los consecuentes obtenidas a partir cada algoritmo. ("—")GK, ("——")MLE, ("•••") modificado

Comparando las graficas 2(a) y 2(b), se puede evidenciar mejor el hecho que el algoritmo GK tiende a generar clusters de igual volumen, ya que los parámetros del primer submodelo fueron bastante próximos a los ideales, mientras que los del segundo son apenas buenos. Los parámetros obtenidos con el algoritmo MLE son bastantes buenos para ambos clusters. Observando la grafica 2(c) se confirma el hecho evidenciado en la Fig. 1(c), donde se observa que la modificación del algoritmo MLE no identifica bien los submodelos lineales, ya que los consecuentes hallados a través de esta modificación se encuentran lejanos a los originales.

En el segundo ejemplo se toma una función compuesta idéntica a la del ejemplo 1, variando la desviación estándar a 0. Los funciones de pertenencia obtenidas para los antecedentes y son mostrados en las figura 3. La función, los datos con ruido agregado y los consecuentes obtenidos son mostrados en la figura 4 Además los parámetros encontrados para los consecuentes son tabulados en la tabla 2.

TABLA 2. PARÁMETROS DE LOS CONSECUENTES OBTENIDOS CON CADA ALGORITMO PARA EL EJEMPLO 2.

	A1	B1	A2	B2
GK	-3.3874	4.3068	0.6955	0.0925
MLE	-3.9561	5.6393	0.6698	0.0683
MOD MLE	-0.9507	-0.3026	-0.6642	0.6653

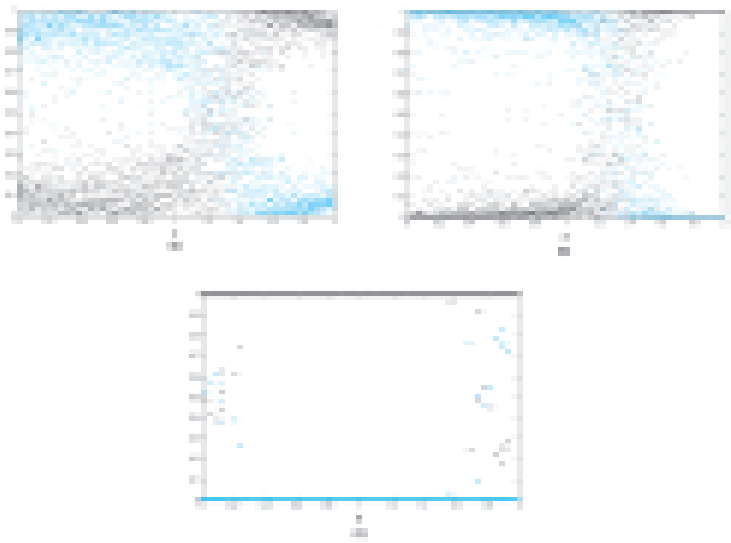

Figura 3. Funciones de pertenencia obtenidas a partir de (a)GK, (b)MLE, (c)MLE modificado, para el ejemplo de aplicación.

De la Fig.3(a) y 3(b) se puede observar que los puntos de frontera se colocan aproximadamente en 1.3 para las funciones de pertenencia obtenidas a partir de los algoritmos GK y MLE. La diferencia radica que con GK se derivan conjuntos más difusos, mientras MLE tienden a generar conjuntos concretos (no difusos). El error en el punto de frontera de nuevo se debe al ruido que en este caso es mayor que en el ejemplo 1. Los resultados obtenidos con la modificación del MLE son pésimos. El algoritmo identificó la partición como un solo cluster, ya que, para uno de los conjuntos, el valor de la función de pertenencia es uno en todo el rango de la variable de entrada, mientras que en el otro conjunto cero, tal como se ve en la Fig. 3.(c).

Al observar la Fig. 4 y la tabla 2 se puede concluir que el algoritmo que mantiene un aceptable desempeño para derivar consecuentes es el algoritmo GK.

Al observar la Fig. 4 y la tabla 2 se puede concluir que el algoritmo que mantiene un aceptable desempeño para derivar consecuentes es el algoritmo GK.

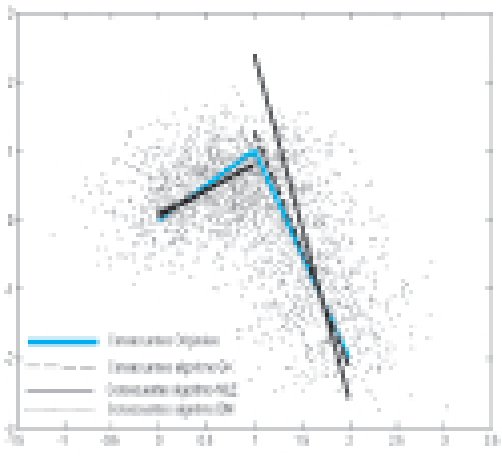

Figura 4. Comparación entre los consecuentes originales (“—”) y los consecuentes obtenidas a partir cada algoritmo. (“—“)GK, (“——”)MLE, (“•••”) modificado, para el ejemplo 2

Con el MLE no se derivan consecuentes aproximados a los reales para ninguno de los dos clusters, su desempeño es pobre en comparación con el obtenido en GK. Confirmando así los resultados obtenidos en las funciones de pertenencia. A partir de la Fig. 4. (c) se puede observar que el MLE modificado presenta un pésimo desempeño a la hora de obtener consecuentes representativos de la dispersión de datos. En el tercer ejemplo se tomó una función no lineal descrita en la siguiente ecuación:

$$Y=1x10^{-4}\sin(0.001x^2)\ x^3 \qquad (58)$$

A dicha función le fue agregado ruido gaussiano $N(0, \sigma^2)$ con media 0 y desviación estándar de 5. Se tomaron 200 muestras en total. Los resultados obtenidos son mostrados en las graficas 5, 6 y 7.

En la figura 5 se puede observar que los 3 algoritmos pueden identificar bastante bien los subgrupos dentro de la dispersión de datos. En el algoritmo GK se presenta conjuntos difusos con transiciones suaves mientras que las transiciones obtenidas con los algoritmos MLE y MLE modificados son bastante abruptas.

En la Fig. 6 se muestran los submodelos lineales obtenidos a partir de cada algoritmo de clustering. Nótese que el algoritmo GK no identifica muy bien los dos primeros cluster mientras MLE y su modificación no identifican muy bien el segundo cluster.

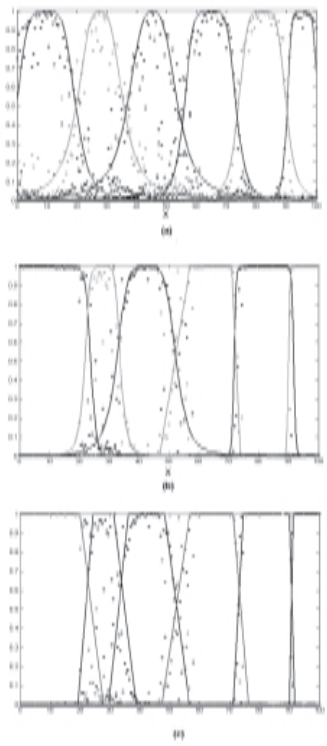

Figura 5. Funciones de pertenencia obtenidas por proyección(".")y las curvas parametricas que las aproximan ("-"), a partir de (a)GK, (b)MLE, (c)MLE modificado, para el ejemplo 3.

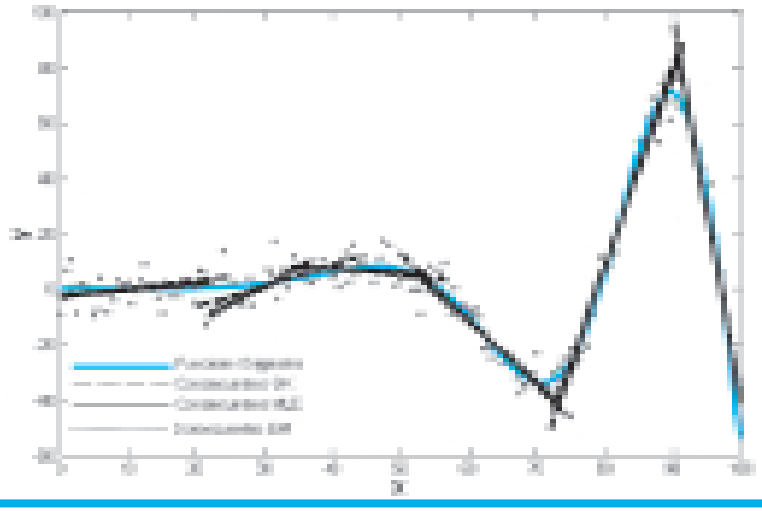

Figura 6. Curva generada por la ecuación (58) ("—"), con ruido agregado ("+"). Submodelos lineales obtenidos a partir de cada algoritmo. ("—")GK, ("——")MLE, ("•••")MLE modificado.

En la figura 7 se muestra la salida que produce cada uno de los modelos. Para poder evaluar cuantitativamente el desempeño del modelo, fue aplicado el criterio RMSE, el cual es definido así:

$$RMSE = \frac{\sqrt{\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}}{N} \qquad (59)$$

TABLA 3. CONTIENE LOS RESULTADOS OBTENIDOS CON EL CRITERIO DE EVALUACIÓN RMSE

Algoritmo	RMSE
GK	0.2983
MLE	0.2741
MOD MLE	0.2543

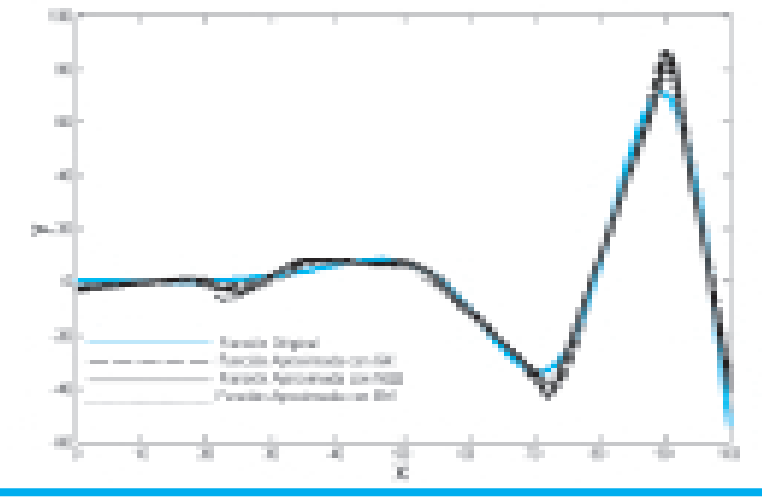

Figura 7. Curva generada por la ecuación (58) ("—") junto con el modelo Takagi-Sugeno-Kang (TSK) obtenido ("-") a partir de los algoritmos ("—")GK, ("——")MLE, ("•••")MLE modificado.

El algoritmo que obtuvo el peor desempeño fue el GK, confirmando el hecho analizado con la figura 6, donde GK fue el algoritmo que no identifico apropiadamente dos de los submodelos lineales. En este caso el que obtuvo un mejor desempeño fue el MLE modificado, porque logro identificar submodelos lineales más cercanos al comportamiento real.

VII. CONCLUSIONES

Las técnicas de clustering son una excelente herramienta para derivar modelos difusos tipo TSK capaces de aproximar sistemas no lineales.

Observando los ejemplos 1 y 2 se puede concluir que el algoritmo más robusto frente al ruido es GK. Por el contrario el más sensible ante el ruido es MLE modificado. Sin embargo en condiciones de poco ruido el algoritmo MLE modificado puede identificar mejor submodelos lineales que sean más próximos al modelo general.

Para trabajos futuros se puede pensar en aplicar técnicas de clustering robusto a los algoritmos con el fin de volverlos menos insensibles ante la presencia de ruido.

REFERENCIAS BIBLIOGRÁFICAS

[1] L.X.Wang, A Course in Fuzzy Systems and Control. Prentice Hall, 1997.

[2] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control", IEEE Trans.on Systems, Man and Cybernetics, vol. SMC-15, No.1, 1985

[3] D. E. Gustafson and W. C. Kessel, "Fuzzy Clustering with a Fuzzy Covariance", Proc.IEEE CDC, pp 761-766, San Diego, CA, USA, 1979

[4] R. Babuska, Fuzzy Modeling for Control. Kluwer Academic Publishers, 1998.

[5] R. Babuska, "et al", "Improved Covariance Estimation for Gustafson-Kessel Clustering", Proc. Of the 2002 IEEE International Conference on Fuzzy Systems,vol. 2, pp. 1081-1085, 2002.

[5] I. Gath and A. B. Geva, "Unsupervised Optimal Fuzzy Clustering", IEEE Trans. On Pattern Analisis and Machine Intelligence, vol. 11, No. 7, pp. 773-781, 1989.

[7] J. C. Bezdek and J. C. Dunn, "Optimal Fuzzy Partitions": A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions", IEEE Trans.on Computers, pp. 835-838, 1975

[8] P. L. Meyer, Probabilidad y Aplicaciones Estadisticas, Addison Wesley, 1992.

[9] J. Abonyi, "et al", "Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models", IEEE Trans. on Systems, Man and Cybernetic, Part B, October, 2002.

[10] J. Abonyi, "et al", "Identification of Nonlinear System Using Gaussian Mixture of local Models", Hungarian Journal of Industrial Chemestry, vol. 29, 134-139, 2001.

Diana Marcela González Chaparro
Miembro del Grupo de investigación: Laboratorio de Automática, Microelectrónica e Inteligencia Computacional LAMIC.

Sergio Barato Quintero
Miembro del Grupo de investigación: Laboratorio de Automática, Microelectrónica e Inteligencia Computacional LAMIC.